Université Sidi Mohammed Ben Abdellah
Faculté des Sciences Dhar El Mahraz- Fès
Centre d'Etudes Doctorales
"Sciences et Technologies"

Formation Doctorale : Sciences et Technologies de l'Information et de la Communication (STIC)

Spécialité : Informatique

Laboratoire : Laboratoire d'Informatique et Modélisation

THESE DE DOCTORAT

Présentée par

Nabil Alami

Contributions to the Improvement of Automatic Summarization of Arabic Texts

Soutenue le 22 /12 /2018 devant le jury composé de :

Pr. Khalid Satori	***Faculté des Sciences Dhar El Mahraz – Fès***	***Président***
Pr. Brahim Ouhbi	***Ecole Nationale Supérieure d'Arts et Métiers – Meknès***	***Rapporteur***
Pr. Arsalane Zarghili	***Faculté des Sciences et Techniques – Fès***	***Rapporteur***
Pr. Si Lhoussain Aouragh	***Faculté des Sciences Juridiques, Economiques et Sociales – Salé***	***Rapporteur***
Pr. Saïd Ouatik El Alaoui	***Faculté des Sciences Dhar El Mahraz – Fès***	***Examinateur***
Pr. Khalid Alaoui Zidani	***Faculté des Sciences Dhar El Mahraz – Fès***	**Examinateur**
Pr. Noureddine En-nahnahi	***Faculté des Sciences Dhar El Mahraz – Fès***	**Examinateur**
Pr. Mohammed Meknassi	***Faculté des Sciences Dhar El Mahraz – Fès***	**Directeur de thèse**

Année universitaire : 2018-2019

Dedication

This thesis is dedicated to:

My loving parents, who never stop giving of themselves in countless ways. I especially appreciate their unconditional support, without them I would not have made it through my PhD studies.

My mother Fatima: This ray of sunshine that never ceases to enlighten my life, this inexhaustible source of love, tenderness and affection, thanks for her empathy and her giving without being asked to.

My father Abdessalam: The symbol of all sacrifice and dedication. His helpers and recommendations have often encouraged me to persevere in the daily effort and to progress in my personal and professional life. Thanks you for being patient with me and for your devoted support.

No words would be enough to reward your sacrifices and patience. I would like to say a heartfelt thanks to you for always believing in me and encouraging me to follow my dreams.

My dearest wife Kamilia, who leads me through the valley of darkness with light of hope. I am very much indebted to her. She has been by my side throughout this PhD, living every single minute of it, and without her, I would not have had the courage to embark on this journey in the first place. I love you so much.

My beloved kids: Iness, and Nizar, whom I can't force myself to stop loving. This work is dedicated to you for being such a good little babies making it possible for me to complete what I started. You have made me stronger, better and more fulfilled than I could have ever imagined. I love you to the moon and back.

My beloved brother Mohammed and sisters Fatima, Latifa, Badia, Aicha, Fouzia and Ikram who gave me continuous encouragement and uninterrupted stimulus throughout all my years at university. I wish you all the best in your life.

To My friends who encourage and support me,

All the people in my life who touch my heart, I dedicate this work.

Acknowledgements

Above all and the most importantly, I owe it all to Almighty God for granting me the wisdom, health and strength to undertake this research work and enabling me to its completion.

Many people contributed in the elaboration of this PhD thesis in one way or another, without whom this work probably could not have been completed, for this reason I will take this opportunity to express my immense gratitude to each and every one of them.

First and foremost, I would like to express my sincere gratitude to my supervisor Pr. Meknassi Mohammed for giving me the opportunity to join his research team, for the continuous support of my Ph.D study and related research, for carrying out stimulating discussions despite his busy schedule, for his patience, motivation, and immense knowledge. I would also like to thank him for his personal and human qualities which have also contributed a lot to the accomplishment of this work. I would like to thank him for always having his door open for lively discussion and for his all valuable comments and suggestions during the accomplishment of this thesis work. His enthusiasm motivated me a lot. Language is incapable of expressing my sincere debt and warm thanks to him.

I particularly thank Pr. Said Ouatik El Alaoui and Pr. Noureddine En-nahnahi for their insightful remarks and suggestions. I cannot but record my deep gratitude and appreciation to them. This work's process provided me with the pleasure of working with them.

I am also happy to acknowledge my thankfulness to my reading committee members: Pr. Brahim Ouhbi, Pr. Arsalane Zarghili and Pr. Si Lhoussain Aouragh for their time, interest, and helpful comments, as well as the three members of my oral defense committee, Pr. Said Ouatik El Alaoui, Pr. Khalid Alaoui Zidani and Pr. Noureddine En-Nahnahi for their time, invaluable comments and advices.

My thanks goes also to the Faculty of Science Dhar EL Mahraz and its faculty and administrative staff, especially the Laboratory of Informatics and Modeling for giving me the opportunity to work with some of the best and brightest to achieve great success. I hereby acknowledge the valuable work of the committee at the laboratory and their continuous efforts to bring the best out of its members and PhD candidates.

I would extend my warm thanks to my family, friends and colleagues who gave me continuous encouragement and uninterrupted stimulus throughout all my years at university.

This thesis is in the world for those who believed in me all along. Too many to name, but not too many to appreciate.

Thank You all.

الخلاصة

هذه الرسالة تخص التلخيص الآلي للنصوص العربية. نحن مهتمون بشكل خاص بتعزيز الأساليب الاستخراجية، وذلك بالاعتماد على الطرق الإحصائية والدلالية والتعلم الآلي. نقوم أولاً بعرض مستوى التقدم الجاري فيما يتعلق بالأساليب الرئيسية لتلخيص النصوص، لا سيما تلك المخصصة للغة العربية. بعد ذلك، نُفصِّل المساهمات الأربع التي قمنا بها خلال هذه الرسالة لتحسين أداء الطرق المستخدمة حاليا. في أول مساهمة، نقترح طريقة جديدة للتلخيص الآلي للنصوص العربية من خلال نمذجة النص في شكل رسم بياني ثنائي الأبعاد بحيث تكون الرؤوس هي جمل النص ويكون رابطان بين كل رأسان يدلان على وجود نوعان من التشابه بين جملتين ويمثلان كلا من الدرجات الإحصائية والدلالية أو المعنوية الموجودة بين كل زوج من الجمل. بالإضافة إلى ذلك، قمنا بدمج خوارزمية إلغاء التكرار وخطوة ما قبل معالجة النص من أجل تحسين أداء الطريقة المقترحة. في الإسهام الثاني، نقترح طريقة جديدة للتلخيص الآلي للنصوص العربية تعتمد على التعلم العميق. وذلك باستخدام التشفير التلقائي التغيري (Variational Auto-Encoder) كتقنية للتعلم غير مراقب للخصائص من أجل استخراج تمثيل مجرّد لكل جملة في فضاء مفهوم. يستخدم هذا التمثيل لتصنيف الجمل في النص وفقا لتشابهها مع طلب المستخدم، ومن ثم، استخراج أكثرها صلة. هناك بديل آخر مقترح وهو دمج هذا التمثيل التجريدي من أجل حساب التشابه بين كل زوج من الجمل من خلال تبني نموذج الرسم البياني المقترح في المساهمة الأولى. تتيح الطريقة المقترحة، من ناحية، تقليص الأبعاد، ومن ناحية أخرى، تحسين عملية استخراج الجمل المعنية والمهمة. في مساهمتنا الثالثة، نعتمد تقنية تضمين الكلمات (Word Embeddings / Word2Vec) كمدخل لتدريب عدة نماذج من الشبكات العصبونية الاصطناعية غير المراقبة. يتم استخدام تمثيلات الجمل الجديدة التي تم الحصول عليها من أجل حساب التشابه بين كل زوج من الجمل وذلك لبناء الرسم البياني السابق. كما نقترح نماذج جديدة تعتمد على تقنية التعلم الجماعي لتحسين جودة التلخيص الآلي للنصوص العربية. وأخيرًا، تتمثل مساهمتنا الرابعة في اعتماد تقنية التحليل العنقودي (او التجميع) لتجميع النصوص في مجموعات متعددة والتي نحدد لكل منها فضاء المواضيع المرتبطة بها وذلك باستخدام طريقة التخصيص دركليه الكامنة Latent Dirichlet) . (Allocationبعد ذلك، نستخدم نموذج تمثيل النص على شكل مصفوفة في فضاء المواضيع المحدد لكل مجموعة كبيانات تدريب خاصة بالشبكات العصبونية الاصطناعية غير المراقبة ونماذج التعلم الجماعي من أجل تعلم تمثيلات تجريدية جديدة للنص. تستخدم هذه التمثيلات الجديدة لتصنيف جمل النص ليتم تلخيصه وفقاً لنموذج الرسم البياني المعتمد سابقا. يتم تقييم أداء جميع المناهج المقترحة باستخدام العديد من مجموعات البيانات. تظهر النتائج التي تم الحصول عليها أهمية مقترحاتنا.

الكلمات الجوهرية: التلخيص الآلي، النصوص العربية، معالجة اللغات الطبيعية، علم المعاني، تضمين الكلمات، التعلم الآلي، الشبكات العصبونية الاصطناعية، التعلّم العميق، التعلّم الجماعي.

Abstract

This thesis work is part of automatic summarization of Arabic texts. We are particularly interested in enhancing extractive methods, drawing on statistical, semantic and machine learning approaches. First, we present a state-of-the-art regarding the main methods of automatic text summarization (ATS), in particular, those designed to Arabic. Next, we describe four contributions to improve the performance of existing methods. In the first contribution, we propose a new method for ATS by modeling the text in the form of a two-dimensional graph whose nodes represent the sentences and the edges are labeled by statistical and semantic scores relating to the degree of similarity between each pair of sentences. In addition, we have integrated a redundancy elimination algorithm and a pre-processing phase (stemming) to further improve the performance of the proposed method. In the second contribution, we propose a deep learning-based approach for ATS. It consists in using the variational auto-encoder (VAE) as an unsupervised features learning technique in order to generate, for each sentence, an abstract representation in a concept space. This one is used to rank the text sentences according to their similarity with a query, and then, to extract the most relevant ones. Another proposed alternative is to integrate this abstract representation in order to compute the similarity between each pair of sentences by adopting the previous graph model. The proposed method allows, on the one hand, the dimensionality reduction, and on the other hand, the improvement of the extraction process of relevant sentences. In the third contribution, we adopt the distributed representation of words (Word2vec) as input for training several unsupervised neural networks models. The new obtained sentences representations are used to calculate the similarity between each pair of sentences in order to construct the previous graph. We also propose ensemble learning models to improve the quality of automatic summarization of Arabic texts. Finally, our fourth contribution consists of adopting the clustering technique to group texts into several clusters for which we identify the related topics space by using Latent Dirichlet Allocation method. Next, we use the text representation model in the identified topic space of each cluster as the training data of the unsupervised neural networks and ensemble learning models in order to learn new abstract representations. These new representations are used to rank the sentences of the text to be summarized according to the graph model. The performance of all the proposed approaches is evaluated using several datasets. The results obtained show the significance of our proposals.

Keywords: Automatic summarization, Arabic texts, Natural language processing, semantic features, word embedding, machine learning, neural networks, deep learning, ensemble learning.

Résumé

Ce travail de thèse s’inscrit dans le cadre du résumé automatique des textes en langue Arabe (RAT). Nous nous sommes particulièrement intéressés à l’amélioration des méthodes extractives en s'appuyant sur des approches statistique, sémantique et d'apprentissage automatique. Dans un premier temps, nous présentons un état de l’art concernant les principales méthodes de RAT et notamment celles dédiées à la langue Arabe. Ensuite, nous décrivons quatre contributions permettant d'améliorer la performance des méthodes existantes. Dans la première contribution, nous proposons une nouvelle méthode de RAT modélisant le texte sous forme de graphe bidimensionnel dont les nœuds représentent les phrases du texte et les arcs sont étiquetés par des scores statistique et sémantique relatifs au degré de similarité entre chaque paire de phrases. De plus, nous avons intégré un algorithme d'élimination de la redondance et une phase préalable de prétraitement (stemming) permettant d'améliorer davantage la performance de la méthode proposée. Dans la deuxième contribution, nous proposons une méthode de RAT basée sur l’apprentissage profond. Elle consiste à utiliser le variational auto-encoder (VAE) en tant que technique d'apprentissage non supervisée des caractéristiques afin de générer, pour chaque phrase, une représentation abstraite. Celle-ci est exploitée pour classer les phrases du texte selon la similarité par rapport à une requête et par la suite extraire celles les plus pertinentes. Une autre alternative proposée consiste à intégrer cette représentation dans le calcul de la similarité entre chaque paire de phrases en adoptant le modèle de graphe précédent (sans l'usage de la requête utilisateur). Cette méthode permet, d'une part, la réduction de la dimensionnalité, et d'autre part, l'amélioration du processus d'extraction des phrases pertinentes. Dans notre troisième contribution, nous adoptons la représentation distribuée des mots (Word2vec) comme entrée pour l’entrainement de plusieurs modèles de réseaux de neurones non supervisés. Les nouvelles représentations obtenues des phrases sont utilisées pour calculer la similarité entre les paires des phrases pour construire le graphe précédent. Nous proposons également des modèles d’apprentissage ensembliste pour améliorer la qualité des RAT Arabe. Enfin, notre quatrième contribution consiste à adopter les techniques de clustering pour regrouper les textes en plusieurs clusters pour lesquels nous identifions l’espace des sujets associés (thématiques) par la méthode d’allocation de Dirichlet latente. Ensuite, nous utilisons la représentation des textes de chaque cluster comme données d’entrainement des réseaux de neurones non supervisés et des techniques ensemblistes pour l’apprentissage de nouvelles représentations abstraites. Celles-ci sont exploitées pour classer les phrases du texte à résumer selon un modèle basé sur les graphes. L'ensemble des méthodes proposées est évalué en utilisant plusieurs corpus. Les résultats obtenus montrent l'intérêt de nos propos.

Mots-clés : Résumé automatique, textes Arabes, traitement automatique du langage naturel, sémantique, représentation distribuée, apprentissage automatique, réseaux de neurones, apprentissage profond, apprentissage ensembliste.

Contents

List of Figures

List of Tables

List of Abbreviations

AE	**A**uto-**E**ncoder
ATS	**A**utomatic **T**ext **S**ummarization
AWN	**A**rabic **W**ord**N**et
BOW	**B**ag-**O**f-**W**ords
BP	**B**ack-**P**ropagation
CBOW	**C**ontinuous **B**ag-**O**f-**W**ords
CNN	**C**onvolutional **N**eural **N**etwork
DAE	**D**eep **A**uto-**E**ncoder
DBN	**D**eep **B**elief **N**etworks
DUC	**D**ocument **U**nderstanding **C**onferences
EASC	**E**ssex **A**rabic **S**ummaries **C**orpus
ELM	**E**xtreme **L**earning **M**achine
ELM-AE	**E**xtreme **L**earning **M**achine **A**uto-**E**ncoder
HMM	**H**idden **M**arkov **M**odels
LDA	**L**atent **D**irichlet **A**llocation
LSA	**L**atent **S**emantic **A**nalysis
MMR	**M**aximal **M**arginal **R**elevance
NLP	**N**atural **L**anguage **P**rocessing
NMF	**N**on-negative **M**atrix **F**actorization
NN	**N**eural **N**etworks
PCA	**P**rinciple **C**omponent **A**nalysis
RBMs	**R**estricted **B**oltzmann **M**achines
Sentence2Vec	**S**entence to **V**ector
SKE	**S**ummarization and **K**ey-phrase **E**xtraction
SLFNs	**S**ingle-**L**ayer **F**eed **F**orward **N**etworks
SVD	**S**ingular **V**alue **D**ecomposition
SVM	**S**upport **V**ector **M**achine
TF	**T**erm **F**requency
TF-IDF	**T**erm **F**requency – **I**nverse **D**ocument **F**requency
TF-ISF	**T**erm **F**requency – **I**nverse **S**entence **F**requency
TS	**T**ext **S**ummarization
VAE	**V**ariational **A**uto-**E**ncoder
WE	**W**ord **E**mbedding
WN	**W**ord**N**et
Word2Vec	**W**ord **T**o **V**ector

General introduction

Context and Motivation

According to the dictionary, a summary is defined as a short description that gives the main facts or ideas about something. According to Mani and Maybury (1999), text summarization is "the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks)". Hovy (2005) defined a summary as "a text that is produced from one or more texts which contains a significant portion of the information in the original text(s), and that is no longer than half of the original text(s)". In addition to a text document, automatic summarization can be applied to all kinds of media such as speech, multimedia documents, hypertext, video or even any combination of these types.

With the rapid growth of the Internet, and the multiplicity of media mass storage, the amount of electronic documents and textual data became huge. Since humans cannot handle large text volumes manually, they seek to save time and reduce the cost by the help of automatic analysis methods. Such methods should avoid the need to look in the whole document in order to decide whether it is of interest or not, by finding the most relevant information quickly. There is no time for user to read the entire document to make critical decisions quickly. The human, unable to manually handle large text volumes, must find automatic analysis methods adapted to automatic processing of personal data. These methods fall into the area of automatic natural language processing (NLP). Among the most popular applications include machine translation, automatic summarizer, information retrieval, text mining, spell correction, speech synthesis, speech recognition or handwriting recognition.

In this thesis work, we are interested in the field of automatic text summarization with a focus on Arabic documents. Text summarization is a challenging task in the natural language processing area (NLP) to fix information content overload issue, and the vast amount of online information. It seeks to facilitate the task of reading and searching information in large documents by generating reduced ones with no loss of meaning.

Automatic summarization can be used to improve other natural language processing tasks such as clustering, classification, indexing, keywords extraction and so on. The first attempt at automatic summarization of texts is started in the late fifties with Luhn (1958). Thus, the need of automatic summarization systems is gradually being felt due to reasons of cost savings that may result from this automation. To date, ATS is a dynamic field with many challenging issues.

Research in Arabic Text summarization is still in its early beginning and the literature that addresses this area in Arabic is fairly small and only recently compared to that on Anglo-Saxon, roman and other Asian languages. Moreover, summarization systems for Arabic have not reached the same level of maturity and reliability as those for English, for instance.

It is worth mentioning that Arabic is the mother tongue of all Arabic countries with a very fast growth pace on the web, given that the number of internet users increased by more than 8000%

between 2000 and 2017 with more than 219 million of Arabic users[1]. Arabic is one of the five most spoken languages in the world and is also in the fourth position in terms of those used in the internet, after English, Chinese and Spanish. However, research on Arabic NLP is still embryonic because Arabic is not given equal attention as other languages. Therefore, the need to develop systems for the processing and summarization of electronic Arabic documents is growing significantly.

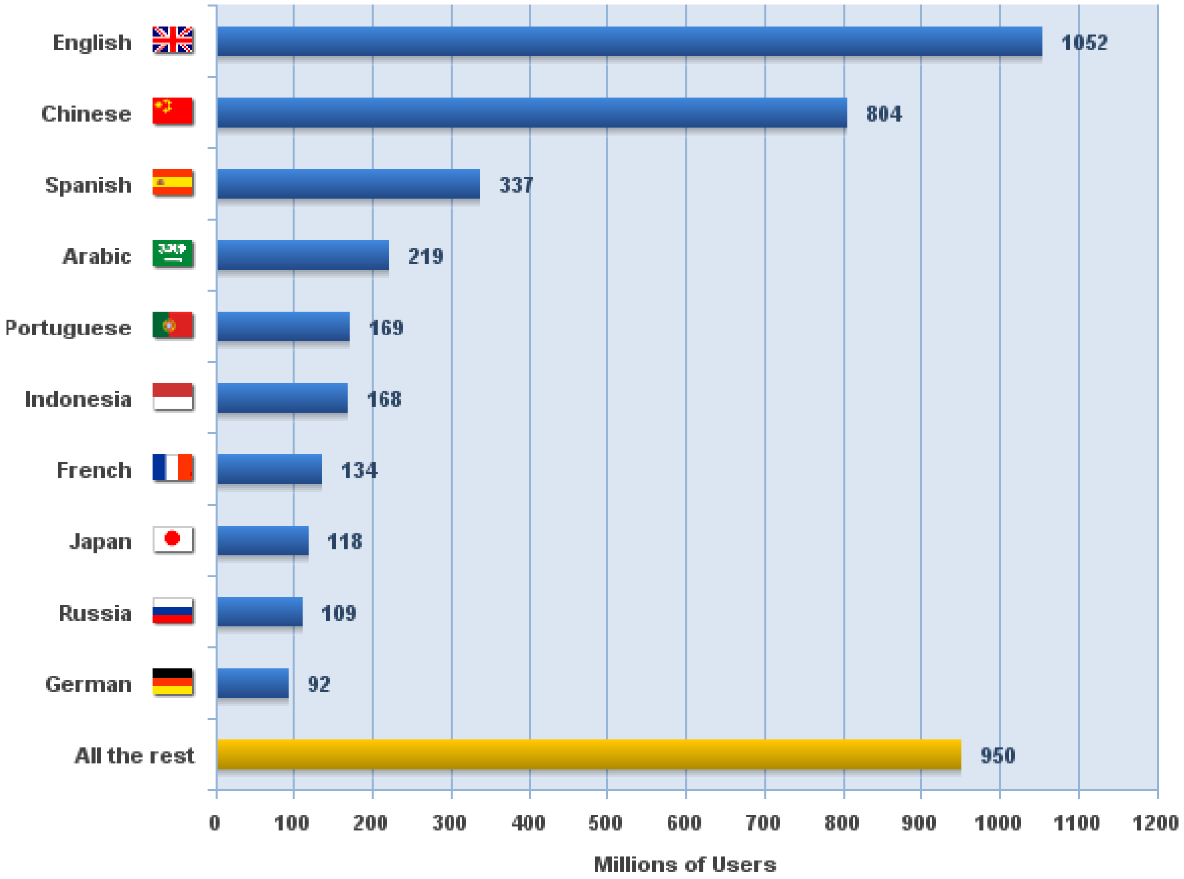


Figure 1. Top ten languages in the internet in Millions of users – December 2017

In this thesis work, we investigate existing works and approaches proposed for ATS in order to understand the limitations and weaknesses of existing systems developed for summarizing Arabic documents. We improve the state of the art by proposing innovative approaches that deal with several detected problems. The obtained results are very satisfactory and provide evidence that the proposed approaches are effective at increasing the performance of Arabic text summarization.

Aims and Contributions of this thesis

This thesis work focuses on the field of automatic text summarization, specifically, the summarization of Arabic documents. According to the existing literature, many problems have

[1] https://www.internetworldstats.com/stats7.htm

been raised and addressed. Firstly, sentence ranking is a main step in any summarization system. Our goal is to ameliorate this process by proposing a new approach based on graph theory. Secondly, in sentence selection phase, redundancy is a critical problem due to the fact that sentences with similar meaning can be included in the summary. Therefore, instead of typically selecting top ranked sentences, we use a specific algorithm to form the final extractive summary without redundancy. Thirdly, we integrate the semantic analysis in the summarization task by introducing an external man-developed knowledge database system to accurately represent the meaning of documents. Fourthly, document representation is an important phase to determine the accuracy of any ATS system. In this work, we address the document representation issue by adopting a distributed representation based on deep learning and neural networks algorithms with word embedding approach instead of the traditional bag-of-words (BOW) representation, which is sparse and do not consider the semantic relationships among textual units. Fifthly, Arabic text summarization task suffers from a shortage of labeled data used in training supervised models. For this purpose, we have adopted the unsupervised feature learning models since unlabeled data are heavily available. The main contributions we made in this thesis are the following:

- Establishing and analyzing a state-of-the-art concerning the field of automatic text summarization. We have focused our study on Arabic text summarization. As a result, we highlighted some limitations and weaknesses that are not addressed in Arabic.
- Improving the Arabic text summarization task by proposing a new approaches that deal with sentences ranking and selection issues.
- Proposing a new graph-based summarization model which integrates statistical and semantic analysis of Arabic documents.
- Adopting a Variational Auto-encoder as an unsupervised deep learning model in order to learn features from an input corpus and produce a good summary according to these features.
- Proposing several unsupervised neural network models with ensemble learning techniques in order to improve the summarization results.
- Adopting the distributed representation based on word embedding in order to build a relevant representation of the input document
- Adopting the clustering technique and topic identification in order to improve documents representation before applying the summarization model.

Outline of the thesis

This thesis is divided into five chapters in addition to the general introduction, conclusion and perspectives. A detailed French summary is also added in the end of this document.

Chapter 1 presents a detailed review of different automatic summarization methods and techniques. It also give a detailed background and challenges in Arabic NLP. Subsequently, the chapter illustrates a deep study on exiting works on Arabic TS with the focus on advantages and limitations of each approach.

Chapter 2 presents our first contribution, which consist of a two-dimensional graph model that makes uses of statistical and semantic analysis. It describes how the proposed approach rank the document sentences and addresses the redundancy and information diversity issues. It provides the results obtained by various experimentations. The chapter also studies the effect of the preprocessing phase in the performance of Arabic text summarization.

Chapter 3 describes our second contribution, which consists of a new method for Arabic text summarization using unsupervised deep learning model. It presents how the variationnal auto-encoder learns unsupervised features from a high-dimensional input data and how to use these new features in the summarization task. It explores several input representations such as term frequency (TF), TF.IDF and both local and global vocabularies. All sentences are ranked according to the latent representation produced by the VAE. It investigates the impact of using VAE with two summarization approaches, graph-based and query-based approaches. The experimental studies confirm that the proposed method leads to better performance than most of the state-of-the-art extractive summarization approaches for both graph-based and query-based summarization approaches.

Chapter 4, which consists of our third contribution, investigates in detail the use of several unsupervised deep neural network models in ATS. It describes how to enhance the quality of ATS by adopting ensemble learning techniques that aggregate the information provided from different sources. It provides a detailed experimentations to evaluate the performance of the investigated models on two kind of datasets (Arabic and English). Results of statistical studies affirm that word embedding-based models outperform the summarization task compared to those based on BOW approach. In particular, ensemble learning technique with Word2Vec representation surpass all the investigated models.

Chapter 5 presents our fourth contribution. It describes in detail our proposed approaches for summarization a large Arabic documents. It provides the foundation of our approaches, which consists of clustering the large dataset and identifying topics of each cluster. Next, it describes how to build a document representation model based on the identified topic space. Then it presents how to use this representation to learn the abstract features using unsupervised neural networks algorithms and ensemble learning models. It also presents the evaluation and results through several experiments and conclude that the proposed models enhance the Arabic summarization task.

Conclusion and perspectives summarizes our contributions in this thesis work with pointers to future works.

Chapter 1

State-of-the-art in Automatic Text Summarization

1.1 Introduction

The number and size of electronic documents available in the web have become huge due to the growth of internet social media and user-created content. The human, unable to manually handle large text volumes, must find automatic analysis methods adapted to automatic processing of personal data. These methods fall into the domain of automatic natural language processing (NLP). In this context, and in order to analyze this massive generated data, many NLP applications are needed. Among the most popular applications include machine translation, automatic summarizer, information retrieval, text mining, spell correction, speech synthesis, speech recognition or handwriting recognition. In particular, Automatic text summarization (ATS) is an increasingly growing and challenging task in NLP area, whose goal is the production of a shortened version of a large text document, while preserving the main idea existing in the original document.

Our work focuses on automatic text summarization which allows user to decide whether the document is of interest or not, without looking at the whole document by extracting brief information from a given text. There is no time for user to read the entire document to make critical decisions quickly. Thus, the need for automatic summarization software is gradually being felt due to reasons of cost savings that may result from this automation.

Automatic text summarization is the process that produce from a source text one that smaller and contains the most relevant information in the text. The aim is to generate a reduced representation of one or more documents without losing the meaning and important information in the original text and it does not include redundant information.

Text summarization has experienced a great development in recent years, and a wide range of techniques and paradigms have been proposed to increase researches in this field. Researches in automatic text summarization have become a new challenges because the new emerging technologies. So it is essential to analyze its progress and present the most important techniques made on this field. In this chapter, we present an investigation on automatic text summarization research works by focusing on Arabic text summarization.

1.2 Types of automatic text summarization

Summarizing a text by human consists of reading, understanding and providing the most important ideas in an orderly and coherent manner. Even for humans, this task becomes very complicated and would take a lot of time and work, especially when the size of the document is large. Thus, there is a vital need for having an automatic system that can perform the task in an efficient and expedient manner. Luhn (1958) was the first to tackle the automatic summarization of texts, which is, to date, a dynamic field with many challenging issues.

As shown in Figure 1.1, there are various categories of text summarization. The most important are single and multi-document summarization. Both of them can use extractive or abstractive approaches. Single document summarization can be used to generate summary of multiple documents since redundancy, which is the one of the biggest problems of this category, is addressed.

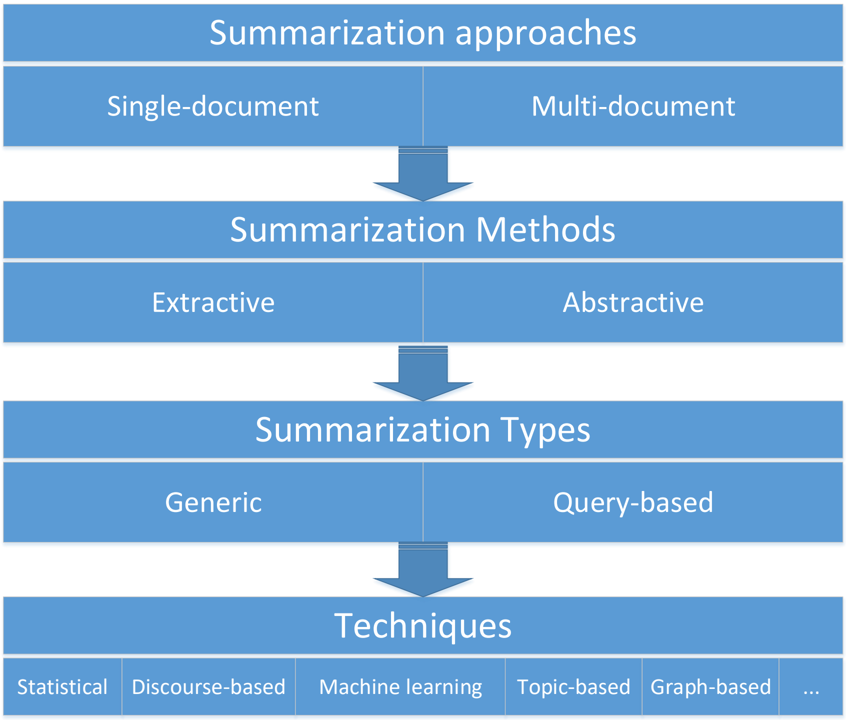


Figure 1.1. Categories of ATS

Text summarization approaches are divided into two major categories: extractive and abstractive. The former methods, known as sentence ranking, consist of ranking and extracting key sentences from the input document and presenting them in the same order as a summary. Deciding on the relevance of sentences rests on the weight of each one as accounted for by statistical and linguistic features. Abstraction, however, involves rephrasing the most relevant parts of the source document, that is to say, digesting the major concepts in the initial text and presenting them in a shorter document. Linguistic and statistical methods, in addition to human knowledge, are prerequisites in this respect. Whereas abstractive summarization needs heavy machinery for language generation and is not easy to implement or stretch to larger domains, simple extraction of sentences has yielded positive results in large-scale applications, namely in multi-document summarization. In this thesis work, we have adopted the extractive summarization approach. An illustration of this approach is given in Figure 1.2.

Text summarization can also be either generic or query-oriented. Generic summarization provides general informations presented bin the document whereas query-base summarization produces informations needed by the user by retrieving sentences that match a query provided by the user.

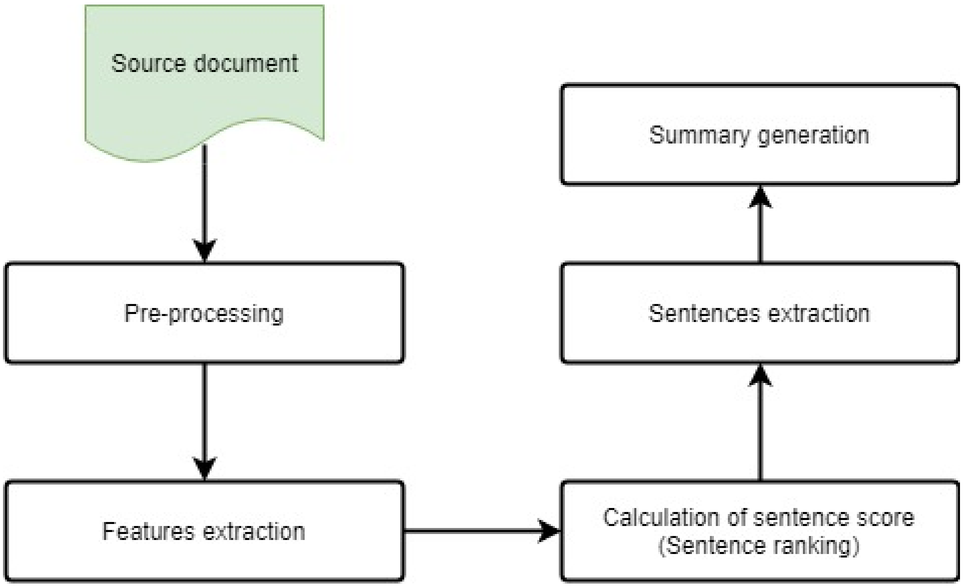


Figure 1.2. Flowchart of a typical extractive automatic text summarization system

Several techniques and methods have been introduced in order to address extractive approaches for both single and multi-document summarization. This shows how important is the summarization task, which is a growing research field in NLP area given the significant number of overlapping approaches and techniques. The following sections will discuss in detail various classifications of summarization approaches with a focus on Arabic.

According to the surveyed state-of-the-art carried out by Kanapala et al. (2017), automatic text summarization is classified into two major techniques depending on the number of summarized documents: mono-document summarization and multi-document summarization.

1.3 Single document summarization

Single document summarization consists of having a quick overview of the most important information existing in a single document. Generally speaking, extractive summarization of single document uses sentences ranking and extraction to identify the most important content in the document. Several approaches have been proposed to deal with this kind of summaries. Table 1.1 gives an overview of some works in single-document summarization identified by Kanapala et al. (2017). They are regrouped in five kinds of approaches: Linguistic feature based approaches, Statistical feature based approaches, Language-independent approaches, Evolutionary computing based approaches and Graph-based approaches.

By analyzing these works, we can conclude the following: First, According to this study, several evaluation datasets and measures have been used by the authors. Second, the DUC2002 corpus and Rouge-1 and Rouge-2 measures have been used by the majority of these works in the evaluation process. After comparing the results of Ghalehtaki et al. (2014), Mendoza et al. (2014), GarcíaHernández and Ledeneva (2013), Abuobieda et al. (2013), Vodolazova et al.

(2013), Wang and Ma (2013), obtained on the DOC2002 dataset, it has been shown that the method proposed by Abuobieda et al. (2013) outperformed other approaches.

Table 1.1 Single document summarization works

Reference	Technique
Linguistic feature based approaches	
Wang and Ma (2013)	LSA
Gong and Liu (2001)	LSA
Pal and Saha (2014)	Lesk algorithm and WordNet
Statistical feature based approaches	
Vodolazova et al. (2013)	Sentence scoring using statistical and semantic features: - Term frequency - Inverse term frequencies - inverse sentence frequencies - Word sense disambiguation - Anaphora resolution - Textual entailment
Ferreira et al. (2013b)	15 Sentence scoring methods: word frequency, TF/IDF, sentence position, sentence length,… etc.
Sharma and Deep (2014)	- Sentence scoring using: Term frequency, Font semantics, Proper nouns and Signal words
Batcha et al. (2013)	- Conditional Random Field - Nonnegative Matrix Factorization
Language-independent approaches	
Cabral et al. (2014)	- Language identification with CALIM - Translation into English with Microsoft API - Sentence scoring using word frequency, sentence length and sentence position
Gupta (2014)	- language independent features - Sentence scoring using seven features like, the similarity between words and title, n-gram similarity with title,… etc.
Evolutionary computing based approaches	
Abuobieda et al. (2013)	- Optimized sentence clustering with Differential Evolution algorithm - Five features for text clustering: title, sentence length, sentence position, presence of numeric data and thematic words
García-Hernández and Ledeneva (2013)	- Genetic Algorithm
Mendoza et al. (2014)	- Genetic Operators - Guided Local Search
Ghalehtaki et al. (2014)	- Redundancy Reduction with Cellular Learning Automata - Scoring sentences with Particle swarm optimization and Fuzzy logic
Graph based approaches	
Hirao et al. (2013)	- Tree Knapsack Problem - RST - Dependency-based discourse tree - Integer linear programming
Kikuchi et al. (2014)	- Nested tree - RST - Combinatorial optimization
Ferreira et al. (2013a)	- Four-dimensional graph - Statistical similarity: content overlap between two sentences - Semantic similarity from WordNet - Co-reference resolution - Discourse relation - PageRank to rank sentences
Ledeneva et al. (2014)	- Maximal Frequent Sequences to represent the nodes in the graph - PageRank to rank sentences

Third, when the summarization is performed on the news dataset, it has been shown that the method proposed by Ferreira et al. (2013b) performed better in comparison with those proposed by Ferreira et al. (2013a), Cabral et al. (2014), Ledeneva et al. (2014), Pal and Saha (2014). Fourth, Kikuchi et al. (2014) and Hirao et al. (2013) used the RST-DTB dataset to evaluate their methods. The best performance was achieved by Kikuchi et al. (2014), which used the nested tree method.

1.4 Multi-document summarization

Multi-document summarization consists of generating a summary from several text documents. Redundant information is a key problem in this kind of summaries while multiple document can have the same information that may be included in the summary. Thus, removing these redundant information from the summary allows to diversify information exposed to the user, and therefore, improve the quality of the multi-document summarization task.

Table 1.2 Multi-document summarization works

Reference	Technique
Linguistic feature based approaches	
Chen and Zhuge (2014)	- Clustering by detecting common facts - Removing redundancy
Gross et al. (2014)	- Association Mixture
Ma and Wu (2014)	- Sentences are ranked by combining the following features: ○ n-gram ○ Dependency word pair ○ Co-occurrence ○ TF.IDF - Cosine similarity is used to detect duplicate sentences
Evolutionary computing based approaches	
Lee et al. (2013)	- Topic model with LDA for scoring sentences - Fuzzy method for extracting important sentences
Kumar et al. (2014)	- Using gazetteer list and named entity recognition to extract component sentences - Genetic-Case Base Reasoning to identify Cross document relationships - Fuzzy reasoning for sentence scoring
Graph based approaches	
Samei et al. (2014)	- Minimum distortion measure to compute the semantic similarity between two sentences - Graph model to represent sentences by node and their semantic relationships by edges - PageRank algorithm to rank sentences and select the important among them.
Ferreira et al. (2014)	- Text representation using four-dimensional graph model based: ○ Statistical similarity ○ Semantic similarity ○ Co-reference resolution ○ Discourse relations - PageRank algorithm for sentence ranking - Removing redundancies with sentence clustering

Several works have been carried out to meet this challenge. They are regrouped in three kind of approaches: Linguistic feature based approaches, Evolutionary computing based approaches and Graph-based approaches. Table 1.2 illustrates some of these works surveyed by Kanapala et al. (2017), which show that most of the surveyed works used a redundancy removal technique to eliminate duplicate sentences from the final summary. Among the approaches evaluated on the DUC2002 dataset, such as, Kumar et al. (2014), Ferreira et al. (2014) and Samei et al. (2014), we found that Kumar et al. (2014) achieved better performance. This method is based on both Genetic-Case Base Reasoning to identify cross-document relationships and Fuzzy reasoning model to score sentences. In addition, Gross et al. (2014) obtained the best results compared to Lee et al. (2013) and Ma and Wu (2014), which are evaluated on news dataset.

1.5 Extractive Text Summarization Approaches

1.5.1 Statistical-based approaches

Statistical approaches rely on surface level features extracted from the input text in order to have relevant information. They are based essentially on the calculation of a score associated with each sentence in order to estimate its importance in the text. The final summary will keep only the sentences with the highest scores. These traditional approaches are among the first that were applied in automatic summarization task. These technique do not need any additional linguistic resources or complex processing tools. Early works (Luhn, 1958; Edmundson, 1969) have studied the following techniques:

1.5.1.1 Word frequency

This method is considered among the first methods investigated in the field of automatic summarization. It was developed by Luhn in 1958 (Luhn, 1958). It is based on the fact that the author express his key ideas using a few key-words that tend to be recurrent in the document.

Indeed, this suggestion is based on the assumption that an author generally focuses on a subject topic by repeating certain words relating to it. Words with high frequency are therefore indicative of the content of the document and are considered more representative of the meaning. Thus, important sentences are those which contain the most frequency words used in the text. The weight of the sentence is calculated by summing the frequency of its words.

$$Score(S) = \sum_{w \in S} tf(w) \tag{1.1}$$

Where S is the sentence, and $tf(w)$ is the frequency of term w_i (number of occurrence of term w in the document).

We can use some predefined key-words instead of considering all the words in the document. The importance of each sentence is calculated depending on the keywords it contains. The following formulation is used:

$$Score_{keywords}(S) = \sum_{w \in S} c(w_i) * tf(w) \tag{1.2}$$

Where:

- $c(w) = \begin{cases} A \ if \ w \in keywords \ list \ (A > 1) \\ 1 \ otherwise \end{cases}$
- $tf(w)$ is the frequency of the word w in the sentence S

The keywords list can be introduced by the user (depending on the domain of interest) or composed of the keywords established by the author. The importance of the weight of the term w is given by $A * tf(w), A > 1$.

$TF.IDF$ feature defined by is a numerical feature that is widely used in information retrieval and text mining applications. It reflects how important a word is to a document in a collection or corpus. It expresses that a word is more important if it is both frequent in the analyzed document and less frequent in the documents corpus. $TF.IDF$ has been used in many automatic summarization systems (Savyanavar and Mehta, 2016). It is calculated by the following formula:

$$TF.IDF(w) = TF * IDF(w) \tag{1.3}$$

$$IDF(w) = log\frac{N}{df_w} \tag{1.4}$$

Where N represents the total number of documents in the corpus and df_w is the number of documents where the word w appears.

1.5.1.2 Sentence position

This method has been introduced by Edmundson in 1969 (Edmundson 1969) in order to complete the method of terms distribution which he called "key method". It is used in combination with other weighting methods to increase or decrease the weight of a sentence. In this respect, this method assumes that the position of a sentence in a text indicates its importance. The first and last sentences of a paragraph, for example, may convey the main idea and should therefore be part of the summary.

Sentence location within a document is exploited by many summarization systems. Depending on the studied domain, document structure is usually considered while scoring sentences. The disadvantage of this method is that it depends on the nature of the text to be summarized as well as the style of the author. For example, this method is effective for summarizing newspaper articles, since important sentences tend to appear in the first of the article.

1.5.1.3 Similarity with title or query

Since the title is the most meaningful expression that summarizes the document in candidate few words, we can say that the sentences which are most similar to the title or the query (if given) are the most significant of the document, because the main idea are covered in general in the document title. In this case, we consider the title/query words as the indexing keywords. The candidate terms are selected from the title and subtitles of the document. Edmundson (1968) proposed to assign an important weight to the full words of the title as well as for the full words of the titles and subtitles of all sections.

1.5.1.4 Cue words and phrases

A sentence can be considered as important if it contains some indicative expressions. In this way, Edmundson (1969) defined the bonus and stigma terms for weighting the sentences. The presence of bonus phrases such as “we confirm” or “this paper presents” in a sentence indicates its importance because the author in this case announces the general idea of the document, and therefore he increases the score of the sentence. In contrast, stigma expressions such as “for example”, “impossible” or “hardly” contain mainly anaphors that indicate the irrelevance of a sentence and, therefore, penalize its weight giving it a higher chance for being excluded from the output summary. The final weight of each sentence is calculated by summing the weights of these expression found among the words that constitute it.

1.5.2 Cluster-based approaches

Clustering is the process of assigning a set of observations into separate subsets (called clusters). It has gained much attention in the past years and adopted by many works in order to improve the quality of Information Retrieval and automatic summarization tasks. Clustering can be performed on words, sentences or documents.

Documents are generally written in a way to address the various subjects one after the other and in an organized manner. They are normally divided into sections, paragraphs and sentences. This organization also applies to the automatic summarization, it should intuitively be thought that the automatic summarization must approach the different topics appeared in a document (or a set of documents). A simply way to generate a summary consists of assigning each sentence to a specific cluster and then selecting one representative sentence for each cluster.

Other automatic summarization systems use clusters to generate a significant summary approaching the various topics of the document. The documents are represented using term frequency-inverse document frequency (TF-IDF). In this context the term frequency (TF) is the average number of occurrence (by document) in the cluster. The system takes as input documents already gathered in clusters, thus each cluster is regarded as a topic. The topic is represented by words of which the value TF-IDF is higher in the cluster. Let us point out that, according to the analyzed corpus, a document can be a text, a paragraph or even a sentence. The selection of the relevant sentences is based on the similarity of the sentences with the topic of cluster.

Zhang and Li (2009) proposed a method of automatic summarization based on the sentences cluster. In this work, similar sentences are gathered in the same cluster. The measurement of similarity between the sentences is based on three similarities: a) Similarity of the words between two sentences; b) Similarity of the order of the words; c) Semantic similarity of words. At the end the similarity between two sentences is calculated by combining the three measurements of similarities mentioned above. Once the calculation of similarity between the sentences is made and the number of cluster of sentences is determined, the K-means method is used to gather the sentences of the document in the clusters.

Another work was published in the same direction, in Ledeneva et al. (2011), the authors proposed a summarization method by sentence clustering using the following steps: terms selection, terms weighting and sentences selection. However, before these steps, the document is analyzed and preprocessed for removing stop words (words without meaning) and applying the stemming algorithm. In the first step, one of the three models of the text is extracted: Bag-of-words model, n-grams model and Maximal frequent Sequence (MFS) model. In the second stage, the terms are weighted by using the Boolean method, TF, IDF or TF-IDF method. In the third steps, the Expectation-maximization algorithm, (often shortened EM) is used to form similar groups of sentences in order to obtain a sentence representing each cluster to be included in the summary.

Heu et al. (2015) proposed a multi-document summarization based on folksonomy system that employs tag clusters generated by a well-known Flickr application in order to detect important sentences from a documents set. After the preprocessing step, uses a HITS algorithm to discover the semantic relationships between words with the help of tag clusters from Flickr. Each sentence is then scored according to the importance of its words and the semantic relatedness to words in other sentences.

1.5.3 Topic-based approaches

Topics identification consists of detecting the main subjects covered by the text. The intuition behind topic-based summarization is that a good summary should cover topics that are of importance to the user. In this context, Teng et al. (2008) proposed an approach, which combines the automatic topics identification technique (identification of subjects covered) with the terms frequency method. This methodology consists of calculating initially the similarity between the sentences, then carry out the identification of the subject covered by gathering similar sentences in clusters. In a second stage, and based on terms frequency, the projecting sentences are selected starting from the local topics already identified.

In another study carried out by Kuo and Chen (2008), not only the frequency of terms is used to detect relevant information in the text, the authors also use informative words and event-driven words to produce automatic summarization. This type of words indicates concepts and the important relations which are used to detect important sentences in the text.

Fang et al. (2015) developed a topic aspect-oriented summarization. They used various features that describe different topics. This approach is used for text as well as image summarization. For document summarization, the authors extracted three features: sentence length, sentence position and word frequency. In order to generate the summary, the greedy algorithm is implemented considering the coverage and diversity issues.

1.5.4 Approaches Based on Lexical Chains

The automatic text summarization by lexical chains was introduced by Barzilay and Elhadad (1997). This method uses the WordNet knowledge database to identify the relations of cohesion between terms (i.e., repetition, synonym, antonym, hyperonyme, and homonymy) then composes chains based on these terms. Scores are given based on the number and type of relation in the chains. The final summary contains the sentences where the strongest chains are very concentrated.

A similar method was presented by Pourvali and Abadeh (2012). It is based on the lexical chains and the graphs using the knowledge bases of WordNet and Wikipedia**.** This method consists initially in finding the exact meaning of each word in the text (Word Sense Disambiguation), then builds the lexical chains and removes those which have a weak score compared with the others. The structure of the lexical cohesion of the text can be exploited to determine the importance of a sentence. To build the lexical chains of a text, all the words with their meanings must be known. For that, the authors used WordNet to solve the lexical and semantic ambiguity of the word.

1.5.5 Discourse-based Approaches

New techniques were born to solve the problem of automatic summarization based on statistical approaches; these techniques are based on the analysis of the discourse and its structure. Discourse-based approaches are the linguistic techniques used to discover connections between textual units, i.e. sentences, sections and paragraphs. Indeed, the discourse-based approach is generally based on a formal representation of the knowledge contained in the documents. They assume that the structure and coherence of a text can be modeled through rhetorical relationships.

Among these techniques we quote the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). RST has two main aspects: (i) coherence, which means that the text is composed by units that are connected together by rhetorical relations such as explanations, causes and elaborations, (ii) cohesion, which means that various fragments of the text have relationships between them. The main challenges in discourse-based ATS are coherence and cohesion, which are processed by advanced linguistic analysis tools.

Khan et al. (2005) combined the RST with a generic summarizer to add linguistic knowledge to the process of automatic summarization. But this mixed approach could not improve the results obtained by the generic summarizer. In other words, the disadvantage of this approach is found at the analyzer level which could not detect all RST relations, in fact, a good analysis and languages knowledge could have improved the output of the summary system.

In Li Chengcheng (2010), the RST was also proposed as an element for the automatic text summarization. The system extracts the rhetorical structure of the text and the components of the rhetorical relations between the sentences, then calculates the weight of each sentence of the text according to its utility and removes the least important parts of the structure having a weak weight.

1.5.6 Graph-based approaches

In graph-based summarization, textual units and the relations between them are represented in the form of undirected graphs. Each unit is represented by a node in the graph. The relation between different units is modeled by an edge between them. Many kinds of relations can be considered, such as, number of common words or cosine similarity.

LexRank and TextRank are the most important methods proposed for graph-based automatic summarization. TextRank (Mihalcea and Tarau, 2004) is a graph-based ranking model used for both automatic text summarization and key-words extraction. It is based on PageRank (Brin and Page, 1998) algorithm in order to rank the graph elements that better describe the text. In the summarization task, each sentence is represented by a node in the graph and the edge between two nodes represents the similarity relation that is measured as a content overlap between the given sentences. The weight of each edge indicates the importance of a relationship. Sentences are ranked based on their scores and those that have very high score are chosen. LexRank is another automatic summarization system which is identical to TextRank. Both of them use graph-based approach for text summarization and the only difference between them is that the similarity measure used by TextRank is based on the number of similar words shared between the two sentences, while LexRank uses cosine similarity measure of TF-IDF vectors.

In the same context, Thakkar et al. (2010) proposed a method based on graphs algorithm for automatic texts summarization. This method consists in building a graph from the text. Nodes of the graph are represented by the text sentences, for each sentence there is a node. The edge of the graph represent connection (lexical or semantic) between the sentences, this connection is evaluated by calculating the similarity between the sentences. This similarity is evaluated if the two sentences share some common words either in terms of lexical symbols or the more similar words. The weight of each node is calculated by using the cosine similarity function. After that, the summary is made up by taking the shortest way which starts with the first sentence of the original text and finishes with the last sentence.

In Baralis et al. (2013), a new graph-based summarizer named GRAPHSUM has been proposed. The authors use association rules discovered in the text in order to represent the correlations among several terms by a graph model. Other graph-based text summarizations have been proposed such as SUMGRAPH (Patil and Brazdil, 2007) and Time stamped Graph (Lin, 2007).

1.5.7 Latent Semantic Analysis-based Approaches (LSA)

Among the most recent methods in the automatic texts summarization, we quote the use of Latent semantic analysis (LSA) method. LSA is an algebraic-statistical method that extracts and represents semantic knowledge of the text based on the observation of the co-occurrence of words (meaning of words). This technique aims to build a semantic space with a very large dimension from the statistical analysis of the whole co-occurrences in a corpus of texts. It was introduced by Deerwester et al. (1990) and has been adopted in several NLP applications such as document classification (Yu et al., 2008), clustering (Song and Park, 2007) and ATS (Yeh et al., 2005; Ozsoy et al., 2011; Mashechkin et al., 2011).

The starting point of the latent semantic analysis consists of a lexical table which contains the number of occurrences of each word in each document, which can be a text, paragraph or even a sentence. The idea is that two words are considered close if they occur in similar contexts and two contexts are similar if they contain similar words.

Gong and Liu (2001) proposed an automatic summarization system of news text with the use of LSA as a way to identify the important topics in the documents without using lexical resources like WordNet. The latent semantics analysis is done in two steps. Initially, the matrix of occurrences is built, this consists to represent the documents in the form of matrix A where columns are sentences and rows represent the words/phrases. The elements *(i, j)* of the matrix (a_{ij}) corresponds to the number of occurrence (the weight) of word *i* in the sentence *j*. if the sentence does not contain the word, the weight is equal to zero, otherwise the weight is calculate using the formula TF.IDF of the word. The next step consists to reduce the dimensions of the matrix; this reduction is achieved by the singular value decomposition method (SVD). In this way, the SVD is applied to matrix A to decompose into three new matrices as follows: $A = UWV^T$. The suggested that the row of the matrix V^T can be considered as various topics (subjects) covered in the original text, while each column represents a sentence in the document. And finally, in order to produce an extractive summary, they consider each row of matrix V^T

consecutively, and select the sentence with the highest value, until the summary with the desired size is reached.

In Yeh et al. (2005) another automatic text summarization method using LSA was proposed. It is a mixed approach between graphs based method and LSA based method. After using the SVD on a matrix of words per sentence and reduction of these dimensions, they build the corresponding matrix $A'=U'\Sigma'V'^{T}$. Each column of A' denotes the sentence semantic representation. These semantic representations of sentences are then used, instead of an occurrence frequency vector of keyword, in order to represent document as a graph of relations between sentences. A ranking algorithm (graph) is then applied to the resulting graph.

In the same context, a Non-negative Matrix Factorization (NMF) algorithm was proposed in (Mashechkin et al., 2011**)**, instead of the SVD, to reduce the dimensions of the matrix (matrix factorization). The idea is that from the matrix *A* whose columns are the *n* sentences of text and rows are the *m* terms, and since the elements of *A* are non-negative, so NMF can then decompose the matrix *A* into two positive matrices W_k and H_k. in order to approximate the matrix *A* in the decomposition form $A_k \approx W_kH_k$. Matrices W_k correspond to the mapping of space of *k* topics and the space of *m* termes, and H_k correspond correspond to the representation of the sentences in the space of topics. Subsequently, we can find out what the terms of the text best characterize each topics (subject) associated with the columns of the matrix W_k. After this decomposition, and based on this representation, a method for selecting the most important sentences is applied.

1.5.8 Fuzzy Logic-based Approaches

In (Kyoomarsi et al., 2008; Suanmali et al., 2009a; Suanmali et al., 2009b), another approach to automatic summarization based on a fuzzy logic has been proposed. This technique takes into account every feature of the text such as word frequency, similarity to keywords, similarity to the title words, sentences position, statistics of co-occurrence of lexical chain, indicative expression etc. After extracting these features and depending on the results, a value of 0-1 is assigned to each sentence of the text according to the characteristics of sentences and rules available in the knowledge base. The value obtained at the output determines the degree of importance of the sentence in the final summary. The membership function is divided into three functions which are composed of the following values: (Low, L), (Very Low, VL), (Medium, M), (High H) and very high value (Very High, VH). The important sentences are then extracted using fuzzy rules IF-THEN based on the criteria of text features. A sample of IF-THEN rule is given below:

IF (Title features is VH) and (SentenceLength is H) and (Term weight is VH)
and (SentencePosition is H) and (SentenceSimilarity is VH)
and (Word similarity is H) and (cue-phrase is VH)
and (TermFreq is H)
THEN (Sentence is important)

The design of such system is generally based on fuzzy rules and membership function. There is various membership function types used in fuzzy logic such as: sigmoid, Gaussian, trapezoidal (trapmf), triangular (trimf), etc. (Hannah et al., 2011). Choosing the right rules and appropriate membership function directly impacts the performance of the system. The architecture of a fuzzy system consists of four levels: Fuzzifier, inference engine, defuzzifier and the Fuzzy knowledge base. In the fuzzifier, the input feature values are converted to linguistic values (very low, low, medium, high and very high) using the membership function. Linguistic value denotes a fuzzy set to which a characteristic of a given sentence belongs. Then, the inference engine refers to the database containing the fuzzy IF-THEN rules to derive linguistic values. Thus, it compares the fuzzy input obtained from the fuzzifier with the knowledge base / rule and decides the importance of a sentence. The output of the inference engine is one of the linguistic values of the set of the membership functions values. In the last step, the linguistic output variables of the inference engine are converted to digital net values by defuzzifier using the membership function to represent the final score of the sentence.

The same concept was used in Hannah et al. (2011). Different characteristics of each sentence were taken into account, such as title words, sentence length, term weight, Sentence to sentence similarity, etc. the values of these features are used by the inference engine to generate the score of each sentence of the text.

1.5.9 Machine Learning-based Approaches

Machine learning allows to acquire and develop a new knowledge from the training data. Several approaches using machine learning have been adopted for text summarization. They are classified into three categories: supervised, unsupervised and semi-supervised. Supervised approaches need a collection of documents and their human-generated summarizes in order to learn new features of the textual units. Supervised summarization systems flag each sentence in the training dataset by two values (or classes): 0, which means that the sentence is not a part of the summary; or 1, if the sentence is a part of the summary. This classification is carried out based on the training data (pairs of documents / summarizes). Unsupervised approaches is a machine learning task of inferring a function from unlabeled data. While semi-supervised approaches use both labeled and unlabeled data to improve the learning performance.

There is a wide range of machine learning techniques for automatic text summarization. The first methods used are based on the binary classifiers (Kupiec et al., 1995), Hidden Markov Model (Conroy and O’leary, 2001; Schlesinger et al., 2008) and the Bayesian methods (Aone et al., 1998).

NetSum (Svore et al., 2007) produce extracts starting from newswire documents based on RankNet (Burges et al., 2005) as a machine learning algorithm to give a score to each sentence and to extract the most important ones. In addition to the common characteristics based on key-words and the position of the sentence, a set of functionalities based on Wikipedia knowledge database and the user’s requests are often used to extract sentences, which are considered important if they contain the query terms of Wikipedia entities.

In García-Hernández et al. (2008), the authors proposed an automatic summarization approach independent of the language and the treated field, it based on sentences extraction by using an unsupervised machine learning algorithm. The idea consists in using an unsupervised algorithm to gather the similar sentences in the same cluster, and in each cluster the most representative sentence is selected to build the summary. The automatic summarization system suggested proceeds in four stages:

- Preprocessing: Consists in eliminating the blank words and to apply a procedure of stemming (process of transformation of the inflections into their radical or root)
- Terms selection: Consists in choosing the size of sequence N-gram to describe the sentences. A sequence *N-gram* is a sequence of N words. It is said that a sequence N-gram occurs in text if these words appear in the text in the same order immediately one after the other.
- Terms Weighting: Consists in giving a weight to each element. This is done by a Boolean weighting (1 if the term exists, 0 if not), TF, IDF or TF.IDF methods.
- Sentences grouping: Consists in using an unsupervised learning algorithm to discover the groups of sentence having similar meaning. For this reason, the authors chose the known algorithm k-means which supposes that the number of clusters (groups) is previously known. K-means is an algorithm of partitioning data, this means a method of which the goal is to divide observations into K partitions (clusters) in which each observation belongs to the partition with the nearest average.
- Sentences Selection: At the end, the most representative sentences of each group are selected to build the summary.

The experiments carried out on documents DUC-2002 show that the method suggested gives more favorable results than other approaches.

1.5.9.1 Naïve Bayesian

Kupiec et al. (1995) implemented a Bayesian classifier that calculates the probability of a sentence in a source document to be included in the summary. In this approach, and in order to train the classifier, authors used a corpus of 188 pairs of documents / summaries (written by professionals abstractors) belonging to several domains. The features taken into account are: sentence length, cue phrases, sentence position in the paragraph, frequency of words in the sentence and words in upper case. Using this approach, the authors found that the best performance of the proposed system is obtained when combining position, cue phrases and length of the sentence.

Lin (1999) introduced a new extractive summarization system based on the decision tree algorithm. The proposed system investigated many features in order to compute the score of each sentence. The author evaluates his approach against several baselines and found comparable results with a simple combination function of features. Thus, a further study of other variations of the decision tree and machine learning algorithms is needed. Other works using decision tree algorithm are investigated in (Jin et al., 2008; Chen and Hung, 2009).

1.5.9.2 Hidden Markov Models

Several works introduced Hidden Markov Models (HMM) in their summarization systems. HMM is a statistical Markov model which uses a set of sequential features with unobserved states (Markov process) in order to detect the dependencies between sentences within a document. The hidden or unobserved states in the model represent whether a sentence is to be included in a summary or not. In the method proposed by Fung and Ngai (2006), HMM is used to calculate the probability of a sentence to be included in the summary depending on the state of the previous sentence (if it has also been included or not). This approach applies only to a story documents, which assume that sentences in the text are inherent. Experiment results show that performance of the proposed system is superior to conventional methods that do not incorporate text cohesion information.

The system described in Zajic et al. (2002) uses HMM to generate informative headlines of stories from English text. The evaluation process indicates that the produced headlines are closely similar to those generated by humans. CLASSY in another HHM-based system proposed by Schlesinger et al. (2008). The authors used HHM for sentences selection within a document.

Conroy and O'Leary (2001) adopted HMM for the summarization task. The used the features: sentence position, number of words in the sentence and probability of the word calculated from the input. The assumption is that the next sentence will be included in the summary depending on whether the current sentence is already part of the summary. Other works based on HMM have been done for domain specific summarization (Fung and Ngai, 2006; Barzilay and Lee, 2004).

1.5.9.3 Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression tasks (Cortes and Vapnik, 1995). The algorithm was applied on automatic summarization in order to rank sentences. Li et al. (2007) used the SVM model for multi-document summarization. Chali et al. (2009) adopted an ensemble learning model built from several SVMs to enhance the performance of a multi-document summarization task.

Schilder and Kondadadi (2008) proposed FastSum for query-oriented summarization. The authors used two kinds of features: word-based features which are calculated based on the frequency of words in sentences, documents, topics and clusters, and sentence-based features which include the position and length of the sentence in the document. The regression SVM is trained on a specific corpus of news data in order to estimate the opportunity of a sentence to be included in the summary.

In the work proposed by Fattah (2014), several features are taking into account: words frequency in the whole document, similarity with the title, the similarity of words among sentences and paragraphs, sentence position, existence of cue-phrases and the occurrence of non-essential information. The author investigated the effect of the combination of these features on several summarization models such as naive-Bayes classifier, maximum entropy and SVM model. The summary is then generated by combining the three models into a hybrid one. Performance evaluation on the DUC 2002 dataset shows that results were promising when compared with some existing techniques.

Better ranking techniques based on machine learning algorithms have been used to improve the quality of ATS systems. Adopting machine learning in Arabic text summarization is now started to show an interest by researchers in this area. Deep learning techniques, which is a part of machine learning algorithms, are successfully applied on many NLP and computer vision tasks. However, until now they have not been employed in summarization of Arabic documents.

1.5.10 Neural networks and automatic summarization

1.5.10.1 Deep learning

Deep Neural Networks are multilayered networks of classical architecture, but with several hidden layers; it is the way of managing their learning which has triggered renewed interest in their study since 2006. Different from shallow models, deep models are more compact and expressive in extracting low-dimensional data with more abstract features. Yet, learning by back-propagation has often proved ineffective in multi-layered networks, due to local minima, often quite bad, in which gradient descent trapped the method. Some researchers (Bengio et al., 2007; Hinton et al., 2006) have proposed new learning methods, usually layer-by-layer, to overcome the practical limitations of back-propagation and to better exploit the internal representation potential of so-called deep networks. The disadvantage of shallow architectures, to which Support Vector Machine (SVM) does not escape, has been debunked and argued by Bengio and LeCun (2007). Bengio et al. (2007) presented a greedy learning algorithm, based on a stacked auto-associators, which makes it possible to build the hidden layers one after the other and it uses back propagation to minimize the reconstruction error. Hinton et al. (2006) have departed from the Boltzmann machine model (Ackley et al., 1985) to define a stack of restricted Boltzmann machines, or Restricted Boltzmann Machines (RBMs) to construct Deep Belief Networks (DBN). One of the arguments put forward to justify the interest of deep networks is that a learning model layer by layer can extract more abstract features from the training dataset.

Additionally, Hinton and Salakhutdinov (2006) proposed a deep auto-encoder (DAE) in order to address the difficulty of unsupervised deep learning. The learning task in DAE is divided into two stages: the pre-training stage which consists of initializing the weights of the networks by appropriate values. The initial wrights are obtained by learning stacked RBMs. The input of the next RBM are the output of the first RBM. After the pre-training stage, the generative weights are obtained by unrolling the stacked RBMs (deep auto-encoder) and fine-tuning the whole network using back-propagation of error derivatives.

Unsupervised deep learning algorithms have been successfully applied to several domains. They have been used as an unsupervised feature learning methods in order to increase the power of features discrimination. An approach for sentiment analysis is presented by Rong et al. (2014). The authors used the capability of auto-encoders in feature extraction and dimensionality reduction to enhance the performance of the proposed method. In Yu, Huang and Wei 2018, the authors proposed a novel unsupervised image segmentation using a Stacked Denoising Auto-encoder to extract deep-level feature representations. Ijjina and C (2016) exploited an unsupervised pre-training phase based on stacked auto encoder in order to classify human actions. In the same context, principle component analysis (PCA) was combined with auto-encoders to achieve the multi-feature learning task designed for the hyperspectral data classification problem (Wang et al., 2016). In order to address the speech recognition task, another hybrid approach was proposed by Noda et al. (2014). It combines a deep denoising auto-encoder used to acquire noise-robust audio features, and a convolutional neural network (CNN) which is utilized to learn visual features from raw mouth area images in order to predict phoneme labels.

Another unsupervised learning algorithm has been simultaneously proposed by Kingma and Welling (2014), and Rezende et al. (2014). It is a novel version of auto-encoder called Variational Auto Encoder (VAE), which combines variational inference methods with deep neural networks. VAE has been successfully applied in automatic text summarization of Arabic documents (Alami et al., 2018).

1.5.10.2 Word embedding

The idea behind word embedding was first proposed in early works (Rumelhart et al., 1986; Pollack, 1990; Elman, 1991). Recently, Bengio et al. (2006) have proposed a neural probabilistic language model in order to predict the probability distribution for each words along with the probability function for word sequences (preceding words). The authors use in this technique a feedforward neural network and a locally linear embedding to learn jointly representations of high dimensional data and a statistical language model. Since 2010, the area of word embedding has been gradually developed, because importance new findings have been made affecting the quality of output vectors and the training speed of the model. Milkov et al. (2013) proposed a new neural network architecture for language modeling based on recurrent neural networks. They created the well-known word embedding model Word2Vec which can be implemented by two different models, namely Continuous Bag-of-Words (CBOW) model and Skip-gram model. In CBOW a windows around the target word and words before and after it (context) are used as the input of the model to predict the output which is the target word. Skip-gram does the opposite, the input to the model is the target word, and the output to predict are the surrounding words in the window around that word, i.e. predict the context around a word. Skip-gram predicts the context around a word, while CBOW predicts the word existing in the context. GloVe (Pennington, 2014) is another unsupervised learning algorithm designed for obtaining vector representations of words. In contrast to word2vec which is a predictive model, GloVe is a count-based model which seeks to build a vector representation of words based on the co-occurrence counts matrix. WE has been successively used in many NLP applications such as opinion classification (Enríquez et al,. 2016), sentiment classification (Ren et al., 2016; Giatsoglou et al., 2017; Yu et al., 2018; Xiong et al., 2018), document representation (Kamkarhaghighi and Makrehchi (2017)), named entity recognition (Das et al, 2017) and synonymy identification (Nguyen et al., 2015).

1.5.10.3 Extreme Learning machine

Presented by Huang et al. (2006), ELM was developed to learn a Single-Layer Feed forward Networks (SLFNs) in an efficient and expedient manner. First, ELM has been applied to supervised regression and classification tasks (Huang et al., 2012); and then it has been adapted to semi-supervised tasks by adding manifold regularization (Huang et al., 2014). Classical feed forward neural networks are usually trained by Back-Propagation (BP) learning algorithm, which faces problems of the slowness of learning speeds and local minimums. ELM can perform the learning stage in a very short time while preserving a better generalization performance. This has been demonstrated in many computer vision applications such as image segmentation and classification (Andrushia and Thangarajan, 2015; Cao et al., 2016), human action recognition (Minhas et al., 2010; Iosifidis et al., 2015) and face classification (Mohammed et al., 2011). In the same context, Huang et al. (2018a) proposed a new clustering method using ELM as an unsupervised feature learning technique. In medical domain, ELM has proven to be so successful for detecting the suspected neovascularization regions in retinal images (Huang et al., 2018b). Kasun et al. (2013) proposed a new unsupervised learning algorithm based on ELM and named the extreme learning machine auto-encoder (ELM-AE) in order to deal with unsupervised tasks. The ELM-AE is a neural network with a single hidden layer and the output is the same as the input data. The random weights and biases of the hidden nodes are randomly initialized and must be orthogonal.

1.5.10.4 Automatic text summarization with deep learning

ATS has been successfully addressed by deep neural networks in many works. As far as extractive approach is concerned, several unsupervised methods are proposed. Zhong et al. (2015) introduced query-oriented unsupervised multi-document summarization via DL model. The method proposed is based on a deep AE and divided into three phases: concept extraction phase used to filter out unimportant words, reconstruction validation phase used to globally fine-tune the whole network by back-propagation algorithm, the final stage consists of summary generation using dynamic programming algorithm. Experiment results demonstrate that the proposed method provides better summary quality compared to state-of-the-art summarization approaches.

In a work proposed by Yousefi-Azar and Hamey (2017), a deep AE is used to learn an abstract (and reduced) representation of the input documents. The authors explore both local and global vocabularies using both term frequency and TF.IDF feature in order to build a matrix representation of the documents in a specific corpus. This matrix is used as the input of the AE. In order to generate the summary, all sentences of a specific document are mapped into the concept space in order to compute the semantic similarity between sentences and a given query. The authors proposed an ensemble learning model in order to improve their model by adding small random noise to the input and selecting the top ranked sentences from several runs using majority voting technique. Evaluation of the proposed models has shown that the ensemble technique provides better summary quality compared to other models.

In supervised approach, Denil et al. (2014) introduced a ConvNet model to build the summary. The proposed method trains the CNN model to classify sentiments (positive or negative) of movie reviews. The classification objective helps in extracting silent sentences, through visualization techniques, to be included into the summary. CNN has also been used by Ha et al. (2015) in the summarization of Korean news articles and map the result into relevant images. Cao et al. (2015) introduced a multi-document summarization system which ranks sentences using recursive neural networks (RNN). The proposed system transforms the sentence ranking task into a hierarchical regression fashion which is modeled by RNN using hand-crafted word features as inputs.

Alami et al. (2018) proposed a novel approach to summarize Arabic documents. They used a deep VAE as an unsupervised feature learning technique. The authors used VAE as a generative model to handle the inference problem. Two summarization methods have been investigated: graph-based and query-based methods. The authors used the concept space learned by a deep VAE in order to compute the semantic similarity between sentences. Comparison with other approaches shows significant improvement of the propose approach and confirms that the VAE offers a more discriminative feature space in which the semantic similarity measure is more accurate. PadmaPriya and Duraiswamy (2018) used a deep learning algorithm in order to deal with a multi-document summarization task. The evaluation process showed promising results compared to the state-of-the-art models.

Abstractive summarization task has been addressed by Song et al. (2018). The authors combine the long short-term memory (LSTM) and CNN in order to build semantic phrases and improve the text summarization performances. The first step in this method extracts key-phrases from the input sentences, while the second step generates the summary using deep learning. The evaluation process used two different datasets, and has shown that the proposed method outperforms the existing abstractive and extractive models.

1.6 Abstractive Text Summarization Approaches

Extractive summarization is the most common approach, because of its simplicity compared to abstractive summarization, however this approach remains far from producing optimal summaries both in terms of content and linguistic quality. While extensive research has been done on the presentation of abstracts by extraction, very little work has been done in the abstractive framework. It is clearly time to make more progress on this alternative approach.

The summary by abstraction is a text smaller than the document to which it refers, and whose meaning is intended to be as close as possible to that of the document, without using sentences or portions of the source document. This approach, which is inspired by the field of artificial intelligence and cognitive models of text comprehension, appeared in the late 1970s. In comprehension-based approaches, the automatic summarization task must be consistent with what the human do. The construction of a text representation is therefore a fundamental step in the abstractive summary. The idea is that we need to extract text informations to build one or more representations. From these representations of which we know the organization and the components, we can find the most important elements and then produce the summary. Generally speaking, abstractive summarization takes place in three stages.

First, a semantic representation of the text sentences is constructed, which constitutes the phase of automatic comprehension of the text. Secondly, the system performs operations of selection, aggregation and generalization from these semantic representations in order to be able to construct a representation of it in the form of an abstract entirely generated from the meaning conveyed by the text. Thus creating new semantic representations, which corresponds to the phase of automatic reduction of the text content (or text compression). Thirdly, from this abstract, the system will be able to automatically build a textual summary by generating a new text with new sentences and syntactic constructs. However, it remains difficult to implement such a system given the need to code a large amount of knowledge that does not always appear explicitly in the source text.

In the following section, we will discuss two abstractive approaches: sentence compression and sentence fusion.

1.6.1 Sentence compression

Sentence compression consists of removing some of its non-essential constituents with the aim of obtaining a shorter sentence while preserving the semantics (sense) and grammaticality of the sentence. In the automatic summarization by extraction, where the most important sentences have been concatenated to produce the summary, no treatment is performed at the intra-sentence level to determine the useless information. Thus, a sentence is either retained in full or completely deleted from the final summary. Therefore, removing the least relevant constituents by sentence compression is a solution to this problem.

There are two major approaches for sentence compression: linguistic approach, which is based on a predefined rules and statistical approach, which uses a large corpus to detect patterns and automatically produce the rules. Statistical approach need a big training corpus, which consists of sentences and a pertinent version of their compressions. The compression task is generally defined so that the words of the compressed sentence all appear in the initial sentence. It is therefore an automatic summarization method that can also be combined with an automatic extractive summarization system, either by first selecting the relevant sentences and then compressing them, or compressing all the sentences beforehand, and then making a selection by extraction. It is very important to perform sentence compression on a summary to improve the compression ratio or to increase the number of sentences (thus the potential number of distinct ideas) present in a fixed size summary.

The work of Gagnon and Da Sylva (2006) focuses on a symbolic method for text compression. It consists in reducing the size of sentences by removing certain non-essential parts, using solely a syntactic analysis of the sentence. Each sentence of the text to be compressed is analyzed using a parser that identifies and classifies the syntactic dependencies between the words of each sentence. The parser output represents each sentence in the form of an analysis tree, which includes grammatical relationships between the words it contains. The sentences can then be compressed by cutting off sub-trees identified as not being essential to the structure of the sentence. In particular, the main sentence phrase is always kept, which makes the reduction of very short sentences (6 words or less) seldom possible. About 80 to 90% of sentences can be reduced by this technique, which applies in particular to French texts. About 25% of the compression were judged incorrect by a human evaluator. An error rate that may be too high for several applications, including automatic summarization. This rate is very dependent on the initial quality of the syntactic analysis of the sentence, and would therefore be improved if the analysis tool used were of better quality.

The method proposed by Cohn and Lapata (2009) is similar to that of Gagnon and Da Sylva (2006). They also use syntactic dependency trees, but they include additional permitted transaction types. Thus, in addition to sub-tree removal, the authors have also created a number of compression rules on trees that are actually rules of tree-to-tree substitution in the context of tree-substitution grammar model (STSG). This kind of transformation allows modifications not only to the tree nodes but also to the structure of the tree. Therefore, more complex new rules that take into account the syntax of the resulting sentence can be included. Thus, more advanced compressions can be made while paying attention to grammaticality. Tree transformation rules also have the advantage of being more generalizable to other tasks since they allow deletion as well as substitution, insertions and reordering. The rules were obtained by transduction, that is to say that observations of tree substitutions on a corpus of sentence compressions make it possible to discover reusable rules on new cases.

To date, many systems using abstractive approach have been emerged. For example, the FRUMP platform proposed by DeJong (1982) aimed to get closer to what the human could do in the context of summarization news articles. The author performs the summarization task by identifying the main events and actors described in the text and choosing the most relevant among them. Information is acquired from events commonly described in news articles.

As shown in the studies of (McKeown and Radev, 1995; Radev and Mckeown, 1998; Aone et al., 1998; McKeown et al., 1999; White et al., 2001; Daume III et al., 2002), predefined intermediate data structures are input to the system for the summary construction. In these structures, information from the source document is extracted, such as the events present in the documents or the predicate-argument type structures. Data structures once fed by this information are input to language generation tools that generate grammatically correct sentences in the desired language. The approach proposed by Knight and Marcu (2002) is based on sentence compression using a decision-based model to learn several rules applicable in different contexts.

Moawad and Aref (2012) proposed a new reduction approach based on semantic graph for abstractive single-document summarization. The authors perform the summarization task in three phase. First, a rich semantic graph is generated from the input text. Second, the generated graph is reduced. Finally, the abstract is generated from the reduced graph. Evaluation by a simulated case study shown that the proposed approach reduce the original text by 50%.

The abstractive approach introduced by Khan et al. (2015) generate the summary from the semantic representation of the source documents. In this method, semantic role labeling is used to represent the content of the input text by predicate argument structures. Then, the semantic similarity between predicate argument structures is calculated in order to group similar structures into the same cluster. Finally, genetic algorithm is employed in order to rank these structures. Experimental results carried out on DUC-2002 corpus, shown that the proposed method performs better than other summarization methods.

1.6.2 Sentence Fusion

In human-generated summaries, in addition to sentence compression, a word is sometimes replaced by another, and information provided from two sentences is often combined to create a more meaningful one. Sentence fusion can be defined as a summary generation task, where the input is sentences with a certain level of overlap, and where the output would be a single sentence containing the most important information contained in the input sentences. Similar sentences would then be merged into the summary. Merging could be seen as a tool that replaces or complements sentence compression in a standard abstractive summarization system, or as a full-fledged summarization technique as in the works we present here.

The information fusion approach was introduced by Barzilay and McKeown (2005) as part of the multi-documents summarization (MultiGen). A common multi-document summarization approach is to find similarities between the input documents and extract common relationships to form the summary. The technique introduced by Barzilay and McKeown (2005) takes a set of similar sentences as input and produces a new sentence containing common information contained in most input sentences. Their approach addresses two challenges: the identification of overlapping sentences that convey common information and the combination of these sentences into a single sentence that is grammatically correct. The results are compared to a human-generated summaries the same theme, based on the recall and precision of the information contained in the merged sentence. The F-measure of the human-generated summary, calculated based on the information contained in the sentence, was 96%. The system proposed by the authors obtains 65% precision, 72% of recall and 68% of F-measure. The resultant sentence was 44% longer than the sentence generated by human. These results are promising and demonstrate the feasibility of the task. However, it should be noted that the loss of grammaticality is important and very problematic in practice.

1.7 The Arabic language

Arabic is the most popular Semitic language in terms of speakers. It shares many characteristics with the other Semitic languages related to morphology, vocabulary, the use of short and long vowels, capitalization, etc. It is composed of 28 letters and written in a cursive script from right to left. Arabic, the language of the Quran, has become the language of a civilization and no longer limited to the inhabitants of the Arab peninsula who spoke it. Arabic is the language spoken by around 400 million people living in the North Africa, Chad, Horn of Africa (Djibouti, Eritrea, Ethiopia, Somalia) and the Middle East (Prochazka, 2006; Farghaly and Shaalan, 2009).

Nowadays, Arabic has become one of the most widely used languages on the Internet, where the number of users has increased significantly. There are almost 219 million of Arabic users, which constitutes 5.3% of the total number of internet users around the world.

There are three categories of Arabic depending on the writing and reading styles:

- Classical Arabic which is associated with religion and classical Arabic literature. It is used by Arabic-speaking people in their daily prayers.
- Classical Arabic is the literary form used in the purposes of writing and printing. It is also the religion language for all Muslims (including Quran), regardless of their vernacular language.
- Modern Standard Arabic (MSA), based on the syntax, morphology and phonology of classical Arabic, it is the official spoken language in the Arab world, standardized and studied at school. It is daily used in the political debates, scientific and literary texts and in various kind of spoken and written media, such as newspapers, journals and news broadcasts.
- Arabic dialects, which are the true forms of mother tongue, used in everyday life. The Arabic dialects can be grouped into five groups: Egyptian dialect, Maghrebian dialect, Gulf dialect, Levantine dialect and Iraqi dialect.

The version of Arabic to which we are interested in this work is the Modern Standard Arabic (ASM). Given its morphological and syntactic properties, Arabic is considered as a difficult language to implement in the field of NLP. With the advent of the Internet and search engines, the amount of Arabic documents available in electronic format has become huge. As a result, several research projects for the automatic processing of Arabic are beginning to emerge.

1.7.1 Characteristics of Arabic

1.7.1.1 Absence of diacritics

Letters in Arabic are accompanied by signs placed below or above of them for distinguishing the word from another homonym in terms of meaning and pronunciation. Diacritical marks are needed for the purpose of morphology, semantic analysis and other linguistic and voice features

(Chennoufi and Mazroui, 2017). Indeed, word without diacritics can have several forms by adding these marks. Table 1.3 illustrates the morphology analysis of word “kataba”/ktb using Alkhalil analyzer (Boudchiche et al., 2017). In the following table, we show some results among 17 results returned by the analyzer.

Table 1.3 Morphology analysis of word kataba / “كَتَبَ”

Word with diacritics	Notation	English meaning
كَتَبَ	Kataba	He wrote
كُتِبَ	Kutiba	it was written
كُتُبٌ	Kutub	Books
كَتْب	Katb	Writing
كتَّبَ	Kattaba	

Usually, Arabic texts do not incorporate diacritical marks and they are often absent (Farghaly and Shaalan, 2009). Habash (2010) mentioned that diacritics are present in only 2% of Arabic texts. These diacritics are written only in certain circumstances such as in the Quran, Hadith and some learning books used to teach Arabic. The absence of diacritical marks is one of the biggest problems making Arabic NLP more complicated.

1.7.1.2 Agglutination

Arabic is a very derivational and inflectional language, which makes the NLP tasks, such as lemmatization and stemming, more difficult. The number of root in Arabic is approximately 10 000, and with adding affixes to these root, there are approximately 120 patterns (Shaheen and Ezzeldin 2014).

Unlike English and French, Arabic words are composed by several morphemes which represent the lexical elements (Shaheen and Ezzeldin 2014). Indeed, articles, prepositions and pronouns are stuck to adjectives, nouns, verbs and practices to which they relate in order to constitute a new lexical unit that convey several morpho-syntactic information. Therefore, words in Arabic can represent a complete English or French statement. This increases the complexity of Arabic NLP tasks, such as segmentation and stemming. For example: le mot « أتتذكروننا » means « *Est-ce que vous vous souvenez de nous?* » in French and « *Do you remember us?* » in English. This characteristic can improve the morphological ambiguity. Indeed, it is sometimes difficult to distinguish between a proclitic or enclitic and an original character of the word. Generally speaking, a word in Arabic is composed of it basic form (stem derived from the root), around which various prefixes, suffixes, proclitic and enclitics are attached. For example, the character “و”" is a part of the word “وصل” (*he arrived*), while in the word “وفتح“(*and he opened*), it is a proclitic.

1.7.1.3 Irregularity of the word order in the sentence

In Arabic, the words in a sentence can be composed in different ways while keeping the same meaning conveyed by this sentence. Indeed, the order of the words is relatively free in a sentence. This provides syntactic ambiguities in the fact that it is necessary to provide in the grammar all the rules of possible combinations of the words order in the sentence (Belguith et al., 2005). Table 1.4 shows how we can change the word order in the sentence to get three sentences with the same meaning (Belguith et al., 2005).

Table 1.4 Irregularity of the word order in the sentence

Verb + subject + complement	فعل + فاعل + متمم	ذهب الولد إلى المدرسة
subject + verb + complement	فاعل + فعل + متمم	الولد ذهب إلى المدرسة
Complement + verb + subject	متمم + فاعل + فعل	إلى المدرسة ذهب الولد

1.7.1.4 Irregular punctuation

The Arabic text is also characterized by the irregular use of punctuation marks. These punctuation marks were introduced recently in the Arabic writing system. In addition, we can find an entire Arabic paragraph containing no punctuation except for a dot at the end of this paragraph. Thus, it should be noted that the presence of punctuation marks cannot guide sentences segmentation as in the case of other Latin languages, such as French or English (Belguith et al., 2005). The segmentation of Arabic texts must be done not only by punctuation marks and typographic markers but also by particles and certain words such as subordination and coordination conjunctions, etc. (Belguith et al., 2008). These particles are attached to the inflected form of words (agglutination of morphemes), and require a rigorous morphological analysis to identify them.

1.7.2 Challenges in Arabic NLP

1.7.2.1 Text segmentation

Text segmentation is one of the important phases in automatic processing. It consists to divide the text into different units such as paragraph, sentences, and words. However, in Arabic language, this operation seems to be difficult for many reasons. First, Arabic alphabet does not have capitalization and therefore it is more difficult to recognize sentence boundaries as well as recognizing Named Entities. Second, Punctuation marks are not used in regular way. Third, there is some grammatical particles, like *“و/ and; ف/ so”* etc. that marks a separation between two sentences.

Despite the importance of this issue, there are a few solutions that address the Arabic text segmentation. Among these solutions we can quote the use of word delimiters such as spaces or punctuation. Stanford Arabic Parser use a morphological analyses generated by the Buckwalter analyzer to segments an Arabic text. We can also mention the STAr system (Baccour, 2004) based on the textual exploration techniques of punctuation marks, connector words and some grammatical particles (Belguith et al., 2005). Other methods for Arabic text segmentation are proposed. They are based on a rigorous analysis dealing with the Arabic morphology complexity. They consist in segmenting clitics: conjunctions, propositions, pronominal clitics, and defined articles (Diab 2009).

1.7.2.2 Morphology analysis

Morphology is the branch of linguistics that deals with the internal structure of words. It studies word formation, including affixation behavior, roots, and pattern properties. It consists to identify and analyze the internal structure of words and other linguistic units, such as stem, root, affixes, part of speech...etc. A word is decomposed to several units (prefix, suffix, root, etc.). Word morphology is very helpful in the process of acquiring linguistic information. It also has an important role to play in the disambiguation of word sense. The main problem with this analysis is the absence of agglutination and vocalization (vowels). There are some morphology analysis systems in Arabic that address this issue. We can quote:

- BAMA is a morphology analyzer implemented by (Buckwalter, 2002), which transliterates the input text to ASCII before any preprocessing. The result is then reconverted into Arabic to be intelligible. This system is limited only to analysis of words appearing in Arabic dictionaries and does not analyze texts containing numbers.
- SAMA: is a new version of BAMA developed by (Graff et al., 2010). The list of analyzed words and proposed solutions are improved.
- MORPH-2 (Kammoun et al., 2010) is a morphology analysis system based on a lexicon containing all the words (3266 roots) with their associated characteristics. It is based on five stages: sentence segmentation into words, morphological preprocessing, affix analysis, morphological analysis and post-processing.
- ALKHALIL2: Alkhalil Morpho System is a morphological syntactic parser of Standard Arabic words developed by Boudchiche et al. (2017). The system can process non vocalized texts as well as partially or totally vocalized ones. For a given word, it identifies all possible solutions with their morph syntactic features: vowelizations, proclitics and enclitics, nature of the word, voweled patterns, stems, roots and Syntactic form. The approach used is based on modeling a very large set of Arabic morphological rules, and also on integrating linguistic resources that are useful to the analysis, such as the root database, vocalized patterns associated with roots, and proclitic and enclitic tables. This is a new version of Alkhalil, which has been released to improve the performance of the system.
- MADAMIRA: is developed by (Pasha et al., 2014). This system combines the result of the two morphological analysis systems: MADA (Habash et al., 2013, 2009) and AMIRA (Diab et al., 2007). It use SVM in order to choose one result from several solutions generated in the first step.

1.7.2.3 Part of speech tagging

Part-of-speech (POS) tagging consists to assign the grammatical category, such as noun, verb, adjective, adverbs, etc., to each word in the text depending to the context which it appears. It plays an important role in lemmatization, parsing, information extraction and information retrieval. In addition to Arabic morphology analysis, POS tagging becomes particularly important because of the lexical ambiguities of words. Some Arabic POS tagging systems was proposed using a combination of both statistical and rule-based techniques since hybrid taggers seem to produce the highest accuracy rates, but the most commonly used is based on a numerical approach. Khoja (2001) proposed a hybrid system by adapting the BNC system, designed for English. This tagger combines a statistical features of the lexical units as well as a set of rules that are derived from the traditional theory of Arabic grammar. In another work, Diab (2009) proposed a new system, called AMIRA. It is the recent version of the first tagger (Diab et al., 2004), faster for text segmentation (words and sentences) and POS tagging.

1.7.2.4 Named-entity recognition:

Another problems occurs in Arabic NLP, the named-entity recognition, it allows the identification and typing of people names, organizations, places, etc. It should be mentioned that there is a little works on the named entity recognition for Arabic texts. Most of these works were usually developed based on rules or based on machine learning.

1.7.2.5 Stop-words removal

Words which contain no particular meaning in the text and appear almost in all documents are considered as unimportant. Stop-words are very common words with a mainly structural function; they are recurrently used in a text, carry little meaning and their function is syntactic only. They do not indicate the subject matter and do not add any value to the content of their documents. In Arabic, words like (هو, هذا, الذي, هي) are frequent in sentences; but with little significance in the implication of a document. These words should be deleted from the text in the preprocessing stage to reduce the size of the vocabulary and to consider just the meaningful terms of the document.

Stop-words removal is mainly based on a set of empty words predefined by the author. There is no typical list of stop-words specific to Arabic, but several lists have been proposed. Khoja and Garside (1999) adopted a list of 168 stop-words in their stemmer system. Chen and Grey (2001) created a list of Arabic stop-words by translating those used for English. They also added the most frequently terms existing in TREC 2001 collection. Bouzoubaa et al. (2009) proposed a standard structure for a stop-words dictionary based on existed resources. Most studies have shown that the elimination of stop-words in the preprocessing stage increases significantly the performance of Arabic NLP systems including automatic text summarization (Darwish and Magdy, 2014)

1.7.2.6 Arabic word stemming

One of the most challenging issues in Arabic language is the word stemming. Arabic words can have different form by adding affixes (prefixes and suffixes) to the original words (root). Stemming an Arabic word consist to find the appropriate root for the giving word by removing the attached affixes. The main characteristic of Arabic is that the most words are built up from roots by following certain fixed patterns and adding infixes, prefixes and suffixes.

Arabic language has been considered as a challenging language for automatic text summarization and information retrieval due to different reasons. Arabic is highly inflectional and derivational, and words can have many different forms which makes morphology a very complex task. In addition, written character in different ways depends on the position of the letter in the word, which can add a complexity to Arabic words analysis. Therefore extracting lemma, stem or root is a hard problem for Arabic.

One of The strengths of Arabic language is the root of words. Arabic words are generally based on a root, which mean that the root can be a base of different words with informative related meaning and with adding suffix on the root we can build a set of derivations. These derivations represent a same area. Finding a root of Arabic word (stemming) helps in mapping grammatical variations of a word to instances of the same term. For example the root لعب “laaeba” is used for many words relating to “playing”, including “لاعب” , “ laaeb”, “player”, “ملعب”, “malaab”.

Based on this consideration, multi derivations of the wording structures in Arabic language allow a semantic representation of the text which is being closer to the semantic foundations.

A good representation of Arabic text may impact positively on the quality and accuracy of automatic text summarization. In our research work we can improve a quality of Arabic text summarization by using not only a statistical feature selection method but also structural and conceptual (semantic) ones. In addition, because words sharing a root are semantically related, feature selection techniques based on the root can improves the methods of Arabic text summarization. Many studies have shown that applying words stemming in the preprocessing stage increases significantly the performance of Arabic NLP systems including automatic text summarization (Froud et al., 2010; Bsoul et al., 2014; Atwan et al., 2014; Alami et al., 2016).

Based to the desired level of analysis, Arabic stemmer algorithms are classified as either root-based (Salton and Buckley, 1997) or stem-based (Manning et al., 2008; Baeza-Yates and Ribeiro-Neto, 1999; Llopis et al., 2002). Root-Based approach uses morphological analysis to extract the root of a given Arabic word. Many algorithms have been developed for this approach. A superior root-based Arabic stemmer is Khoja's stemmer (Khoja and Shereen, 1999), which is the widely used system to extract the roots of Arabic words. The Khoja algorithm removes suffixes, infixes, and prefixes and uses pattern matching to extract the roots. However, the algorithm suffers from problems especially with names and nouns. While the stem-based approach or light stemmer approach aims only to remove the most frequent suffixes and prefixes of a given Arabic word. Light stemmer is mentioned by some authors (Larkey et al., 2002; Aljlayl and Frieder, 2002; Larkey and Connell, 2002; Chen and Gey, 2002). Larkey’s stemmer or light10, developed by Larkey et al., 2002 and Larkey et al., 2007, is the most widely used Arabic light stemmer. Light stemming does not deal with patterns or infixes; it is simply a process of stripping off prefixes and/or suffixes. Although light stemmers produce fewer errors than aggressive root-based stemmers, aggressive stemmers reduce the size of the corpus significantly. Both Arabic root-based and stem-based algorithms suffer from stemming errors. The main cause of this problem is the stemmer’s lack of knowledge of the word’s lexical category (e.g., noun, verb, and preposition) (Atwan et al., 2014). Recently, Alkhalil (Boudchiche et al., 2017) and MADAMIRA (Pasha et al., 2014) are two robust systems which have been proposed for the purpose of morphological analysis including roots and stems extraction. They identify all possible solutions with their morph syntactic features including root and stem. These systems are able to identify one unique solution by using a specific technique such as SVM or Viterbi algorithm.

1.8 Arabic summarization approaches

Most systems of automatic text summarization are made to handle the most popular languages such as English, French, etc. On the other hand, though many achievements are achieved for English, few works have been proposed for Arabic text summarization. Therefore, there is a growing need to develop systems that process and summarize electronic Arabic texts. Arabic content, such as texts, documents and videos, has become huge due to the availability of this language on social networks such as Twitter and Facebook. Thus, automatic processing of this language is becoming necessary. Arabic as an important language in the world, has not been studied enough, and the numbers of research are still few in Arabic NLP due to its complex nature. Some of those reasons are, first the different ways that certain combinations of characters can be written. Second, the wide range of derivations and inflection of functional words makes morphology analysis a very complex task. Third, Arabic words are often ambiguous due to the tri-literal root system. Compared to research on English, works on automatic text summarization for Arabic are fairly small and recent. Douzidia and Lapalme (2004) was to our knowledge the first research work designed for Arabic text. It uses a linear combination of many sentence scoring features: terms frequency, sentence position, cue words and title words.

According to Al-Saleh and Menai (2015), there are three approaches to extractive ATS: numerical, symbolic, and hybrid. Numerical-based approaches assign scores the selected text elements (e.g. words or sentence). Several methods are used to compute the score of these elements. Key examples include statistical, probabilistic and machine learning techniques. The final summary is generated by selecting the important sentences according to their scores while respecting the predefined summary length. In the symbolic approaches, a discourse structure and coherence of the text is represented by rhetorical relations using discourse analyzing methods such as Rhetorical Structure Theory (RST). The hybrid approach combines both the numerical and symbolic approaches.

The survey carried out by Al-Saleh and Menai (2015) shows that the majority of research in Arabic text summarization focused on numerical approaches. However, symbolic and hybrid approaches are not studied enough due to the lack and limitations of resources and tools that are able to automatically process Arabic texts (Al-Saleh and Menai, 2015).

1.8.1 Symbolic approaches

The only work in the literature that followed a pure symbolic approach is Al-Sanie (2005). The author developed an automatic summarization system for Arabic texts based on rhetorical structure theory (RST) where 11 relations were identified and used in the summarizing task. This system, after the rhetorical analysis of the text, generates all possible representations of the text in the form of a rhetorical tree (RS-tree) using textual units based on cue phrases. Then, the summary is extracted from the highest level of the generated trees. The system was evaluated by comparing the results obtained with manually generated summaries. This system worked out quite well with small and medium articles size.

1.8.2 Numerical approaches

The first summarization system designed for Arabic texts was proposed by Douzidia and Lapalme (2004) and named *Lakhas*. The main motivation behind *Lakhas* is to minimize the error caused by machine translation when summarizing Arabic documents. The authors proposed to summary the Arabic documents rather than summarizing translated documents provided by English systems. The *Lakhas* architecture is composed of the following modules: 1) sentence segmentation which extracts each sentence from the given text and assigns a weight according to its initial position; 2) word segmentation, which extracts each word in each sentence and calculates the frequency of each word in the sentence; 3) Normalization by replacing the different variants of characters by a single one; 4) stop-words removal; 5) Lemmatization using a simple prefix and suffix removal (Darwish and Oard, 2002); 6) Calculation of the weight of each sentence based on *TF.IDF* of each word in the sentence, title words, indicative expressions in the sentence and sentence position in the text.

Lakhas uses classical sentence scoring features defined by (Luhn, 1958; Edmundson, 1969), which are: sentence position, terms frequency, title words and cue words. The score of each sentence is calculated by a linear combination of the four features using the following formulation:

$$Score(S) = \alpha_1 Sc_{lead} + \alpha_2 Sc_{title} + \alpha_3 Sc_{cue} + \alpha_4 Sc_{tf.idf} \tag{1.5}$$

Where:

- Sc_{lead} is equals 2 if the sentence is in the first position, and 1 otherwise.
- $Sc_{title} = \sum_{w \epsilon S} a(w) * tf(w)$ where $tf(w)$ is the frequency of the word w in the sentence S; and $a(w)$ is set to 2 if w is a part of the document title (zero otherwise)
- $Sc_{cue} = \sum_{w \epsilon S} c(w) * tf(w)$ where $c(w)$ is set to 1 if w is a cue word (zero otherwise)

Lakhas was the first summarization system designed for Arabic language that has been evaluated and compared with English systems in DUC 2004 competition and has achieved good results. It should be noted that, in this competition, *Lakhas* used the same evaluation corpus than other systems. In order to evaluate the result of *Lakhas*, the generated summary is translated into English by using a commercial machine translation system; and then compared with provided English summaries using the ROUGE framework (Lin, 2004). This is due to the shortage of Arabic corpora designed to evaluate Arabic summarization systems.

El-Haj and Hammo (2008) developed a query-oriented summarization for Arabic texts. The summary is generated from the most relevant passages extracted using the vector space model (VSM) with the cosine similarity measure. The size of the output summary was configured so it did not exceed the half of the input text. The evaluation task is based on articles extracted from Arabic Wikipedia. The generated summary is manually compared by 1500 volunteers from different educational levels.

El-Haj et al. (2011a) proposed two Arabic text summarization systems: AQBTSS, a query-based Arabic single-document summarizer, and ACBTSS, a concept-based summarizer. The first system takes an Arabic document and Arabic user’s query in order to generate a summary for the document in agreement with the given query. The second system is based on a bag-of-words representation of some concepts proposed by the authors. In both systems, the vector space model is used to score sentences and the summarization result is consistent with the query or concept. In both systems, all sentences all scored according to the vector space model (VSM). The results showed that, in both systems, the summarization result is consistent with the query or concept.

In another work, El-Haj et al. (2011b) adopted the clustering technique to generate generic, extractive and multi-document summaries, while eliminating redundancy within these summaries. In the first experiment, the authors used the k-mean clustering technique to assign each sentence to a specific cluster based on the cosine similarity measure. Then, the summary is generated by selecting sentences through the use of two different methods. The first one selected sentences from the largest cluster, while the second selected the first sentence from each cluster. In the second experiment, the difference is that the sentences are selected before applying the clustering. This method selects only the first sentence and the one that is most similar to it. Due to the absence of Arabic gold standard summaries, the authors evaluated and compared their system with other English summaries using the English and Arabic version of DUC 2002 dataset. The Arabic version of DUC 2002 datasets was generated using Google Translate.

The system described by Haboush and Al-Zoubi (2012) is oriented towards the determination of the root of each word in a sentence. Based on roots found in the text, the words can be grouped into separate clusters. The authors assume that the important words in the text appear several times. Thus, the main feature considered for the proposed method is words frequency. The preprocessing stage is first applied on the input text. It consists of three steps: i) splitting the text into paragraphs, sentences and words; ii) removing stop-words such as هو, هذا, الذي, (هي); iii) and stemming each word in each sentence to find its root. All the words with the same root are then grouped into the same cluster, and, in a cluster, the frequency of a word is represented by the number of roots in the cluster. After that, the weight of the words w_i in the sentence S_j is calculated using the following formula:

$$W_{ij} = \log(N/rw_i) * tf \tag{1.6}$$

In this equation, N represents the number of words in the sentence S_j, rw_i represents the weight of the word w_i and, finally, tf is the term frequency, which is calculated by dividing rw_i over the maximum frequency in the document.

The score of each sentence is calculated by summing the weights of its words. The importance of each sentence is then increased if it contains some indicative expression such as (يدل ذلك, اهم الامور ...). At the end, the summary is generated by taking the sentences that have the highest weight. In the evaluation process, the authors used a corpus of ten documents. For each document, four summaries are manually generated. They compared the summary generated automatically by the proposed system with the reference summaries using recall and precision measures.

In El-Shishtawy and El-Ghannam (2012), the authors proposed an Arabic automatic text summarization by identifying the important key-phrases of the document. The stemming and lemmatization of Arabic words were used in this work to compute the text features. The algorithm is based on the assumption that the key-phrases represent the most important concepts in the text. The extraction process of key-phrases used in this work is based on the system described in El-Shishtawy and Al-sammak (2009). Instead of using only the statistical information such as terms frequency and distance of terms, the extractor is also provided with linguistic knowledge to improve its effectiveness.

The method proposed by Oufaida et al. (2014) deals with both single and multi-document summarizations for Arabic. The system extracts the summary sentences by ranking the terms of each sentence. To build a summary with minimum redundancy, the authors extract and assign scores to the most relevant terms by using both a clustering technique and an adapted discriminant analysis method: mRMR (minimum redundancy and maximum relevance) (Peng et al. 2005). The experimental results on EASC (Essex Arabic Summaries Corpus) for single-document summarization and TAC 2011 Multi-Lingual datasets for multi-document summarization showed that the suggested approach is competitive to standard systems and outperformed the lead baseline.

Many Arabic text summarization studies include machine learning approaches in their summarizers, such as (Sobh et al., 2007; Fattah and Ren (2009); Boudabous et al., 2010; and Belkebir and Guessoum, 2015). Sobh et al., 2007 developed an Optimized Dual Classification System for Arabic Extractive Generic Text Summarization. The authors integrated Bayesian classification and Genetic Programming (GP) in an optimized way to get better results using reduced features of each sentence. The extracted features are based on sentence position and counting methods in addition to Arabic morphological analysis and POS tags. The system used five basic features: 1) Sentence weight; 2) Sentence length; 3) Sentence position in the text; 4) Sentence position in the paragraph; 5) Sentence length in the paragraph. In addition to these basic features, six other features have been proposed to take into account the relation between sentences in terms of similarity and to use the results of the morpho-syntactic tagging (POS tagging) of words in each sentence. To evaluate the methods, authors built their own corpus composed of 213 Arabic documents. The results of recall, precision and F-measure shown that the summaries of the proposed system are comparable with a human generated summaries.

Fattah and Ren (2009) investigated several machine learning models for ATS. The authors extracted ten features, such as, sentence position, keywords, sentence resemblance the title and other sentences and named entity. They used their own corpus composed of 100 Arabic documents and 50 English documents in order to train several models: genetic algorithm, mathematical regression, feed forward neural network, probabilistic neural network, and Gaussian mixture model. Sentences are then ranked according to the trained models. The experimentations shown that the obtained results are promising especially when using the Gaussian mixture model. It should be noted that, the authors used a limited corpus to train their proposed models (100 Arabic documents and 50 English documents). This impact negatively the performance of the machine learning based models, which require a large amount of training data.

SVM was also investigated for Arabic summarization (Boudabous et al., 2010). It is used to classify each sentence as a summary or not a summary sentence. Boudabous et al. (2010) used 15 features including the TF.IDF score in the learning stage. They built their own corpus from 500 Arabic documents covering different domains. Human experts have been request to produce three summaries for each document. The average F-measure achieved by the proposed method was 0.991, which is very high. This high result closes to 100% is obtained because the authors used their own corpus and do not compare their result using standard corpora.

Belkebir and Guessoum (2015) presented a machine learning-based approach which uses the AdaBoost algorithm to generate the summary of Arabic documents. When the training stage is finished, AdaBoost boosts the SVM classifier to check whether or not a new sentence can be incorporated in the summary on the basis of a number of features taken out from each sentence. The used set of features include the position and length of the sentence, the number of keywords and title words, and whether it is the introductory or the closing sentence in the text. To assess their approach, the authors used their own corpus that consisted of twenty news articles in Arabic and their human generated summaries. This approach was compared against other Machine Learning techniques that use multilayer perceptron and j48 decision trees. The obtained outcome shows that this method does better than other existing methods.

1.8.3 Hybrid approaches

Based on the RST technique (Al-Sanie, 2005), Azmi and Al-Thanyyan (2012) built an extractive Arabic summarization system that produces various-size summaries relying on the choice of the user. The proposed system is based on two main stages. Firstly, RST is used to produce RS-tree that is used to generate the initial draft of the summary. Secondly, in the primary summary every single sentence obtains a score that is the sum of five features taken from these sentences. These features are sentence position, whether it has numbers or it is located on the first line of the document, the total frequencies of its words and the existence of title words. The score of each sentence S is computed based on the *FarsiSum* scoring formula designed for Farsi text summarization (Mazdak, 2004). This formula was modified by integrating features that are more suitable for Arabic:

$$Score(S) = Position + C_{Numerical} + C_{Firstline} + C_{Titlewords} + \frac{1}{3}\sum_{w_i \in S} w_i \quad (1.7)$$

These features are sentence position ($Position$), whether it has numbers ($C_{Numerical}$) or it is located on the first line of the document ($C_{Firstline}$), the total frequencies of its words ($\frac{1}{3}\sum_{w_i \in S} w_i$)and the existence of title words ($C_{Titlewords}$). The feature $Position$ depends on the sentence position. The closer to the beginning, the higher is the position score. It is calculated by the following formula:

$$Position = \frac{1}{sentence\ position} * 10 \quad (1.8)$$

In order to include the most suitable sentence in the final summary, the authors formulated the selection problem as a 0/1-Knapsack problem (Horowitz et al., 1998). This problem was solved using dynamic programming algorithm by selecting a subset U of sentences that satisfies:

$$max \sum_{S_i \in U} Score(S_i) \quad (1.9)$$

Finally, the generated summary must not exceed a maximal length indicated by the user. To evaluate their method, the authors used their own corpus containing 32 documents extracted from popular Arabic newspapers. The model summaries were generated by a single human professional. For each document, one human summary was generated and did not exceed 31% of the original document. Experiments on sample texts using recall, precision, F-measure, and ROUGE, showed that the proposed system outperforms some already established Arabic summarization systems, even those requiring machine learning such as the system proposed by Al-Sanie (2005).

Ibrahim and Elghazaly (2013) followed a hybrid approach that investigated two summarization techniques by extracting the most important paragraphs from Arabic texts: RST technique for Rhetorical Representation and Vector Space Model (VSM) technique for Vector Representation. The former builds a Rhetorical Structure Tree (RS-Tree) of the input text using the RST technique and construct the summary with the most significant paragraphs. The latter makes a text representation using the VSM technique based on the cosine similarity measure. More specifically, the score of each paragraphs is computed using its cosine similarity with the title:

$$CosineSimilarity(T,P) = \frac{T.P}{|T||P|} \tag{1.10}$$

Where T represents the title vector and P represents the paragraph vector. These vectors are constructed using the $TF.IDF$ weights.

The experimental results showed that Rhetorical Representation method yields better first in terms of precision measure and secondly in terms of the quality of the produced summaries that are more readable than Vector Representation; however, the performance of the second was better with long articles. It is worth noting, that, in this work, there is no details about how the reference summaries were created.

Fejer and Omar (2014) introduced single and multi-document summarization approaches by combining clustering technique and key-phrase extraction. After preprocessing the input text, the authors used single and complete linkage clustering in order to identify the number of appropriate clusters. Based on this number, the k-means clustering technique is applied to cluster the text and put similar sentences into the specific cluster. Then, key-phrases, which reflect the subject of the text, are extracted from each cluster by choosing noun phrases with several features, such as the frequency of the key-phrase in the text and the number of sentences in which the key-phrase appears. After that, the rank of each sentence is calculated by the following formula:

$$Score(S) = \frac{The\ total\ number\ of\ words\ in\ every\ keyphrases\ in\ the\ sentence\ S}{The\ number\ of\ words\ in\ the\ sentebnce\ S} \tag{1.11}$$

The experimentation dataset used in this work is the Essex Arabic Summaries Corpus (EASC). The authors evaluate their method by comparing the system generated summary with the model summaries available in the EASC dataset using several ROUGE metrics. It is worth noting that the compression ratio (length of the system summary) is an important parameter of the evaluation process; and, in this work, there is no information about the value of this parameter used in the experimentations.

1.8.4 Synthesis and limitations of current approaches

It is clear from Table 1.5 that most studies in Arabic text summarization rely on a statistical approach. The statistical approach is based on the words existing in the document, and one of the obvious disadvantages of this approach is that, it overlooks the semantic relationship among words, which amounts to saying that its meaning representation of documents is not accurate. The system is always limited to the words explicitly mentioned within the input text document. For instance, if the system cannot find the relationships between terms like « بترول » (Petroleum) and « نفط » (Oil), it would handle these words separately as two different unrelated terms, and this may affect negatively their importance in the input document. The ability to detect such relationship between terms in a document requires additional knowledge that are external to the analyzed document, and an analysis module for learning the relationships between different terms.

Statistical-based systems are also affected by the same limitations in concept detection. For example, with expressions like «استخراج النفط», «انتاج النفط», « استخراج البترول » and «انتاج البترول», the system should be able to understand that these expressions refer to the same concept. The relationship between different concepts detected in the analyzed document is not exploited in the statistical-based systems. Normally speaking, entities such as "Hiroshima", "Nagasaki" and "atomic bomb" should be marked by the system as entities connected by a relation.

Similarly, in the supervised machine learning approaches like (Boudabous et al., 2010; Sobh et al., 2007; Belkebir and Guessoum, 2015; El- Fishawy et al., 2014; and Fattah et al., 2009), training is a decisive step to ameliorate the precision of the system. Therefore, in this kind of approach all the words that appear in the testing documents but not in the training documents are ignored and no new information, outside what is already available in the test documents, is considered.

In this thesis work, we adopt the semantic analysis in order to take into account additional informations that are not explicitly present in the input document. Such informations are extracted from external knowledge databases built by human. Over the past few years, several semantic-based approaches have been advanced. WordNet (Miller, 1995), is one of the most widely used thesauruses for English, and because of its semantic relations of terms, it has been heavily used to improve the quality of several NLP applications, such as automatic text summarization (Ferreira, 2014; Pal and Saha, 2014; Estiri et al., 2014), text clustering (Wei et al., 2015; Bouras and Tsogkas, 2012; Chen et al., 2010; Dang et al., 2013; and Fodeh et al., 2009),word sense disambiguation (Sachdevaet al., 2014; Dhungana et al., 2015) and other NLP tasks (Gao et al., 2015; Li et al., 2012; and Varelas et al., 2005).

In addition, redundancy is a key problem in automatic summarization due to the fact that sentences with similar meaning can be included in the summary because they have a high score. Simply and as other summarization systems, we can choose the sentences with the highest score to be included in the final summary. However, several information will be redundant in the summary, because many similar sentences representing the same idea in the document have similar score, so they can be included together in the summary. Moreover, some important sentences may not be included in the final summary and other ideas of the text can be ignored. Therefore, in our thesis work, instead of typically selecting top ranked sentences, we use a specific algorithm to form the final extractive summary by re-ranking all sentences and selecting the most relevant between them without redundancy.

Moreover, sentence ranking is a key problem in all extractive summarization methods. Much research has been done to improve the quality of this process. Some works used statistical features (Luhn, 1958; Ferreira et al., 2013b; Ferreira et al., 2014) and some approaches are based on graphs (Erkan and Radev, 2004; Mihalcea and Tarau, 2004; Baralis et al., 2013), while others adopted supervised and unsupervised machine learning techniques (Fattah, 2014; Yang et al., 2014; Alguliyev et al., 2015). After investigating these works, we have found that they rely on bag-of-words (BOWs) approach in sentence representation. BOWs representation can cause two main problems. First, the system does not have enough observing data in the training stage. Thus, traditional systems use a sparse word representation as input (Yousefi-Azar and Hamey, 2017), which mean that many of the values are zero. Second, the system is based on a large vocabulary, and data are represented in a high dimensional space, which mean that the performance of the system is decreased. Moreover, it has been shown that distributed representation of words outperforms BOWs representation in capturing the semantics of the input text. In our thesis work, we propose several unsupervised deep learning models for Arabic

TS. These models are used to learn latent features (the abstract representation) from a large Arabic corpus. These learned features are used to rank sentences instead of using the original BOW representation.

Table 1.5 Comparison between existing Arabic summarization systems

Reference	**Approach**	**Techniques used**	**Dataset**	**Evaluation**
Douzidia and Lapalme (2004)	Numerical	Sentence position, terms frequency, title words and cue words	Translation of DUC 2004 corpus	ROUGE
Sobh et al. (2007)	Numerical	Machine learning: Bayesian classifier, Genetic Programming classifier	Authors' corpus	Recall, precision, and F-measure
Fattah and Ren (2009)	Numerical	Machine learning: Probabilistic neural networks, Feed forward neural networks, Gaussian mixture, Mathematical regression, Genetic Programming	authors corpus and translation of DUC 2001 corpus	Precision, Recall, ROUGE-1
Boudabous et al. (2010)	Numerical	Machine learning: SVM classifier	Authors corpus	F-measure (with Recall and precision)
El-Haj et al. (2011a)	Numerical	Query-based and Concept-based	Authors corpus	Manual
El-Haj et al. (2011b)	Numerical	Clustering technique	DUC 2002 corpus and MT of DUC 2002	Precision, Recall, ROUGE-1
Azmi and Al-Thanyyan (2012)	Hybrid	Statistical features, RST	Authors corpus	ROUGE, recall, precision, and F-measure
Haboush et al. (2012)	Numerical	Root weight, Term frequency	Authors corpus	Recall and precision
Ibrahim and Elghazaly (2013)	Hybrid	RST VSM	Authors corpus	Precision
El-Fishawy et al. (2014)	Numerical	Similarity between tweets. Machine learning: Decision tree with linear regression	Authors corpus	F-measure, Normalized Discounted Cumulative Gain
Oufaida et al. (2014)	Numerical	Minimal-redundancy maximal-relevance (mRMR)	EASC corpus and TAC 2011 MultiLing Pilot corpus	ROUGE-1 and ROUGE-2
Belkebir and Guessoum (2015)	Numerical	Machine learning: SVM classifier AdaBoost	Authors' corpus	F-measure

Many existing Arabic text summarization methods adopted traditional supervised machine learning techniques (Boudabous et al., 2010; Sobh et al., 2007; Belkebir and Guessoum, 2015; El- Fishawy et al., 2014; and Fattah et al., 2009). These techniques suffer from two main problems. First, they need advanced domain knowledge and feature extractors to reduce the complexity of the data and make patterns more visible to the learning algorithm. In the context of Arabic, this kind of knowledge and tools are very limited and present a challenging and time-consuming task. Second, to build a powerful system, this kind of supervised approaches need a large labelled dataset (documents with their summaries), which is not available in the context of Arabic text summarization. Whereas, Deep Learning (DL) algorithms have not been studied enough for Arabic NLP including ATS. These techniques have proven their effectiveness in many domains. They have been successfully used in many computer vision applications and NLP tasks including text summarization. To the best of our knowledge, there exists no Arabic summarization system that integrate deep learning methods for generating summaries. Therefore, it may be beneficial to explore this kind of methods on Arabic text summarization. In our thesis work, we propose several unsupervised deep learning methods for Arabic TS.

On the other hand, document representation is an important phase in any machine learning method used in the context of NLP. This phase allows the conversion of text into numerical values, which are represented as input vectors to these kind of algorithms. In ATS, BOW is the most frequently technique used to transform the original text into numerical vectors. In the BOW model, documents (or sentences) in the corpus are represented by a matrix of vectors in which each row represents the document (or sentence) and each column corresponds to a word generated from the vocabulary of the corpus. The value associated with each row and column relies on metrics based on word frequency. This approach, despite its simplicity, it suffers from two main problems. Firstly, it provides a sparse data in a high dimensional vector space, which impact negatively the performance of the classifier. Secondly, the semantic relation between different text units is ignored and not captured by the BOW representation. However, choosing an adequate representation for Arabic documents has become critical to ensure a high quality and performance of any Arabic NLP tasks especially for ATS. As part of our contributions in this thesis work, we adopt a distributed representation of Arabic documents instead of using BOW representation.

Finally, after investigating the stat-of-the-art of Arabic ATS approaches, we found that the existing works do not consider the context of documents to be summarized. We assume that the summarization task can be improved if we take into account the key concepts presented in the text. For this, in our thesis work, an attempt is made to improve the proposed DL models by adopting the clustering technique and topic modeling approach. In more details, given a big Arabic corpus, a clustering technique is applied in order to group similar documents in the same cluster. Then, a topic modeling technique is applied on each set of documents grouped in the cluster to identify topics and terms belonging to each cluster. These generated topics are used to build a distributed vector representation of each document to be summarized.

1.9 Evaluation of Arabic text summarization methods

1.9.1 Datasets

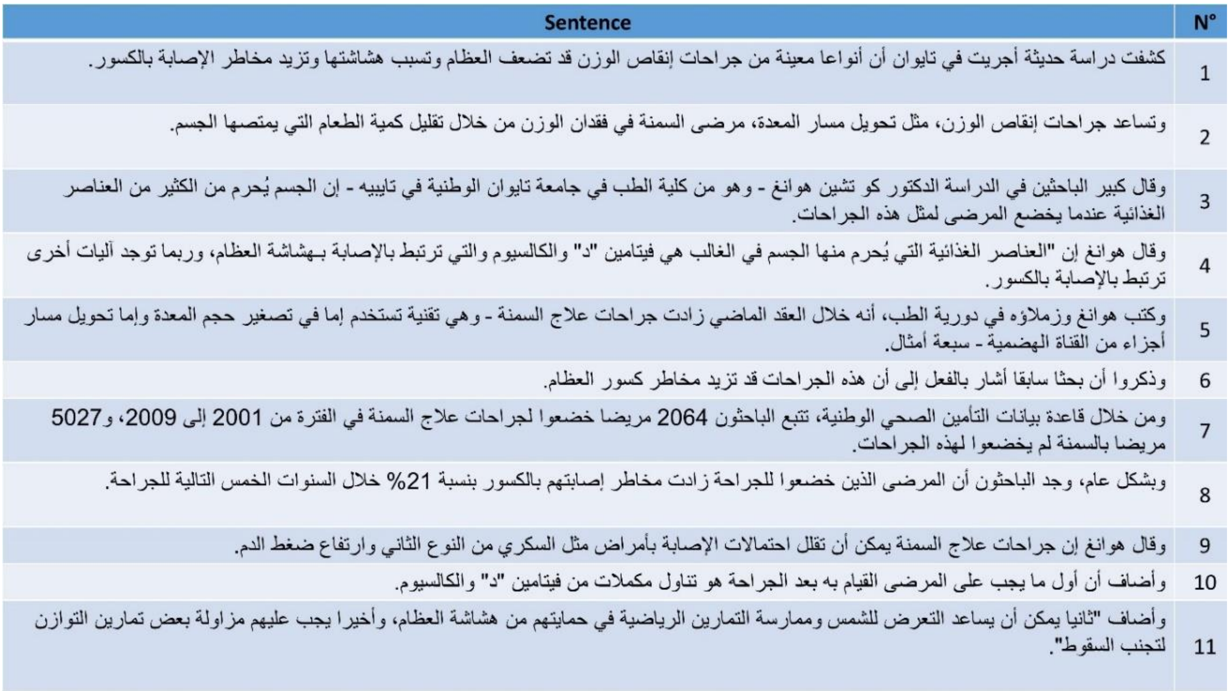
The performance of a summarization method is usually evaluated by comparing the results with the summary that was extracted manually. In Arabic language, several studies aimed at getting over the scarcity of the Arabic language in corpora. El-Haj, Kruschwitz, and Fox (2010) drew on Amazon's Mechanical Turk to build Essex Arabic Summaries Corpus (EASC). The dataset consists of 153 Arabic articles taken from two Arabic newspapers and the Arabic version of

Wikipedia. The dataset contains 10 main topics: science and technology, finance, health, environment, art and music, education, politics, religion, sports and tourism. The number of words in EASC is more than 51846. After applying the preprocessing step which consists of removing stop-words and stemming the remaining words using Khoja's stemmer (Khoja, 1999), the dataset is reduced to 40208 words comprising 4195 unique words. The number of sentences is 2293 with an average of 14 sentences per document. The average word is 338 per document and 22 per sentence. For each document, five model extractive summaries are available. These reference summaries were created by native Arabic speakers using Mechanical Turk. Each user is asked to include in the human-generated summary, a set of sentences close to the meaning of the document without exceeding 50% of the source document's size. The dataset is produced in two different encoding formats: ISO-Arabic and UTF-8.

Furthermore, we have developed our own corpus of Arabic articles covering various topics. We have collected a sample of 42 articles of a number of news articles and blogs from three popular Arabic newspapers: Al Jazeera (www. aljazeera.net), Al Arabiya (www.alarabiya.net) and Hespress (www.hespress.com). These websites were selected because of: (i). their large circulation: as electronic papers, they are popular and read on a large scale; (ii). They are written in genuine Arabic text by native speakers active in different sectors. The sample document covers various topics (health, religion, business and politics) in different sizes. The number of sentences is 647 with an average of 15 sentences per document. The average number of words per document is 441 and per sentence is 28. For each document, one manual summary is generated by fluent speaker of Arabic language.

Nonetheless, no standard dataset exists nowadays for Arabic that is applied in the evaluation of the Arabic text summarization task. Table 1.5 shows different metrics and datasets used in the evaluation of existing Arabic summarization systems. In this thesis work, we used the two datasets described above so as to assess our proposed models on Arabic documents. Figure 1.3 shows a sample Arabic text used in the evaluation process.

To confirm our results obtained with the above Arabic datasets, we also use the SKE dataset written in English. It is composed of a sets of emails extracted from Enron mailboxes and 30 were provided by volunteers. There are 349 emails divided into single and threads emails. The minimum number of sentences in a single email is 10, and the minimum number of emails in a thread email is 3. Single emails are of at least 10 sentences and threads email are at least 3 emails. The number of words in the dataset is more than 100,000 words. After removing stop-words and applying stemming with Porter 1980, the dataset is reduced to 46,603 comprising 7478 unique words. The number of sentences in SKE is 6801. The average of words per email is 303 and the average of sentences per email is 19.5. The average words per sentence is 15.5. SKE is comes with two annotators for each email representing a set of key phrases and a summary generated by human. The first annotator is an abstractive summary which contains between 33 and 96 words, while the second annotator is an extractive summary which identified the best 5 sentences ranked as a summary of the email. In this work, we investigate two summarization approaches on the SKE dataset. The first approach consists of generating the system summary based on graph theory. The second approach is to use the email subject as the user input query to the system. In this case, the summary is generated from the most relevant (similar) sentences to a given query (email subject).



Sentence	N°
كشفت دراسة حديثة أجريت في تايوان أن أنواعا معينة من جراحات إنقاص الوزن قد تضعف العظام وتسبب هشاشتها وتزيد مخاطر الإصابة بالكسور.	1
وتساعد جراحات إنقاص الوزن، مثل تحويل مسار المعدة، مرضى السمنة في فقدان الوزن من خلال تقليل كمية الطعام التي يمتصها الجسم.	2
وقال كبير الباحثين في الدراسة الدكتور كو تشين هوانغ - وهو من كلية الطب في جامعة تايوان الوطنية في تايبيه - إن الجسم يُحرم من الكثير من العناصر الغذائية عندما يخضع المرضى لمثل هذه الجراحات.	3
وقال هوانغ إن "العناصر الغذائية التي يُحرم منها الجسم في الغالب هي فيتامين "د" والكالسيوم والتي ترتبط بالإصابة بـهشاشة العظام، وربما توجد آليات أخرى ترتبط بالإصابة بالكسور.	4
وكتب هوانغ وزملاؤه في دورية الطب، أنه خلال العقد الماضي زادت جراحات علاج السمنة - وهي تقنية تستخدم إما في تصغير حجم المعدة وإما تحويل مسار أجزاء من القناة الهضمية - سبعة أمثال.	5
وذكروا أن بحثا سابقا أشار بالفعل إلى أن هذه الجراحات قد تزيد مخاطر كسور العظام.	6
ومن خلال قاعدة بيانات التأمين الصحي الوطنية، تتبع الباحثون 2064 مريضا خضعوا لجراحات علاج السمنة في الفترة من 2001 إلى 2009، و5027 مريضا بالسمنة لم يخضعوا لهذه الجراحات.	7
وبشكل عام، وجد الباحثون أن المرضى الذين خضعوا للجراحة زادت مخاطر إصابتهم بالكسور بنسبة 21% خلال السنوات الخمس التالية للجراحة.	8
وقال هوانغ إن جراحات علاج السمنة يمكن أن تقلل احتمالات الإصابة بأمراض مثل السكري من النوع الثاني وارتفاع ضغط الدم.	9
وأضاف أن أول ما يجب على المرضى القيام به بعد الجراحة هو تناول مكملات من فيتامين "د" والكالسيوم.	10
وأضاف "ثانيا يمكن أن يساعد التعرض للشمس وممارسة التمارين الرياضية في حمايتهم من هشاشة العظام، وأخيرا يجب عليهم مزاولة بعض تمارين التوازن لتجنب السقوط".	11

Figure 1.3. A randomly selected sample from EASC corpus

1.9.2 Evaluation metrics

A suitable evaluation techniques are needed to evaluate the quality of any summarization system. In this thesis work, we have used three important measures in the assessment of our system's performance: precision, recall and F-measure. Precision (P) measures the amount of correct information returned by the system. It corresponds to the number of correct summary sentences of the system divided by the number of its summary sentences. Recall (R) measures the system's coverage. It reflects the ratio of relevant sentences extracted by the system. It equals the number of correct summary sentences of the system, divided by the aggregate number of human generated summary sentences. These two measures are antagonistic in that a system striving for coverage will obtain lower precision, and lower recall will be the result of a system striving for precision. F-measure (F) strikes a balance between the first two measures using a parameter β, and calculated by the following formula:

$$F = (\beta^2 + 1) * P * R / (\beta^2 * P + R) \quad (1.12)$$

The (F-Measure/summary size) ratio is significant in the comparison of systems. The F1 score is obtained by setting the value of β to one. More formally, the three measures are represented by the following formulas:

$$P = \frac{|S_{manual} \cap S_{auto}|}{|S_{auto}|} \quad (1.13)$$

$$R = \frac{|S_{manual} \cap S_{auto}|}{|S_{manual}|} \quad (1.14)$$

$$F = \frac{2 * P * R}{P + R} \quad (1.15)$$

Where S_{manual} is the set of sentences in the summary generated manually and S_{auto} represents the set of sentences in the summary generated by the system.

In addition, to evaluate our proposed models, we used the well-known automatic evaluation method ROUGE (Recall-Oriented Understudy for Gisting Evaluation) as advanced by (Lin, 2004). ROUGE is an extensively used set of automatic evaluation metrics that allows us to make an intrinsic evaluation of automatic text summaries against human-made abstracts. It is of great importance in judging the quality of any summary. ROUGE has been adapted by DUC since DUC 2004. The main reason behind ROUGE is to enumerate the number of text units overlaps between the candidate summary and other human generated summaries. It consists of a package that includes several metrics, such as N-gram Co-Occurrence Statistics (ROUGE-N), Longest Common Subsequence (ROUGE-L), Weighted Longest Common Subsequence (ROUGE-W), and Skip-Bigram Co-Occurrence Statistics (ROUGE-S). Formally, ROUGE-N (N=1 in our experiments) is an n-gram recall measure between a system generated summary (i.e. candidate summary) and a set of human generated summaries (i.e. reference summaries). This measure evaluates the summary by computing the n-gram recall between the summary itself and the set of references summaries. ROUGE-N is given by the following formula:

$$\frac{\sum_{S\in\{ReferenceSummariez\}}\sum_{gram_n\in S} Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummariez\}}\sum_{gram_n\in S} Count(gram_n)} \tag{1.16}$$

Where n stands for the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in both candidate summary and a set of reference summaries, and Count is the total number of n-gram in the reference summaries. Before applying, ROUGE, several language-dependent preprocessing steps must be applied (Lloret and Palomar, 2012). In this work, we applied the stop word removal process before calculating the ROUGE score.

1.10 Conclusion

In this chapter, we have presented an overview of the most recent advances and challenges of automatic summarization raised in the last years. First, we have explained the new approaches proposed in automatic summarization of texts in different languages and especially in English texts. After that, we examined the characteristics of the Arabic language and the proposed techniques for Arabic automatic language processing. Next, we have presented existing works and recent advances on automatic summarization of Arabic texts. Then, we have synthetized our study by showing some limitations of the existing Arabic summarization methods. Finally, we have described the datasets and metrics used for the evaluation of the models we proposed in this PhD thesis.

Arabic, which is the official language of over 250 million peoples and the second for 40 million, deserves much more interest by researchers due to the lack of works in NLP area in general and in ATS in particular. For this, we propose in our thesis work to develop and implement a new methods for automatic summarization designed for texts in Arabic.

The next chapters (chapter 2, chapter 3, chapter 4, and chapter 5) are dedicated to provide a more detailed presentation of our contributions and proposed methods for Arabic text summarization.

Chapter 2

Hybrid approach using statistical and semantic analysis for Arabic Text summarization

2.1 Introduction

In the previous chapter, we detailed the state-of-the-art in ATS. We focused our study on the most well-known approaches and proposed works. We also investigated a detailed study on existing works on Arabic text summarization. Following this study, carried out among several works, research on Arabic Text summarization is still in its early beginning and the literature that addresses this subject area in Arabic is fairly small and only recently compared to that on Anglo-Saxon, roman and other Asian languages. Moreover, summarization systems for Arabic have not reached the same level of maturity and reliability as those for English, for instance. Arabic Text summarization is not studied enough compared to the large number of studies carried out for English. This is because the scientific community devotes a little attention for Arabic than other language. Therefore, and regarding the importance of this language, there is a considerable opportunity for further research in this field, and the need to develop methods and techniques that perform the summarization of Arabic documents is growing significantly, especially when the existing systems are note mature enough and efficient as we need. The main goal of this thesis work is to enrich the existing state-of-the-art of Arabic summarization systems with other advanced approaches more efficient and more powerful.

Most research works in Arabic document summarization use statistical approaches for extracting sentences. In these kind of approaches, ranking sentences is one of the most important phase in any summarization system. Existing Arabic summarization systems usually use statistical features, such as *TF*, *TF.IDF*, sentence position, similarity with title etc. to rank sentences. The main problem with this kind of systems is that they rank sentences without taking into account the relationships between them.

In addition, and as mentioned in chapter 1, traditional Arabic summarization systems do not address the redundancy and the information diversity issues. When producing the summary of the input text, the likelihood of having similar sentences in the output is much higher, so similar sentences can appear together in the summary because they have a similar score. Further, other ideas of the document may not be selected and relevant information may be overlooked and accordingly not included in the final summary that is supposed to encapsulate the maximum amount of information from the input text. Hence, redundancy elimination has become crucial in text summarization (Atkinson and Munoz, 2013) and especially in Arabic, with the aim of diversifying the information included in the summary. In our proposed system, we deal with this problem by using a MMR algorithm proposed by Carbonell and Goldstein (1998). Therefore, in this thesis work, instead of typically selecting top ranked sentences, we propose to use a specific algorithm to form the final extractive summary without redundant information.

Moreover, according to the state-of-the-art, traditional Arabic summarization systems are based on statistical approaches. One of the obvious disadvantages of this approach is that it cannot accurately represent the meaning of documents, because it ignores the semantic relationship existing between different textual units. The system is always limited to the words explicitly mentioned within the input text document. For example, if the system is not able to find the relationships between terms like « بترول » (Petroleum) and « نفط » (Oil), it would handle these words separately as two different unrelated terms, and this may affect negatively their importance in the input document.

Seeking to overcome these drawbacks, we propose in this chapter an extractive summarization system for Arabic documents. We describe our own improvements on some important aspects of Arabic summarization systems, including sentence ranking by means of a graph theory and sentence selection via Maximal Marginal Relevance algorithm.

We propose a new graph-based Arabic summarization system that combines statistical and semantic analysis. The proposed approach utilizes ontology hierarchical structure and relations to provide a more accurate similarity measurement between terms in order to improve the quality of the summary. The proposed method is based on a two-dimensional graph model that makes use of statistical and semantic similarities. The statistical similarity is based on the content overlap between two sentences, while the semantic similarity is computed using the semantic information extracted from a lexical database whose use enables our system to apply reasoning by measuring semantic distance between real human concepts.

First, we develop (sentence ranking) an Arabic Text summarization system based on graph model. Recently, graph-based ranking algorithms, motivated by the PageRank algorithm (Page and Brin, 1998), have shown their effectiveness in text summarization (Erkan and Radev, 2004; Nguyen-Hoang et al., 2012; Baralis et al., 2013). To construct a graph, a node needs to be added for each sentence in the text, and the edges between nodes are established through sentence inter-connections that are defined by their relationships. The underlying assumption is that more important sentences are likely to have more relationships with other sentences.

Second, we introduce a new graph-based Arabic summarization system that combines statistical and semantic analysis to achieve the summarization task, and to avoid the usual problems of this task: information redundancy and diversity. Generally speaking, when the summarization process is done by humans, they read the text first and understand it using some basic level of background knowledge. In this work, an attempt is made to enable the proposed system to understand the relationships between different textual components of an Arabic document, by providing some human constructed knowledge repositories and integrating concepts from a wide range of areas. Traditional Arabic summarization methods do not take into account the semantic relationships among words so they cannot represent the meaning of documents accurately. To solve this issue, several languages like English have resorted to the introduction of semantic information from ontologies to improve the quality of text summarization. Several semantic-based approaches have been advanced. WordNet (Miller, 1995), is one of the most widely used thesauruses for English, and because of its semantic relations of terms, it has been heavily used to improve the quality of several NLP applications, such as automatic text summarization (Ferreira, 2014; Pal and Saha, 2014; Estiri et al., 2014), text clustering (Wei et al., 2015; Bouras and Tsogkas, 2012; Chen et al., 2010; Dang et al., 2013; and Fodeh et al., 2009), word sense disambiguation (Sachdevaet al., 2014; and Dhungana et al., 2015) and other NLP tasks (Gao et al., 2015; Li et al., 2012; and Varelas et al., 2005).

The proposed system works as follows: In the first step, it proceeds to document preprocessing. Second, it computes the similarity between each pair of sentences. For this, two types of similarity are adopted: (i) statistical similarity based on the overlap of content between two sentences, (ii) semantic similarity measure based on the semantic information extracted from Arabic WordNet (AWN). If an Arabic word does not exist in the AWN ontology, the proposed algorithm uses a machine translation from Arabic to English and interrogates the WordNet ontology so as to calculate the semantic similarity. By using text representation, the proposed algorithm aims at converting the text into a graph model with a two-fold relation between sentences: statistical similarity and semantic similarity. For this purpose, we make a two-dimensional graph representation of the input document. The graph represents two kinds of similarity: statistic similarity and semantic similarity between sentences. All sentences are ranked according to their importance in the graph. Subsequently, the weighted ranking algorithm PageRank (Brin and Page, 1998) is executed on the graph to produce relevant score for each sentence in the input text. The top- ranking sentences are identified to form the final summary for the input document and an adapted MMR algorithm version (Carbonell and Goldstein, 1998) is applied to eliminate unneeded information and enhance the quality of the final summary. Experimental results on EASC and our own datasets (described in section 1.9.1) showed the effectiveness of our proposed approach over existing summarization systems.

On the other hand, stemming is a process of reducing inflected words to their stem or root from a generally written word form. This process is used in many text mining application as a feature selection technique. Therefore, one of the main objectives of this chapter is to evaluate the impact of three different Arabic stemmers (i.e. Khoja, Larekey and Alkhalil’s stemmer) on the text summarization performance for Arabic language. The evaluation of the proposed system, with the three different stemmers and without stemming, on the dataset used shows that the best performance was achieved by Khoja stemmer in term of recall, precision and F1-measure. The evaluation also shows that the performances of the proposed system are significantly improved by applying the stemming process in the pre-processing stage.

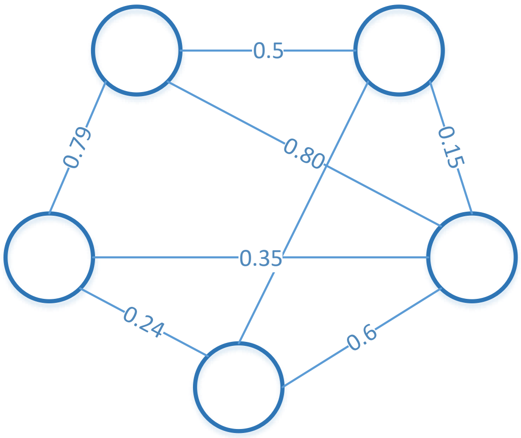
2.2 Graph theory and summarization

Recently, graph-based algorithms have been applied successfully to different NLP tasks. Methods for deciding term importance have a very strong mathematical base. In Graph-based ranking algorithms, the process of deciding a textual unit’s importance has become very popular. Graph-based ranking algorithms are a way to decide the relative importance of a node within the graph. These algorithms take into consideration the global information, i.e., the whole graph, when deciding the importance of a node and not just the local, vertex-specific information. A text represented with a graph, interconnects sentences or other parts of a text with meaningful relations. Using the graph for displaying the structure of the text will help us to better understand the connection between different parts. Graph-based algorithms use a ranking algorithm to rank different sections of a text where each section is considered as a node. Edges will represent the lexical or semantic relations between two nodes. Regardless of the type and characteristics of the text that we want to draw the graph for, a graph-based ranking algorithm includes the following basic steps:

a. Identify text units that best define the task, these units can include sentences, words or other units and to consider them as vertices in the graph.

b. Identify relations that connect such text units, and use these relations to draw edges between vertices in the graph. Edges can be directed or undirected, weighted or unweighted.

c. Run the graph ranking algorithm to find a ranking over the nodes in the graph until convergence. Then all nodes are sorted according to their final score. Use the values attached to each vertex for ranking/selection decisions.



For text summarization task, given a document *d*, let *G=(V,E)* be an undirected graph represent the document *d* with the set of nodes *V* and set of edges *E*. Under this model, the nodes represent the sentences in *d*. Each edge E_{ij} has weight w_{ij} that indicates the similarity between nodes (Sentences) v_i and v_j. Two sentences are connected if and only if they are similar to each other and must satisfy a similarity threshold *t*. Each node in *V* is also labeled with their salient score based on the relations with others connected nodes. This score, computed by ranking algorithm, illustrates the amount of information that a sentence contains.

Figure 2.1 An example of text representation by graph model

As it is stated in the third step, after specifying the final score for each node, the nodes are ranking based on their final score. Then, sentences with the highest score are selected to attend in final summary. Figure 2.1 illustrates an example of text representation using graph model. For text summarization, the text units are represented as vertices and their interconnecting relationships as the edges. TexRank (Mihalcea and Tarau) and LextRank (Erkan and Radev, 2004) are the most important algorithms based on the graph model. Both of them use PageRank algorithm to compute relative importance of sentences. In (Ribaldo et al., 2012), a hybrid graph-based method was presented annotating relationship maps with cross-document Structure Theory, and using network metrics. It helped for Portuguese multi-document summarization. Nguyen-Hoang et al. (2012) developed a graph-based summarization system for Vietnamese documents. In addition, SUMGRAPH (Patil and Brazdil, 2007) and Time stamped Graph (Lin, 2007) are two automatic summarization systems based on graph theory.

2.3 A new Arabic text summarization system

The Arabic summarization method propounded in this work builds on statistical methods and semantic processing with the aim of increasing information diversity of the output summaries, addressing redundancy and considering the semantic relationships among words. Our proposed system has many advantages:

- It needs no training data or annotated corpus.
- It is domain-independent, which means that we do not need to take domain-specific knowledge into account.
- It is knowledge-rich: unlike the existing methods, the semantic relationships among words are considered, and the semantic information from a man-developed knowledge database system is introduced so as to accurately represent the meaning of documents and improve the quality of Arabic text summarization.
- It deals with a lack of semantic resources dedicated to Arabic by using a machine translation between Arabic and English to benefit from the richness and opportunities offered by the English language in this field.
- It uses a graph model based on both statistical and semantic similarities to make a two-dimensional graph representation of the input text as a set of sentences linked by meaningful relations.
- The proposed system uses the MMR method to address redundancy and information diversity problems.

2.3.1 System architecture

Figure 2.2 illustrates the general architecture of our proposed method. A large Arabic document constitutes the system input while the output is a small Arabic document containing the major ideas of the original text. The following section explains in detail each stage of the proposed system. The advanced method consists of the three main steps below:

a) Pre-processing: The aim of this step is to prepare the original text for the later steps. It consists of tokenizing the input document, removing stop-words so as to reduce the size of the input document, and finally extracting the root of each word.

b) Analysis: In this stage, a number of statistical features are extracted, the measure of the semantic and statistical similarity within each sentence of the original text is computed, and a two dimensional graph is built representing the input document. The ranking algorithm is then performed on a resulted graph in order to rank each sentence against others. The final score of each sentence is computed by adding the weight of the extracted statistical features.

c) Post-processing: The goal of this stage is to generate the final summary according to the score of each sentence and by avoiding information redundancy.

2.3.2 Preprocessing phase

The preprocessing phase consists of cleaning the source documents, as well as splitting and tokenizing the sentences. In our system, the sentence is the extraction unit and the term is considered as a scoring unit. We implement this phase in three steps:

2.3.2.1 Tokenization

The Tokenization process consists of dividing the text into tokens. The input text is normalized through two steps: first, all punctuations, non-letters and diacritics are removed, secondly some characters are replaced by the normalized ones (أ, آ, and إ with ا and last ى with ي and last ة with ه). In our system, and depending on the datasets used, we consider the character "." as a sentences separator and the character " " (space) as a word separator. This consideration makes the splitting process easy in order to segment the text document into sentences and each sentence into words.

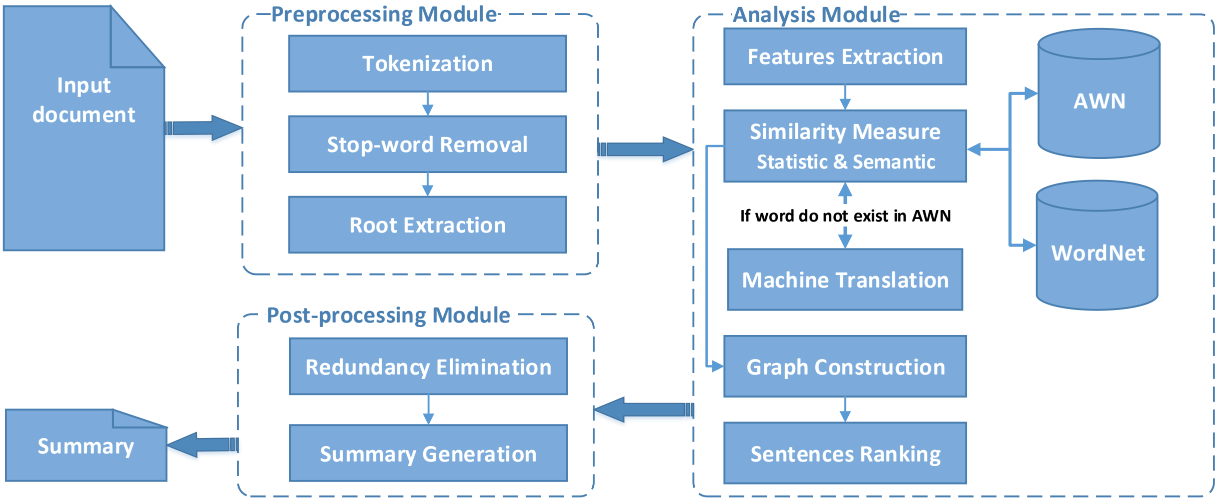


Figure 2.2 The main steps of the proposed Arabic Summarization system

2.3.2.2 Stop words removal

Stop-words are very common words with a mainly structural function; they are recurrently used in a text, carry little meaning and their function is syntactic only. They do not indicate the subject matter and do not add any value to the content of their documents. In Arabic, words like (هو, هذا, الذي, هي) are frequent in sentences; but with little significance in the implication of a document. These words can be deleted from the text to help identify the most meaningful words in the summarization process. There is no typical list of stop-words specific to the Arabic text; this is why we simply use, in this work, a list of 168 words proposed by (Khoja, 2001).

2.3.2.3 Root extraction

Words in Arabic are generally derived from a root, which is indeed a base for diverse words with a somehow related meaning. A set of derivations representing a same area can be constructed by adding suffixes to the root. Identifying a root of an Arabic word (stemming) helps in its grammatical variations mapping to the instances of the same term. As shown in Table 2.1, all the three Arabic words "كتب", "كتاب" and "مكتبة" are related to the same root "كتب". This amounts to saying that multi-derivations of the Arabic wording structures make the semantic representation of the text possible. The quality and performances of a text summarization task may be positively impacted by an adequate representation of Arabic text. Moreover, since words sharing the same root have a semantic relation, using this root in features selection can improve the accuracy of similarity measure and frequency analysis in Arabic text because the words in a text can have more than one occurrence, but in different forms.

It is to be noted that determining the root of any Arabic word is a difficult task as the text has to undertake a detailed morphological, syntactic and semantic analysis. Word stemming is considered as one of the most difficult problems in Arabic. A wide body of research has been carried out in this field. We used the Khoja stemmer presented by (Khoja, 1999), which is a root-based stemmer which extracts the roots by using pattern matching and removing affixes.

Table 2.1 different derivation of root kataba (“كتب”)

Arabic word	English sense	Root
كتب	Write	كتب
كتاب	Book	كتب
مكتبة	Library	كتب

2.3.3 Analysis phase

After preprocessing the input Arabic document, the analyzing phase begins scoring the sentences based on the computed set of features. Each sentence is given two kinds of scores: statistical score and semantic score. This section describes the chosen statistical features, and how the semantic and statistical similarities between two sentences are computed. The main steps of this phase are:

- Extract statistical features for each sentence.
- Calculate the statistical similarity measure between each pair of sentences.
- Calculate the semantic similarity measure between each pair of sentences.
- Build a two-dimensional graph based on both statistical and semantic similarities.
- Apply a random walk on a graph using a PageRank algorithm. All the sentences are then ranked according to semantic and statistical relationships between them.
- Compute the final score of each sentence.

2.3.3.1 Statistical features

TF-IDF is a statistical measure often used in text mining and information retrieval. It evaluates how important a term is to a document in a collection or corpus. It is obtained by multiplying the term frequency (TF) and the inverse document frequency (IDF). TF represents the number of times the term appears in the document. The assumption here is that the word becomes more important when its number of occurrence in a document is high. IDF represents the importance of a term in a document collection or corpus. It strikes a balance for local frequencies that are likely to increase the importance of a certain term simply because of its high frequency in a single document. IDF is obtained by comparing the number of documents containing the term with the whole number of documents in the corpus. In this work, we use the Inverse Sentence Frequency (ISF) measure which is the same as IDF, where a set of sentences substitutes for a set of documents. The inverse sentence frequency (ISF) measures the importance of a term within the sentence collection. The formula below computes the ISF of each word in the sentence (Alguliyev et al., 2015):

$$ISF_{wi} = \log_2 \frac{N}{df_{w_i}} \tag{2.1}$$

Where N represents the number of sentences in the document and df_{wi} represents the number of sentences where the word wi appears.

2.3.3.2 Statistical similarity

The statistical similarity presents the number of similar words shared between two sentences. Formally, the statistical similarity measure between two sentences S_i and S_j as described in (Mihalcea and Tarau, 2004) is:

$$Similarity(S_i, S_j) = \frac{|\{w_k | \, w_k \in S_i \cap S_j\}|}{\log(|S_i|) + \log(|S_j|)} \tag{2.2}$$

Where $|S_i|$ represents the length of the sentence S_i.

Other sentence similarity measures are also possible and could be interesting, such as Euclidean Distance, Jaccard Coefficient, Cosine similarity, etc. The measure presented in Equation (2.2) is shown to be a simple and fast alternative to other similarity measures.

2.3.3.3 Semantic similarity

Semantic similarity is becoming more and more popular and has a significant role in different NLP tasks such as information retrieval, information extraction, text summarization, text clustering and so on. That two sentences (Or two documents) do not have common terms does not necessarily mean that the sentences are not semantically related (Varelaset al, 2005). Semantic similarity involves measuring the relationship between lexicographically dissimilar concepts. Still, two terms can have semantic similarity (e.g., can be synonyms or have similar meaning) despite their lexicographic difference (Varelas et al, 2005). Therefore, summarization by using only classical methods will not be able to recover sentences (Or documents) with semantically similar terms. This is one of the problems we addresses in this work.

Several semantic similarities have been proposed to quantify the semantic similarity based on ontology hierarchy. Some utilize the taxonomies within WordNet and the relations defined between its units. WordNet (WN) is the hierarchically-structured repository that was created by linguistic experts and its richness stems from its lexical relations that are explicitly defined. This kind of measure based on WN has been used on a wide scale in NLP applications (Pedersen, 2010). Arabic WordNet (AWN) builds on Princeton WordNet to provide a lexical resource for standard modern Arabic.

In this work, we have integrated two lexical databases, WN and AWN, to take into consideration the semantic relationships between terms, and provide a more accurate similarity measurement between sentences.

Words semantic similarity

In this work, we have adopted the concepts based representation model to calculate the semantic similarity between terms. In the concept-based representation of the text, each term is replaced by its associated concepts in AWN. Two stages are required to allow such representation:

i) The first is the projection of the terms into concepts, and each term in this stage is substituted by the corresponding vector of concepts (not including the words that do not appear in AWN).

ii) The second stage is the application of a disambiguation strategy in order to assign one concept to one term, and avoid the loss of information caused by the replacement of a term by a list of concepts. In our approach, we have adopted the “First concept” strategy as a simple disambiguation method. AWN provides for every word a list of ordered concepts, arranged from the most appropriate to the least appropriate concept. This disambiguation strategy focuses only on the first concept of the list as the most appropriate concept (Elberrichi and Abidi, 2012). Table 2.2 and Table 2.3, show the projection of two terms (“منزل”, “بيت”), and their associated concepts in AWN.

Table 2.2 Mapping of term “manzil”/“*منزل*” in Arabic WordNet

Concept Id	**Concepts in Arabic WordNet**
manozil_n1AR	[مَنْزِل, بَيْت]
masokan_n1AR	[سَكَن, مَسْكَن, مَقَرّ, مَنْزِل, بَيْت]
sakan_n2AR	[وَطَن, سَكَن, مَسْكَن, مَسْكَن, مَرْكَز, مَقَرّ, مَقَام, مَنْزِل, بَيْت, حِلَّة]
sakan_n3AR	[سَكَن, مَسْكَن, مَنْزِل]
manozil_n2AR	[مَنْزِل, بَيْت, عَائِلَة, أُسْرَة]

Table 2.3 Mapping of term “bayt”/ “*بيت*” in Arabic WordNet

Concept Id	**Concepts in Arabic WordNet**
manozil_n1AR	[مَنْزِل, بَيْت]
bayot_n2AR	[مَكَان, بَيْت]
bayot_n1AR	[بَيْت]
masokan_n1AR	[سَكَن, مَسْكَن, مَقَرّ, مَنْزِل, بَيْت]
sakan_n2AR	[وَطَن, سَكَن, مَسْكَن, مَسْكَن, مَرْكَز, مَقَرّ, مَقَام, مَنْزِل, بَيْت, حِلَّة]
bayot_n3AR	[دَار, بَيْت]
manozil_n2AR	[مَنْزِل, بَيْت, عَائِلَة, أُسْرَة]
bayot_n4AR	[دَار, بَيْت]

In this work, the focus is on using the Wu and Palmer measure proposed by Wu and Palmer (1994) when computing the semantic similarity between any two concepts. This measure was found to be simple to calculate and presents more performances while remaining competitive and expressive as other similarity measures (Lin, 1998). This is why we have adopted this measure as a base of our work. The Wu and Palmer (1994) measure calculates a similarity measure between concepts by examining the depths of the two terms in the ontology, together with the depth of the least common subsume (LCS) node that connects their senses. This measure is based on path lengths (in number of nodes), common parent concepts, and distance from the hierarchy root (Wei et al., 2015).

An example is given in Figure 2.3 which represents an ontology constituted by a number of nodes and a root node (Root). *Concept_Xi* and *concept_Yj* correspond to two ontology elements for which the similarity will be calculated. This similarity measure considers the distance (*N1* and *N2*) which separates nodes *concept_Xi* and *concept_Yj* from the hierarchy root and the distance (*N*) which separates the most specific common concept (the common parent related with the minimum number of IS-A links with the two concepts) of *concept_Xi* and *concept_Yj* from the node Root. In the given example, the *LCS* of *concept_Xi* and *concept_Yj* is the node *concept_LCS* that represents the lowest common node between the paths of these two senses from the root of WordNet hierarchy. Once the *LCS* has been found, the distance between two senses is defined by the following formula:

$$sim(X,Y) = \frac{2*N}{N1+N2} \tag{2.3}$$

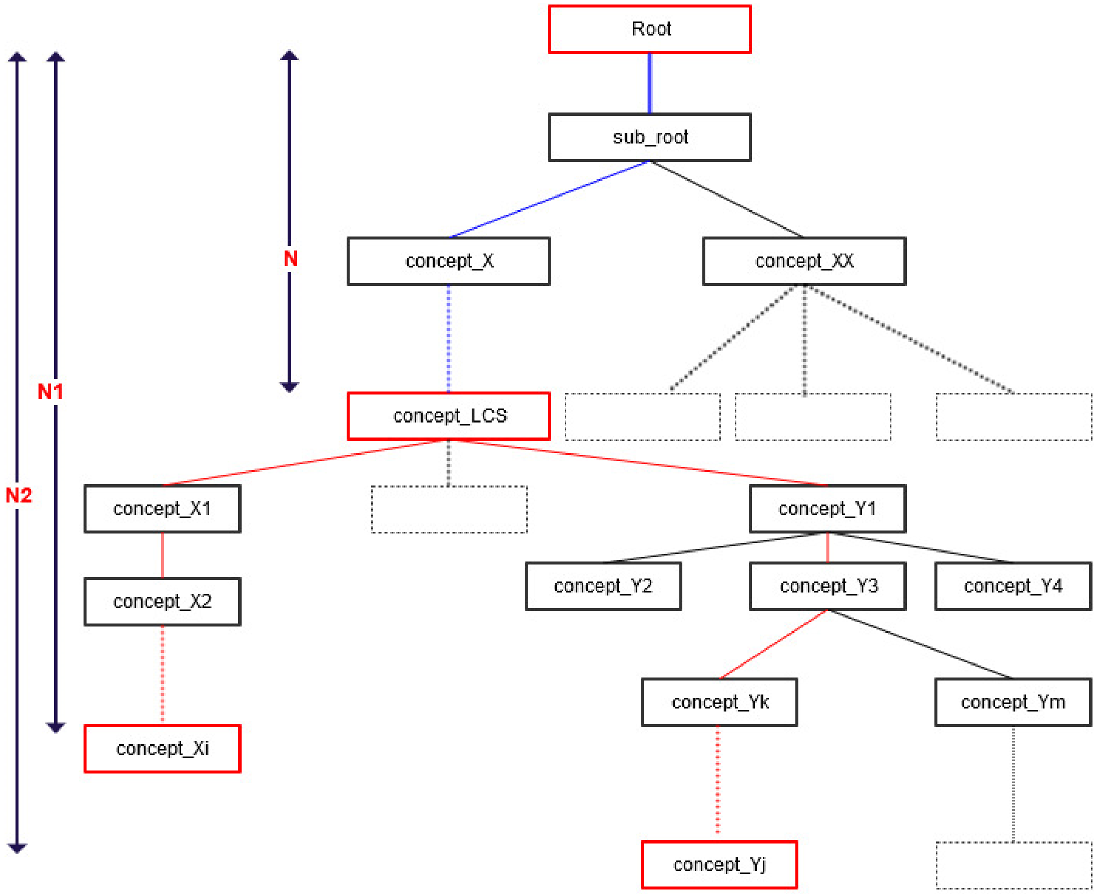


Figure 2.3. Example of a concept hierarchy

We should point out that in the case where one of the two terms does not appear in Arabic WordNet, we use a machine translation and we interrogate the English WordNet ontology to compute their semantic similarity. Figure 2.4 illustrates the flowchart of the semantic similarity measure process between two Arabic words wi and wj.

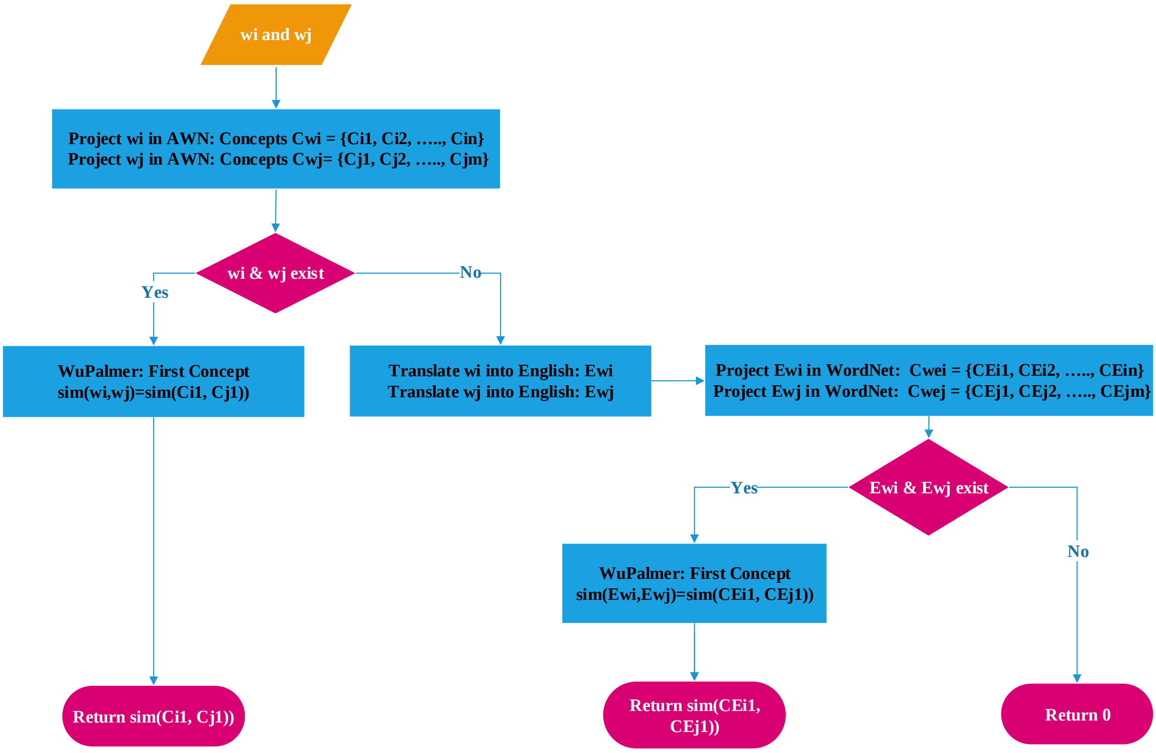

wi and wj
Project wi in AWN: Concepts Cwi = {Ci1, Ci2, ….., Cin}
Project wj in AWN: Concepts Cwj= {Cj1, Cj2, ….., Cjm}
wi & wj exist
Yes
No
WuPalmer: First Concept
sim(wi,wj)=sim(Ci1, Cj1))
Translate wi into English: Ewi
Translate wj into English: Ewj
Project Ewi in WordNet: Cwei = {CEi1, CEi2, ….., CEin}
Project Ewj in WordNet: Cwej = {CEj1, CEj2, ….., CEjm}
Ewi & Ewj exist
Yes
No
WuPalmer: First Concept
sim(Ewi,Ewj)=sim(CEi1, CEj1))
Return sim(Ci1, Cj1))
Return sim(CEi1, CEj1))
Return 0

Figure 2.4. Semantic similarity measure between two Arabic words wi and wj

Sentences semantic similarity

The semantic similarity between each pair of sentences is computed using a measure proposed by Malik et al. (2007). This measure is computed by summing the maximum scores of all words similarity divided by the sum of the sentence length. First, each sentence is represented as a word vector, and then the semantic similarity for each pair of words of the given sentences is computed based on Equation (2.3). Equation (2.4) defines the semantic similarity formulation between two sentences:

$$Sim(S_i,\ S_j) = \frac{\sum_{w\in S_i} \max Sim_w(w, S_j) + \sum_{w\in S_j} \max Sim_w(w, S_i)}{|S_i|+|S_j|} \qquad (2.4)$$

In this equation: S_i and S_j are the given sentences; $\max Sim_w(w,\ S_j)$ represents the maximum similarity scores of the word w and all the words in S_j; and $|S_i|$ represents the length of the sentence S_i.

2.3.3.4 The need of a machine translation

One of the particularities of our system is dealing with a lack in semantic resources dedicated to the Arabic language. We use a machine translation between Arabic and English to benefit from the richness and opportunities offered by the English language in terms of automatic linguistic resources. The semantic similarity measurement between two Arabic words is calculated using Arabic WordNet. Arabic WordNet is a very important lexical resource for Arabic; it is currently under construction, and is not as mature as its correspondent in English. That is why in this work we propound the integration of the English WordNet ontology and use it in the computation of the two words semantic similarity if one of them does not exist in Arabic WordNet. Both of the given words will be translated to English and the similarity will then be calculated according to English WordNet.

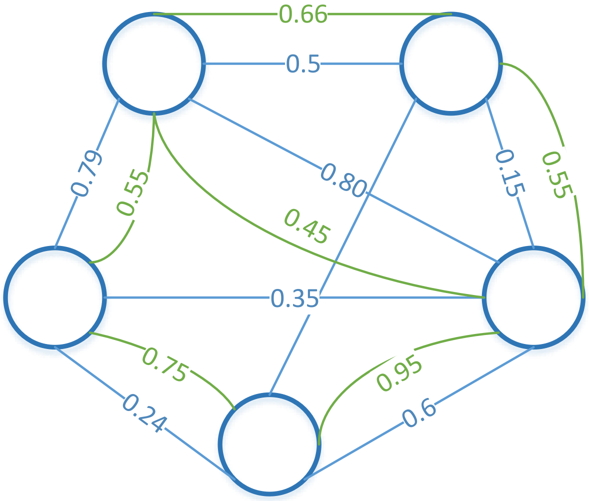
2.3.3.5 Graph construction

Here, we convert the input Arabic document into graph format. To draw the graph, we need to find textual units that best describe the task of automatic summarization and consider them as nodes of the graph. Then, we need to identify relations that connect those units. In this work, we consider the sentences of the input Arabic document as a text unit and the similarity between those sentences as a relation between them.

As shown in Figure 2.5, the system we have put forward relies on the two-dimension graph model. An undirected weighted graph G = (N, E) is built in which sentences are represented by a set of nodes N and the relation between each sentence is represented by the edge that connects the two correspondent vertices. Two types of edges are used: *statistical similarity* and *semantic similarity*:

Statistical similarity (Section 2.3.3.2): The edge between the pair of sentence is created if this measure exceeds a predefined threshold. The weight of the edge represents the number of common tokens between the two sentences divided by the length of each sentence.

Semantic similarity (Section 2.3.3.3): similar sentences have an edge between them. While the graph edges represent the semantic similarity between the sentences, the edge weight represents the degree of this similarity. The two-dimensional undirected weighted graph built in this step is input of the process in the next section to compute the score for each sentence.



Here, an Arabic document is represented as a two-dimensional graph:

- Each sentence is represented by a node
- Blue edge represents the statistical similarity between two sentences
- Green edge represents the semantic similarity between two sentences

Figure 2.5. A proposed graph representation of Arabic documents

2.3.3.6 Sentence ranker

The input of this process is the undirected weighted graph resulted from the previous step. PageRank algorithm (Brin and Page, 1998) was used to calculate a salient score for each vertex of the graph. PageRank is a very popular link analysis algorithm that was developed as a method for Web link analysis. It determines the importance of a vertex within a directed graph, on the basis of the information elicited from the graph structure. In our case, the key intuition is that a sentence should be highly ranked if it is recommended by many other highly ranked sentences.

PageRank can as well be used on undirected graph. In this respect, the output-degree and the input-degree for a node are equal. In our case, *In(Ni)* is equal to *Out(Ni)* since the graph is undirected. Equation (2.5) provides the score of a node *Ni*, where *adj (Ni)* is the set of vertices adjacent to *Ni*, w_{ij} is the weight of the edge between node *Ni* and node *Nj*, and *d* is a damping factor that can be set between *0* and *1*. The factor *d* has the role of incorporating into the model the probability of moving randomly from a given node to another in the graph. This factor is often set to *0.85* (Mihalcea and Tarau, 2004).

$$PR(N_i) = (1-d) + d * \sum_{N_j \in adj(N_i)} w_{ij} \frac{PR(N_j)}{\sum_{N_k \in adj(N_j)} w_{jk}} \quad (2.5)$$

We apply equation (2.5) iteratively on a weighed graph *G* to compute *PR*. First, all nodes are assigned an initial score of *1* and then equation (2.5) is applied to bring the scores difference between iterations below a threshold of *0.001* for all vertices. The salient scores of the sentences are represented by the weights of the vertices. When they correspond to vertices with higher scores, these sentences become important, salient to the document and have strong ties with others sentences.

It is to be noted that equation (2.5) is applied on both statistic and semantic edges. We obtain two scores for each node $PR_{static}(N_i)$ and $PR_{semantic}(N_i)$. The following formula gives the silent score of each node in the graph by summing statistical and semantic scores:

$$PR(N_i) = PR_{static}(N_i) + PR_{semantic}(N_i) \quad (2.6)$$

2.3.4 Post-treatment phase

Post-treatment is the final step of our system. It consists of eliminating redundancy from the best scored sentences by the formula (2.6). In this way, we are sure that our final generated summary covers a diversity of most information contained in the original input document.

In this step, and after carrying out the ranking process, each sentence has its salient score *Score(Si)*. Simply and as other graph based summarization systems, we can choose to include in the final summary (depending on the summary size) the sentences with the higher scores. However, this will create redundancy in the summary, since many similar sentences that represent the same meaning in the document have similar score, so they can be included together in the summary. Also, the remaining ideas of the document may not be identified and relevant information of the document may be overlooked and does not appear in the final summary. That is why the adapted version of MMR (Carbonell and Goldstein, 1998) is used to re-rank and select appropriate sentences to include into the summary without redundancy. MMR is an iterative method for content selection. In the case of automatic summarization task, it iteratively chooses the best sentence to insert in the summary according to two characteristics:

- Relevant: That is, the sentence must be highly relevant to the content of the text. So, the sentence with the higher ranking score will be considered.

- Novel: which means that the sentence must be minimally redundant with the summary, so the similarity between the sentence and other previously selected sentences in the summary needs to be low.

Algorithm 2.1. Ranking and generating summary via maximizing marginal relevance

```
Input: set of sentences R, score of each sentence, semantic similarity matrix, summary size n
Output: set of summary sentences S,
S←∅
for n=1, …, n do
$maxPos = argmax_{i:\, s_i \epsilon R \backslash S} \left[\lambda * score(s_i) - (1-\lambda) * max_{s_j \epsilon S} * sim(s_i, s_j)\right]$
    S←S U R(maxPos)
    R←R\ R(maxPos)
end for
return S
```

As shown in Algorithm 2.1, the sentence is incorporated if it is highly ranked and its similarity to any existing sentence in the summary must not be very high. First, the sentence with the highest rank is added to the summary *S* and removed from the ranked list *R*. The next sentence with the highest re-ranked score from Equation (2.7) is selected from the ranked list. It is then deleted from the ranked list and added to the summary. The same process is repeated until the summary attains the predefined length. The MMR method works according to the following equation:

$$MMR = argmax_{s_i \epsilon R \backslash S} \left[\lambda * score(s_i) - (1-\lambda) * max_{s_j \epsilon S} * sim(s_i, s_j)\right] \tag{2.7}$$

In this equation, *R* is the set of all sentences, *S* is the set of summary sentences, *score(s)* is the ranking score for sentences calculated in previous section and $sim(s_i, s_j)$ is the semantic similarity measure between sentences s_i and s_j; λ is a tuning factor between importance of a sentence and its significance to previously chosen sentences. We choose the value λ = 0.7 for the best performance in the experiments. According to the way we construct the graph, the sentences that are similar to one or more other sentences, tend to have higher scores and thus higher ranks. These kinds of sentences are often selected to form the summary. In contrast, the sentences, which are less similar to the others, and thus have less voting members, are hardly selected to the final summary. It was also revealed that the use of MMR is necessary to reduce the redundancy issue.

2.4 Experimental results

2.4.1 Experiment setup

Our system was compared with a set of the baseline approaches (i.e., only a statistical-based summarizer; graph-based summarizer without redundancy elimination; and a graph-based summarizer with redundancy elimination) to show the effectiveness of our method. The first system is a simple Arabic text summarizer based on TF.ISF feature. The summary is generated from the highest scored sentences. The score of each sentence is computed as follows:

$$score(S_i) = \frac{\sum_{wj \in S_i} TF.ISF(w_j)}{rootCount(S_i)} \qquad (2.8)$$

Where $TF.ISF(w_j)$ is the term frequency / inverse sentence frequency of the root w_j*; and* $rootCount(S_i)$ is the number of root in the sentence.

The second system is TextRank (Mihalcea and Tarau, 2004). TextRank is a graph-based ranking model used for both automatic text summarization and key-words extraction. It is based on PageRank (Brin and Page, 1998) algorithm in order to rank the graph elements that better describe the text. In the summarization task, each sentence is represented by a node in the graph and the edge between two nodes represents the similarity relation that is measured as a content overlap between the given sentences. The weight of each edge indicates the importance of a relationship. Sentences are ranked based on their scores and those that have very high score are chosen.

The third system is LexRank (Erkan and Radev, 2004). LexRank is another automatic summarization system which is identical to TextRank. Both of them use graph-based approach for text summarization and the only difference between them is that the similarity measure used by TextRank is based on the number of similar words shared between the two sentences, while LexRank uses cosine similarity measure of TF-IDF vectors.

For further comparison of our approach, we have implemented another graph-based summarizer with redundancy elimination proposed by Alami et al. (2015). This Arabic text summarizer based on graph theory (ATSG), uses a cosine similarity measure to calculate the similarity between sentences. It makes a graph representation for an input Arabic document and applies the PageRank algorithm in order to rank each sentence in the graph. The system is then performed by removing redundancy from the final summary.

We implemented all of these systems in java language. As mentioned above, we used two datasets to test and evaluate the performances of our system. The first dataset (Dataset-1) is the EASC corpus while the second dataset (Dataset-2) is our own built corpus. The two datasets are described in detail in section 1.9.1. Then we ran our algorithm to produce summaries for these sample texts in five various sizes: 20%, 25%, 30%, 35% and 40%.

2.4.2 Results and discussion

To assess the quality of the automatic generated summary of different systems, we have calculated Recall, Precision, F score and ROUGE- 1 score. Table 2.4 summarizes the results of running our algorithm on the ESCAS corpus and our own corpus with different sizes. As Table 2.4 illustrates, the recall decreases when the compression ratio goes down because the co-occurrence between candidate summary and gold summary increases.

The comparison between average Recall, precision and F-measure of our system with other baseline systems is given in Table 2.5. The summary size taken into account in this comparison is 30% of the original document. We can easily notice that our system has the highest average F score value when compared to other systems and for both of the used datasets. With summary size 30%, the best F-measure score of the other systems is reported by the ATSG system, with

46.76% for dataset-1 and 47.43% for dataset-2. However, in our experiment, the average value of F-measure reported by our system is 57.89% for dataset-1 and 63.41% for dataset-2. This amounts to saying that our algorithm enhances the performance of the graph-based summarization system.

Table 2.4 Evaluation Results of the proposed system on Dataset-1 and Dataset-2 using mean Recall, Precision and F-measure.

Dataset-1			Dataset-2			Summary size
Precision	Recall	F1-measure	Precision	Recall	F1-measure	
60.64	47.20	53.08	75.70	47.52	58.39	20%
58.23	53.33	55.67	71.68	54.48	61.91	25%
56.89	58.93	57.89	68.49	59.03	63.41	30%
53.13	65.30	58.59	62.65	61.32	61.98	35%
51.15	71.03	59.47	57.74	64.18	60.79	40%

Table 2.5 Comparison against other systems using average Recall, Precision and F-measure with 30% of summary size

	Dataset-1			Dataset-2		
System	Precision	Recall	F1-measure	Precision	Recall	F1-measure
Proposed method	56.89	58.93	57.89	68.49	59.03	63.41
ATSG (Alami et al., 2015)	46.22	47.31	46.76	51.58	43.90	47.43
TextRank	44.26	36.24	39.85	60.23	39.76	47.90
LexRank	31.03	25.71	28.12	42.22	27.95	33.63
TF.ISF	39.46	33.71	36.37	42.81	27.30	33.34

Table 2.6 Rouge-1 scores of the proposed system on Dataset-1 and Dataset-2

Rouge-1 for Dataset-1	Rouge-1 for Dataset-2	Summary size
0.3909	0.4458	20%
0.4446	0.5114	25%
0.4875	0.5765	30%
0.5366	0.6265	35%
0.5859	0.6685	40%

Table 2.7 Rouge-1 comparison against other systems with 30% of summary size

System	Rouge-1 for Dataset-1	Rouge-1 for Dataset-2
Proposed method	0.4875	0.5765
Proposed method without MMR	0.4011	0.4906
ATSG (Alami et al., 2015)	0.4753	0.5734
ATSG (Alami et al., 2015) without MMR	0.3678	0.4841
TextRank (Mihalcea and Tarau, 2004)	0.3892	0.4518
LexRank (Erkan and Radev, 2004)	0.3093	0.3841
TF.ISF	0.3759	0.4032

To confirm the previous results, we additionally applied Rouge metric. Table 2.6 shows the Rouge-1 score of our algorithm applied on both dataset-1 and dataset-2. Table 2.6 also shows that average Rouge-1 score of the proposed system increases when the compression ratio is increased.

Table 2.7 draws a comparison between Rouge- 1 score of our system with other systems. We notice that our system has the highest value of Rouge- 1 score, and outperforms all the other systems on both datasets. With summary size 30%, the best Rouge-1 result of the other state-of-the art methods is reported by the ATSG system, with 47.53% for dataset-1 and 57.34% for dataset-2. Whereas, in our experiment, Rouge-1 score is 48.75% when our system was run on dataset-1 and 57.65% when our system was run on dataset-2.

We observe from the obtained results, that our method outweighs all other methods thanks to the fact that our system can spot the relationships between similar words among all sentences using lexical databases, WordNet and Arabic WordNet. These relationships cannot be identified by the reference systems used in the experimentation. In addition, all of the reference systems do not have a redundancy-removal component, except the ATSG system that produces a reasonable result compared to other systems. This shows that removing redundancy is an important part of Arabic text summarization.

By this work, we can also confirm that various reasons account for the difficulty to compare the proposed approach to other existing systems. Firstly, unlike English, there is no approved benchmark reference for Arabic language against which to assess our approach in Arabic text summarization. Hence, the comparison of the performance of the proposed approaches is intricate given that a different dataset and different evaluation measures are used in each work. Dissimilarly, benchmarking in English can rely on DUC human generated summaries. Moreover, the community working on Arabic text summarization is still quite small. Add to this, lexical, syntactic, and semantic ambiguity are higher in Arabic because of the complexity of the language as far as spelling, vocabulary and morphology are concerned.

2.4.3 Effect of stemming on Arabic text summarization

2.4.3.1 Preliminaries

Automatic processing of Arabic language has been considered as a challenging task for automatic text summarization and information retrieval due to different reasons. First, Arabic is highly inflectional and derivational, and words can have many different forms which makes the task of morphology very complex. Second, written character in different ways depends on the position of letter in the word, which can add a complexity to Arabic words analysis. Third, Arabic words are often ambiguous due to the tri-literal root system. Based on such specifications in Arabic language, it is a hard matter to determine the root/stem of any Arabic word since it requires a detailed morphological, syntactic and semantic analysis of the text.

We should point out that in automatic NLP area and especially ATS field, a preprocessing step is indispensable to transform the unstructured data in textual documents into structured format in order to apply data mining techniques. This transformation aims to make a representation of an Arabic document and depending on the quality of this representation, the accuracy of any text mining tasks may be impacted positively or negatively. There are several methods used in text mining for preprocessing text documents, such as tokenization, stop-word removal, stemming, and term weighting.

This section presents the results of an experimental study of comparison of the three stemmers designed for Arabic language in automatic text summarization using graph-based approach and redundancy elimination as described Alami et al. (2015). Hence, the idea of comparing the three different Arabic stemmers is to shed light on its effect in increasing the summarization system effectiveness for Arabic documents.

One of the most challenging issues in Arabic language is the word stemming. A wide body of research has been carried out in this domain. The two most effective Arabic stemmers are Khoja (Khoja and Shereen, 1999) based on root-extraction stemmer and Larkey's light stemmer (Larkey et al., 2002 and Larkey et al., 2007).

Despite the stemming errors, it has been empirically demonstrated that stemming improves retrieval in many languages, including Arabic (Aljlayl and Frieder, 2002; Froud et al., 2012a; Al-Anzi and AbuZeina, 2015). Based on the investigation made by Atwan et al. (2014), it is shown that the light10 stemmer increase the information retrieval effectiveness for Arabic documents. The evaluation of the three different stemmers shown that light10 stemmer achieved the best result in term of mean average precision. In (Bsoul et al., 2014), the authors evaluated the impact of five similarity/distance measures on document clustering using two stemming algorithms, morphology- and syntax-based Arabic lemmatization algorithm; and morphology-based Information Science Research Institute (ISRI) stemming and compare the results to raw data clustering 'without stemming'. Based on the experimental results, it can be concluded that similarity/distance measures are more effective in the lemmatization stemming of morphological and syntactically structured words than ISRI and raw data that is expected where ISRI has over-stemming, and Raw Data "without stemming" has under-stemming. The results obtained by Froud et al. (2012a) showed that the Light Stemming outperformed the stemming approach because Stemming affects the words meanings. Froud et al. (2010) evaluated the impact of the stemming on the Arabic Text Document Clustering with five similarity/distance measures. The experiments showed that the use of the stemming will not yield good results, but makes the representation of the document smaller and the clustering faster. The work made by Osama et al. (2012) evaluated stemming techniques in clustering of Arabic language documents and determined the most efficient in preprocessing of Arabic language. The evaluation used three stemming techniques: root-based Stemming, light Stemming and without stemming. From experiments, results show that light stemming achieved best results in terms of recall, precision and F-measure when compared with other stemming techniques. The experiments depicted that Light Stemming is the best technique for feature selection in Arabic language document clustering, but root based stemming get deteriorated results for Arabic language document clustering; because Arabic language has a complex morphology, and it is a highly inflected language. The study carried out by Froud et al. (2012b), compares and analyzes the effectiveness of Latent semantic analysis model with a wide variety of distance functions and similarity measures to measure the similarity between Arabic words in two cases: with and without stemming, for two testing data. The obtained results show that the use of the Stemming gives more accuracy in some cases and the opposite in the others. A study of the effect of stemming on Arabic text categorization was performed in Al-Anzi and AbuZeina (2015). The literature shows that stemming is not the optimal choice for feature reduction. Moreover, the authors find that the stemming process is good in some cases and poor in others, as well as a result of using both the light stemming and the root-based stemming.

2.4.3.2 Experimental setup

In order to test the effect of the stemming on our Arabic summarization system, we selected three famous stemming algorithms for which we had ready access to the implementation: The Morphological Analyzer from (Khoja and Shereen, 1999), the Light Stemmer developed by Larkey et al. (2002) and Alkhalil morphological system developed by Boudlal et al. (2010).

Khoja’s root-extraction stemmer: A superior root-based Arabic stemmer is Khoja's stemmer. It removes suffixes, infixes, and prefixes and matches the remaining word with verbal and noun patterns, to extract the root. The stemmer uses several linguistic data files such as a list of all diacritic characters, punctuation characters, definite articles, and 168 stop words.

Larkey’s light stemmer: is a stem-based approach or Light Stemmer approach presented by Boudlal et al. (2011). It produces a stem instead of a root of a given Arabic word. The aim of the Stem-Based approach is to eliminate the most frequent prefixes and suffixes. Light stemming does not deal with patterns or infixes; it is simply a process of stripping off prefixes and/or suffixes. Although light stemmers produce fewer errors than aggressive root-based stemmers, root-based stemmers reduce the size of the corpus significantly.

Alkhalil Morpho Sys: is a morphological syntactic parser of Standard Arabic words presented by Boudlal et al. (2011). The system can process non vocalized texts as well as partially or totally vocalized ones. For a given word, it identifies all possible solutions with their morph syntactic features: vowelizations, proclitics and enclitics, nature of the word, voweled patterns, stems, roots and Syntactic form. The approach used is based on modeling a very large set of Arabic morphological rules, and also on integrating linguistic resources that are useful to the analysis, such as the root database, vocalized patterns associated with roots, and proclitic and enclitic tables. However, the number of lexical items and stems makes the lexicon voluminous and as a result the process of analyzing an Arabic text becomes long. Alkhalil analyzer give for each word several possible stems and roots. We use the Viterbi algorithm to select the most appropriate root of the given word.

2.4.3.3 Experimental Results

We ran our system with four configurations: using Khaja’s stemmer, Larkey’s stemmer, Alkhalil’s stemmer and without the use of stemming process. Table 2.8 shows the average of recall, precision and F1- measure obtained with each configuration. The size of the summary is 30%. The above results show that the result obtained with Khoja’s stemmer (with 0.51 of F1-measeur) outperformed those obtained using both Larkey’s stemmer and Alkhalil’s stemmer in term of precision, recall and F-measure. On the other hand, Larkey’s stemmer and Alkhalil’s stemmer get the same performances to our system with 0.48 of F1-measeur. This result was confirmed when the size of the summary is configured to 20%.

As we have mentioned previously, Arabic Text summarization is not studied enough in the literature. Moreover, we can state that it is very difficult to provide an improved evaluation of the proposed Arabic summarization system, due to the lack of gold standard corpora, for Arabic language, there is not any approved benchmark to evaluate our approach in Arabic text summarization. By contrast, in English there are DUC human generated summaries that can be used as a benchmark.

Table 2.8 Performance evaluation with the three stemmers and 30% summary size

	Precision	**Recall**	**F1-measure**	**Size**
Khoja's stemmer	0.55	0.48	0.51	30%
Larkey's stemmer	0.52	0.45	0.48	30%
Alkhalil's stemmer	0.52	0.44	0.48	30%

Table 2.9 Performance evaluation with the three stemmers and 20% summary size

	Precision	**Recall**	**F1-measure**	**Size**
Khoja's stemmer	0.57	0.35	0.44	20%
Larkey's stemmer	0.56	0.34	0.42	20%
Alkhalil's stemmer	0.51	0.31	0.39	20%

The forth run aims to evaluate the effect of the stemming process on Arabic text summarization effectiveness and how sensitive is Arabic summarization to the use of stemmer. Table 2.10 shows that the use of stemming performed well the Arabic text summarization.

We conclude that whether with the use of Khoja's stemmer (root-based), Larkey's stemmer (stem-based) or Alkhalil's stemmer, the system performances are improved. The worst result of our proposed system is obtained when our system was run without stemming process.

Table 2.10 Performance evaluation with and without stemming

Size	**Stemmer**	**Precision**	**Recall**	**F1-measure**
30%	Without stemming	0.49	0.42	0.45
	Khoja's stemmer	**0.55**	**0.48**	**0.51**
20%	Without stemming	0.52	0.32	0.39
	Khoja's stemmer	**0.57**	**0.35**	**0.44**

2.5 Conclusion

In this chapter, we have introduced a novel automatic summarization system for Arabic language with statistical and semantic treatment of the input document. The proposed system incorporates the advantages of a graph-based system and scoring sentences according to PageRank algorithm performed on the proposed two-dimensional graph that represents the Arabic document with both semantic and statistical relationships existing between the document sentences. The proposed system deals with a well-known problem in Arabic text summarization (redundancy and information diversity) by using an adapted version of MMR technique to remove redundancy from the final summary. The proposed system is knowledge-rich because it integrates an external knowledge database developed by human. In addition, the proposed system deals with a lack in the semantic resources dedicated to Arabic by using a machine translation between Arabic and English to benefit from the richness and opportunities offered by the English language in this field.

The comparison of performance measures clearly shows that the advantages of our system outweigh those of other summarization systems. Benchmarking the proposed algorithm using two different datasets showed that it outperforms all other systems. In addition, the system does not need any training data, and does not use any structural or domain-dependent features and was, therefore, successfully used to summarize Arabic texts. We have shown the results of the automatic evaluation of the system, and compared our summaries with human-made summaries using the ROUGE method and F-measure. We accordingly conclude that our approach outperforms other existing systems in terms of Rouge-1 and F1-measures.

In this chapter, we also investigated the effect of three stemmers (Light 10, Khoja and Alkhalil) on Arabic text summarization. Based on our experiments and results we conclude that the Khoja stemmer got best stemmer for Arabic text summarization using benchmark dataset, in general, our experiments shows superior significant improvement by Khoja compared to light 10 (Larkey stemmer), because it gets the highest performance result, but comparing the raw data without stemmer got the worst performance of the system, but without significant improvement between light10 and Alkhalil stemmers. The performances of our system are significantly improved by applying the stemming process in the pre-processing stage. We concluded that the summarization of an Arabic text is more effective when using the stemming process.

In the next chapter, we will turn to address the question of how to ameliorate the performance of Arabic text summarization by addressing the documents representation problem. Generally speaking, Traditional Arabic text summarization systems are based on bag-of-words representation, which involve a sparse and high-dimensional input data. We try to deal with this problem by adopting an unsupervised deep learning approach in order to learn the abstract representation of Arabic documents.

Chapter 3

Using Unsupervised Deep Learning for Automatic Summarization of Arabic Documents

3.1 Introduction

In the previous chapter, we have introduced a new Arabic text summarization method, which incorporates the semantic relationships between textual units provided by the lexical database AWN. Although this method produced positive results, AWN suffers from two main problems. First, it is incomplete and several terms and concepts do not exist. Second, the process of extracting information from this database is time-consuming and impact the performance of the proposed method. Accordingly, it is necessary to provide a more powerful solution that is able to detect the existing semantic relationships between different textual units (words, sentences … etc.).

In addition, several works on Arabic text summarization (Sobh et al., 2007; Boudabous et al., 2010; El- Fishawy et al., 2014; and Fattah et al., 2009) are based on traditional supervised machine learning algorithms. In order to build powerful systems with these kind of algorithms, we must deal with two main problems. First, they need hard feature extraction tools and domain knowledge in order to reduce the complexity of the data and facilitate the learning process. This kind of tools and knowledge present a major challenge in the context of Arabic. Second, they need a large annotated corpus in the learning stage to build more meaningful systems that deal with a specific task. In the context of Arabic text summarization, labeled data are very limited, which has a negative impact on the performance of the supervised approaches.

In this chapter, we adopt an unsupervised deep learning model for the following reasons:

- To express the implicit semantic relations instead of using the explicit semantic relations provided by AWN.
- To learn automatically high-level features from data by unsupervised feature learning instead of using feature extractors or domain expertise.
- To deal with a shortage in labeled data, while unlabeled data are widely available for learning meaningful representations.

We present a new method for Arabic text summarization based on the variational auto-encoder (VAE) model (Kingma and Welling, 2014) to learn a feature space from a high-dimensional input data. Instead of using BOWs representation, we use a probabilistic generative model that projects the input sentences of a specific document into a latent semantic space. We explore several input representations such as term frequency (tf), tf-idf and both local and global vocabularies. We use the VAE as the basic of the generative model. VAE is an unsupervised deep learning technique that learns features from the input text in order to build a distributed latent semantic vector for each sentence. Our summarization system uses this new representation in order to rank each sentence and extract the most salient among them.

The motivation behind using VAE can be summarized as follows: First, VAE as an unsupervised approach for learning a lower-dimensional feature representation, deals with a shortage in labeled data, while unlabeled data are widely available for learning meaningful representations. Second, new generative models such as VAE and generative adversarial networks have been proposed to handle the inference problem in traditional generative models such as Gaussian mixture model, Hidden Markov model and Naive Bayes. In this work, VAE is used as a generative framework. Third, VAE has shown better result than other neural networks in learning a lower-dimensional feature representation of input data (Li and Misra, 2017).

Unsupervised feature learning methods have shown successful results in visual features extraction from input images (Bengio, 2009; Akbarizadeh, 2013; Rahmani and Akbarizadeh, 2015; Akbarizadeh and Moghaddam, 2016). In order to create more powerful learning models, unsupervised feature learning methods are combined with deep learning and have been shown outstanding results in feature extraction of visual data (Krizhevsky et al., 2012; Sermanet et al., 2014; Donahue et al., 2017). Recently, they have been successfully applied to natural language processing tasks (NLP) such as sentence modeling (Er et al., 2016), named-entity recognition (Li et al., 2017), text categorization (Ayinde and Zurada, 2017), machine translation (Firat et al., 2017) and ATS (Yousefi-Azar and Hamey, 2017; Zhong et al., 2015). The focus of this chapter is to apply this technique on Arabic text summarization.

The main contributions of this chapter are as follows: First, we present a new unsupervised approach for Arabic text summarization using deep learning based on VAE. To the best of our knowledge, there is no work exploring the use of deep learning and VAE for summarizing Arabic documents. This work is the first attempt that utilizes unsupervised feature learning with VAE in Arabic text summarization. Second, we assess how VAE handles a sparse word representation such as *tf-idf* and how the model works when using word representation based on local term frequency (*Ltf*) of a document-specific vocabulary. Third, we investigate the use of two summarization techniques, graph-based and query-based techniques. We show that the VAE outperforms both summarization techniques. Finally, in the evaluation stage, experimental results on two benchmark datasets specifically designed for Arabic text summarization prove that our framework achieves better performance than the state-of-the-art models. We show that the VAE using *tf-idf* representation of global vocabularies clearly provides a more discriminative feature space and improves the recall of other models. Experiment results confirm that the proposed method leads to better performance than most of the state-of-the-art extractive summarization approaches for both graph-based and query-based summarization approaches.

The next section presents a critical review of the literature in unsupervised feature learning. The method description with it architecture and training algorithms is presented in section 3.3. The experimental setup, results and discussion are described in section 3.4. The conclusion is explained in section 3.5.

3.2 Unsupervised Feature Learning

Unsupervised feature learning has shown promising results in image analysis. Akbarizadeh (2013) proposed a new method to segment sensor of synthetic aperture radar (SAR) satellite images. The proposed method used cellular learning automata for image feature extraction. The experimental results shown that the proposed approach has produced better results than other models such as Markov random field model and region-based hierarchical model. A new spectral clustering method for the segmentation of SAR image is proposed by Rahmani and Akbarizadeh (2015). The authors explored a sparse coding algorithm in order to learn the extracted features with an unsupervised manner. The segmentation of the SAR image is achieved by the spectral clustering method performed on learned features. In another work of Akbarizadeh et al. (2014), curvelet method is used to extract features of various textures existing in SAR images. The process of image segmentation and recognition is completed by applying the watershed transform to the matrix formed by the extracted features. Akbarizadej and Moghaddam (2016) presented a new method for the detection of lung nodules in CT scans by using an unsupervised feature learning with a fuzzy inference system in order to decrease the system error.

In order to create more powerful learning models, unsupervised feature learning methods are combined with deep learning and have been shown outstanding results in computer vision. Ahmadi and Akbarizadeh (2017) presented a new hybrid method for iris recognition. After performing the preprocessing step, the input data of the network are composed of a two dimensional Gabor kernel features extracted from the iris dataset. In order to increase the generalization performance, the proposed algorithm use a multilayer perception neural network trained by a practice swarm optimization algorithm for data classification. The experimental results show better results than many other well-known techniques. Wang et al. (2016) addressed the classification problem of hyperspectral data by introducing a hybrid approach that combines principle component analysis (PCA), guided filtering and uses a stacked denoising auto-encoders (Vincent et al., 2008) as a deep learning model to achieve the multi-feature learning task. In the work published by Noda et al. (2014), the authors proposed a new algorithm based on two deep learning architectures as unsupervised feature learning models in order to perform the speech recognition task. The first model is a deep denoising auto-encoder utilized to acquire noise-robust audio features. The second model is a convolutional neural network (CNN) (Lecun et al., 1998) which is utilized to extract visual features from raw mouth area images in order to predict phoneme labels.

Deep learning has also been applied for feature learning in medical imaging. In Kim et al. (2016), CNN was used as a feature extractor in the application of cytopathology image classification. The proposed model has been trained on millions of generic images and achieved remarkable results. Esteva et al. (2017), trained a CNN using a dataset of 129,450 clinical images and achieved comparable performance to dermatologists at skin cancer classification task. Gulshan et al. (2016) proposed an algorithm based on a deep convolutional network which is trained on a large set of retinal images in order to detect diabetic retinopathy in retinal fundus photographs.

Recently, variational auto-encoders (VAEs), simultaneously discovered by Rezende et al. (2014) and Kingma and Welling (2014), have been proposed as a powerful method for unsupervised learning. VAEs combine variational inference methods with deep neural networks techniques in order to transform the hard inference problems into optimization problems. This approach is similar to deep auto-encoder. The main advantage of VAEs is that they provide more effective inference over continuous distributions in the concept space by using a regularization method based on KL-divergence. Thus, the optimization of the variational inference objective by using stochastic gradient descent and standard backpropagation becomes easy. This technique is known as reparametrization trick.

3.3 Method

The proposed method is divided into three major stages: i) preprocessing stage; ii) training stage; iii) and summarization stage. In the preprocessing stage: First, we generate a vocabulary with a size *V* based on the most frequency words existing in the dataset. Second, we build the matrix representation for each document in the corpus. Consider a document *d* consisting of *n* sentences, $d = \{x_1, \ldots, x_n\}$ is the feature vectors of the document represented by BOWs approach; and x_i is the vector representation of sentence *i* in document *d*. The training stage consists of feature learning of input sentences. We build our model based on the unsupervised VAE model which is designed to project sentences from the term vector space to the latent semantic space. The produced matrix in the preprocessing stage is fed to the deep architecture as a visible layer. We train the network using stochastic gradient method until the reconstruction error reaches the minimum value.

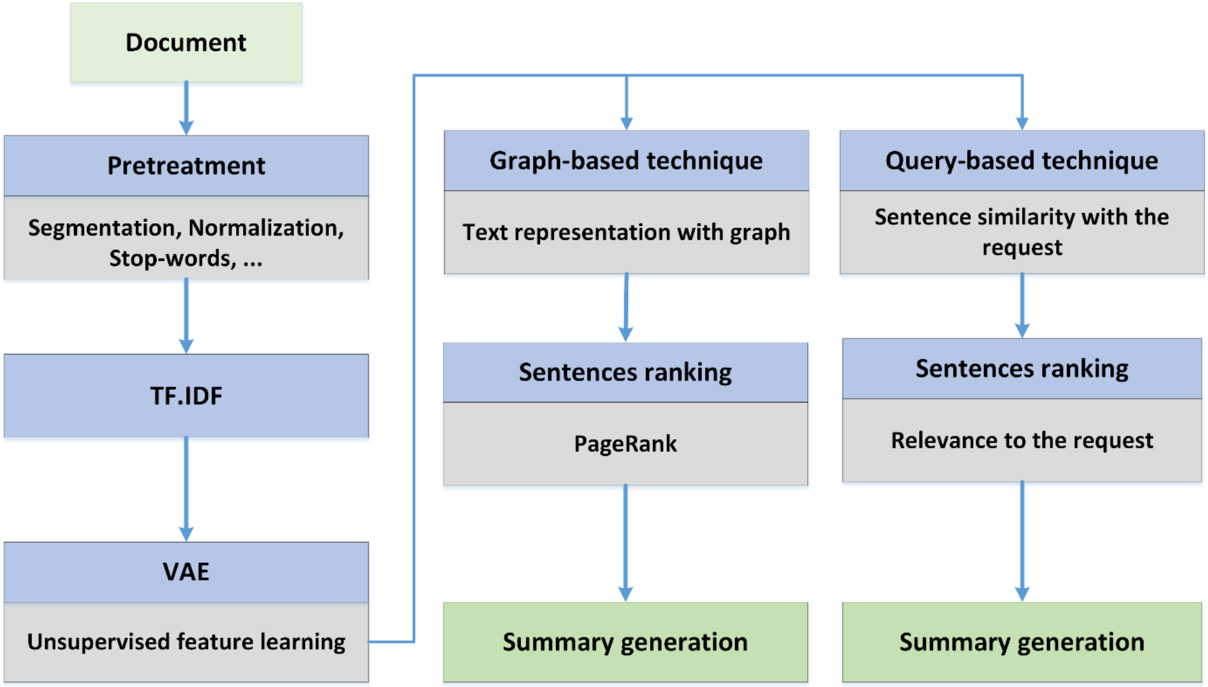


Figure 3.1 Overview of the proposed method.

The dimensionality of the input matrix is reduced to the number of hidden layers in the latent space. Hence, the input matrix is transformed into a concept space representing the latent values of the input. Finally, the summary is generated based on the cosine similarity measure in the semantic latent space. Two summarization approaches are utilized: graph-based and query-based summarization. Figure **3.1** shows an overview of the proposed method.

3.3.1 Variational Auto-Encoder

Recently, the use of unsupervised learning techniques has become very promising in many applications due to the increasing availability of unlabeled data. Many applications in NLP performed handsomely when using AE with pre-training phase, to exemplify, image analysis (Wang et al., 2016; Noda et al., 2014), document retrieval (Hinton et al., 2006; Hinton and Salakhutdinov, 2006) and automatic text summarization (Yousefi-Azar and Hamey, 2017; Zhong et al., 2015). In recent years, important new finding has been reported in training AEs and their association with generative models (Kingma and Welling, 2014; Kingma et al., 2014). These breakthroughs made the AEs a powerful alternative for training unsupervised and semi-supervised tasks. The variational auto-encoder (VAE) is one of these recent breakthroughs and it constitutes, in this work, a main paradigm for the proposed method.

VAE was introduced by Kingma and Welling (2014) which combine neural networks techniques with variational Bayesian methods. The idea behind it is to benefit from the opportunity provided by variational inference so as to convert a hard problem of inference into an optimization problem. In Kingma and Welling (2014), the use of neural networks in VAE allows to approximate the intractable conditional posterior. Furthermore, it is possible to optimize the variational inference by using back-propagation techniques such as stochastic gradient descent. For more details on VAEs, see Kingma and Welling (2014).

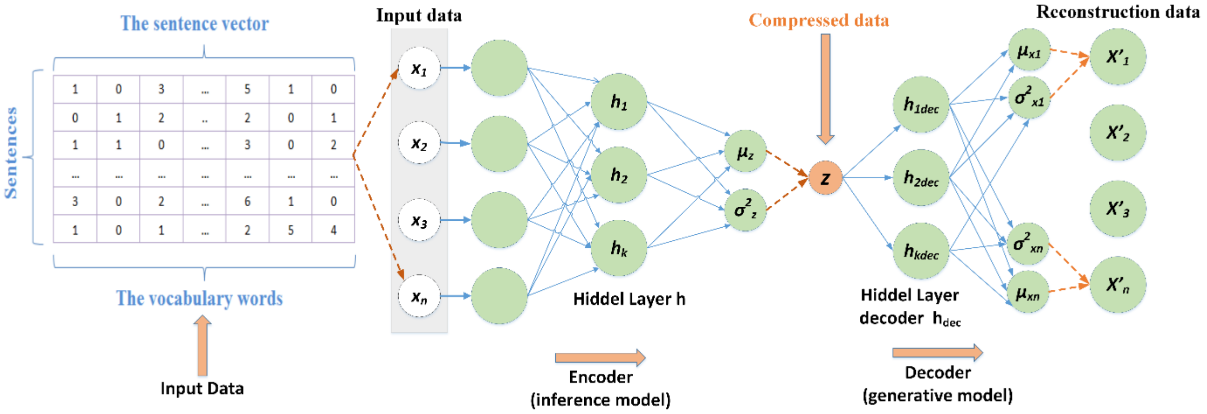
3.3.2 Training the VAE

Assuming we have a dataset composed of a set of documents *D* and a generated vocabulary of length *V* corresponding to the most frequency words appearing in the dataset. For simplicity, we consider a neural network with one hidden layer h, a latent space *z* and an input feature vector *x*.

VAE (Figure 3.2) can be viewed as an unsupervised learning technique with two models: variational encoder (inference model) and variational decoder (generative model). Each sentence, in a document to be summarized, is modeled by a vector $x \in \mathbb{R}^V$ which represents a feature vector of this sentence according to the vocabulary *V*. x_i (element of vector *x*) can be the *tf* or *tf.idf* of the word *i* in the given sentence. The inference model (encoder) attempts to encode the input vector *x* into a concept space represented by a latent semantic vector $z \in \mathbb{R}^L$ (L is the size of the latent space). Then, the features, which are the abstract representations of the input, from concept space *z* are used by generative model (decoder) to generate an approximate reconstruction $\hat{x}$ of the original input sentence features vector. The goal is to maximize the likelihood of each vector *x* in the document set *D* using the generation process according to:

$$p_\theta(x) = \int p_\theta(x,z)\,dz \qquad (3.1)$$

where θ is the generative network parameters, x is the input observation and z is the latent variable in the deep neural network.



1	0	3	...	5	1	0
0	1	2	..	2	0	1
1	1	0	...	3	0	2
...	...	...	...	...	...	...
3	0	2	...	6	1	0
1	0	1	...	2	5	4

Figure 3.2 Our VAE topology for dimensionality reduction. In the left, the generated matrix representation of a document is fed to the VAE as a visible layer and projected into a concept space z.

VAE introduced a recognition model $q_{\Phi}(z|x)$ with parameters Φ in order to approximate the intractable true posterior $p_{\theta}(z|x)$. Now, we are facing an optimization problem which consists of fitting the approximate posterior $q_{\Phi}(z|x)$ to the true posterior $p_{\theta}(z|x)$ by reducing the Kullback-Leibler divergence between them. The marginalized likelihood becomes:

$$D_{KL}[q_{\Phi}(z|x)||p_{\theta}(z|x)] = \int_z q_{\Phi}(z|x)\log\frac{q_{\Phi}(Z|X)}{p_{\theta}(Z|X)} = E_{q_{\Phi}(z|x)}[logq_{\Phi}(z|x) - logp_{\theta}(z|x)] \tag{3.2}$$

After applying Byes rule on $p_{\theta}(z|x)$, we obtain the following:

$$D_{KL}[q_{\Phi}(z|x)||p_{\theta}(z|x)] = \log p_{\theta}(x) + E_{q_{\Phi}(z|x)}[\log q_{\Phi}(z|x) - \log p_{\theta}(x|z) - log\ p_{\theta}(z)] \tag{3.3}$$

The marginal likelihood for the input vector x can be written as:

$$\log p_{\theta}(x) = D_{KL}[q_{\Phi}(z|x)||p_{\theta}(z|x)] + E_{q_{\Phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}[q_{\Phi}(z|x)||p_{\theta}(z)] \tag{3.4}$$

Let $\mathcal{L}(\theta, \phi;\ x) = \mathrm{E}_{q_{\Phi}(z|x)}[\log p_{\theta}(x|z)] - D_{KL}[q_{\Phi}(z|x)||p_{\theta}(z)]$ (3.5)

We have $\log p_{\theta}(x) \geq \mathcal{L}(\theta, \phi;\ x)$ $since$ D_{KL}, the KL-divergence between the approximation and true posterior distribution, is positive, so $\mathcal{L}(\theta, \phi;\ x)$ represents the variational lower bound to the marginalized likelihood. The main idea of training the VAE is to find the good optimal parameters θ and ϕ of the network in order to maximize the lower bound $\mathcal{L}(\theta, \phi;\ x)$. The neural network uses stochastic back-propagation of the gradient in order to update the parameters θ and ϕ.

The proposed method follows the idea published in (Kingma and Welling; 2014). We expect that both the posterior (probabilistic encode) and prior (probabilistic decode) of the variables in the latent space are Gaussian, which mean that $q_{\Phi}(z|x) = \mathcal{N}(z, \mu, \sigma^2 I)$ and $p_{\theta}(z) = \mathcal{N}(0, I)$. Where σ and μ, called the reparameterization trick, represent the standard deviation and variational mean, respectively. The reparameterization trick σ and μ as well as the latent semantic vector *z* can be computed with a multilayer perception by the following steps:

1. Given the input sentence, we build the term vector representation x according to the vocabulary *V*.
2. Project the vector *x* to an encoder hidden layer *h* using an activation function. In this work, we use the sigmoid function as follows:

   $$h = sigm(\mathrm{W_{xh}x} + \mathrm{b_{xh}}) \quad (3.6)$$

   where $sigm(x) = {}^{1}\!/_{(1+e^{-x})}$, *Wxh* and *bxh* are the network parameters. Other activation function can be used in this step such as $relu(x) = \max(0, x)$.
3. Then the Gaussian reparameterization trick, μ and σ, can be calculated using a linear transformation of the hidden layer h:

   $$\mu = \mathrm{W_{h\mu}h} + \mathrm{b_{h\mu}} \quad (3.7)$$

   $$\log(\sigma^2) = \mathrm{W}_{\mathrm{h}\sigma}\mathrm{h} + \mathrm{b}_{\mathrm{h}\sigma} \quad (3.8)$$
4. The semantic vector *z* of the latent space is computed with the following formula:

   $$z = \mu_{\Phi}(x) + \sigma_{\Phi}(x) \odot \epsilon \quad (3.9)$$

   where $\epsilon \sim \mathcal{N}(0, \mathrm{I})$ represents an auxiliary noise variable.

It is clear that the projection from *x* to *z* (encoding process) is similar to the encoding process used in the classical AE. Moreover, Equation (3.5) can be divided into two parts: the stochastic version of the negative reconstruction error in the general AE presented by the first term and an additional regularization term which allows the approximate posterior and the prior to be close to each other.

Given the latent semantic vector z, a new term vector $\hat{x}$ is generated by reconstructing the input x via the conditional distribution $p_\theta(x|z)$. Under the neural network framework, the reconstruction process (generative model) is similar with the decoding process of the typical AE model:

$$h_{decoder} = relu(\mathrm{W_{zh}z} + \mathrm{b_{zh}}) \quad (3.10)$$

$$\hat{x} = sigmoid(W_{hx}h_{decoder} + b_{hx}) \quad (3.11)$$

Finally, based on the reparameterization trick in Equation (3.9), we get the analytical representation of the variational lower bound $\mathcal{L}(\theta, \phi;\ \mathrm{x})$:

$$log\, p(x|z) = \sum_{i=1}^{|V|} x_i \, log\, \hat{x} + (1 - x_i).log(1 - \hat{x}_i) \quad (3.12)$$

$$-D_{KL}[q_{\Phi}(z|x)||p_\theta(z)] = \frac{1}{2}\sum_{i=1}^{K}(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2) \quad (3.13)$$

All the parameters {W, b} can be learned using back-propagation method. The optimum parameters of our model are given in Table 3.1.

Table 3.1 Optimum parameters of our model VAE.

Parameter	**Value**
Batch size	100
Number of iteration	25
Optimizer	Adam
Learning rate	0.01
Number of layers	2
Size of the first hidden layer	250
Size of the second hidden layer	20

3.3.3 Summary Generation

After training the whole network, the VAE becomes globally adjusted with the best optimal parameters θ and ϕ. Each sentence in the document to be summarized is presented by a feature vector according to the vocabulary V. The encoder maps each sentence vector into an abstract representation in the concept space. More precisely, assuming we have a document d with a set of sentences $S = \{S_1, \dots, S_n\}$, and a mapping function $M(S)$ given by the encoder of our VAE. Each sentence S_i in S is mapped into the low-dimensional latent space in order to produce it latent representation $\widehat{S_\iota}$. The similarity score between each pair of sentence is computed by the cosine similarity as following:

$$sim(S_i, S_j) = \frac{\widehat{S_\iota}.\widehat{S_J}}{\|\widehat{S_\iota}\|\|\widehat{S_J}\|} \tag{3.14}$$

where $\widehat{S_\iota} = M(S_i)$

In this part of our thesis work, we investigate two summarization techniques on Arabic datasets: graph-based and query-based text summarization techniques.

3.4 Experimental Setup

It is worth mentioning that evaluating Arabic text summarization is a thorny issue due to the lack of standard benchmark dataset, unlike English where the task can rely on the human-generated summaries available in Document Understanding Conferences (DUC) or Text Analysis Conference workshops. In order to properly evaluate the effectiveness of our approach, we conduct several experiments on two different available Arabic corpora specifically developed for the task of Arabic summarization (see section 1.9.1). In this section, we describe the baseline methods we compare with, and implementation details of our approach.

3.4.1 Results and discussion

We evaluated the performance of our system by using ROUGE metric (Recall-Oriented Understudy for Gisting Evaluation) (Lin, 2004). In this work, we have explored several input representations with different vocabulary size*: tf-idf* with global vocabulary (*Gtf-idf*), *tf* with local vocabulary (*Ltf*), the latent representation of *Gtf-idf* and the latent representation of *Ltf* produced by our VAE. We add a noise to the input representation in order to avoid the sparsity because with a large vocabulary size, we have a lot of zeroes in the input, which affect negatively the performance of our VAE.

We investigated two different experimentations on the EASC corpus. The first experiment used graph-based summarization technique which use graph model in order to summarize Arabic documents. The second experiment used query-oriented summarization technique. In this technique, words appearing in the title of each document are considered as the query terms. **Figure 1.3** is an illustration sample of Arabic text used in the evaluation process.

3.4.1.1 Evaluation Results on EASC Dataset

Table 3.2 presents the ROUGE-1 recall of the proposed model using graph-based summarization technique and other representation models with different summary size. It is clear from Table 3.2 that recall increases when summary size increases and vocabulary size for *Gtf.idf* decreases. The recall of our proposed system using global *Gtf.idf* representation as the input of the VAE is improved with higher summary size. However, decreasing the length of the vocabulary reduces the recall, and the performances of the VAE become impaired. The VAE needs a large vocabulary to improve the quality of graph-based summarization system of Arabic documents.

Comparing the results in Table 3.2 clearly shows the following important points. First, for all summary size, the VAE (*Gtf-idf and V=1000*) is much better than the baselines representations using real *Gtf.idf*, and this shows that the VAE (*Gtf-idf*) provides a better projection of the input into the latent space. Second, the performances of our VAE decrease when the vocabulary size decreases, which demonstrates that the VAE works better when using a large vocabulary. Third, when using the *tf* representation of local vocabulary, the performances of our model (*VAE Ltf*) are decreased compared to the baseline representation (*Ltf*); this is because our model needs larger vocabulary size to give better results. Finally, in all case, the VAE (*Gtf-idf and V=1000*) outperforms other representations (baselines *Gtf.idf*, *Ltf*, VAE (*Ltf*) and VAE with small vocabulary) when the vocabulary is much large.

Table 3.2 ROUGE-1 of graph-based summary on EASC with different vocabulary and summary size

Model	10%	15%	20%	25%	30%	35%	40%	45%
Gtf.idf V 1000	0.098	0.166	0.253	0.325	0.369	0.438	0.488	0.517
Gtf.idf V 500	0.100	0.174	0.254	0.321	0.370	0.434	0.481	0.515
Gtf.idf V 300	0.101	0.181	0.263	0.318	0.365	0.433	0.484	0.521
Gtf.idf V 50	0.100	0.169	0.251	0.314	0.359	0.425	0.471	0.506
VAE Gtf.idf V 1000	**0.110**	**0.188**	**0.282**	**0.345**	**0.402**	**0.472**	**0.529**	**0.561**
VAE Gtf.idf V 500	0.107	0.182	0.279	0.345	0.397	0.465	0.518	0.549
VAE Gtf.idf V 300	0.101	0.178	0.270	0.338	0.386	0.450	0.499	0.534
VAE Gtf.idf V 50	0.096	0.158	0.235	0.292	0.335	0.399	0.456	0.496
Ltf V 50	0.099	0.169	0.257	0.320	0.372	0.445	0.503	0.533
VAE Ltf V 50	0.076	0.132	0.207	0.262	0.302	0.359	0.406	0.441

For further evaluation of our model, we use a query-based technique as a summarization approach. We consider the title of the document as a query if one exists; otherwise, we can use the first sentence as a query since our corpus is extracted from Wikipedia and newspapers websites. Table 3.3 shows the ROUGE-1 recall of the baselines and VAE-based models with various input representations. We can observe that recall is increased compared to graph-based summarization approach (Table 3.2) since the summarization in this case extracts relevant sentences close to the document topic (title or first sentence). All models are improved by this summarization technique, and the amount of enhancement varies for each model.

Table 3.3 ROUGE-1 of query-based summary on EASC with different vocabulary and summary size

Model	10%	15%	20%	25%	30%	35%	40%	45%
Gtf.idf V 1000	0.112	0.182	0.274	0.329	0.379	0.447	0.492	0.528
Gtf.idf V 500	0.112	0.181	0.274	0.331	0.377	0.441	0.488	0.530
Gtf.idf V 300	0.111	0.178	0.267	0.332	0.382	0.448	0.502	0.537
Gtf.idf V 50	0.109	0.179	0.269	0.328	0.378	0.449	0.501	0.537
VAE Gtf.idf V 1000	**0.115**	**0.194**	**0.286**	**0.350**	**0.403**	**0.466**	**0.526**	0.561
VAE Gtf.idf V 500	0.112	0.180	0.272	0.335	0.381	0.446	0.506	0.547
VAE Gtf.idf V 300	0.108	0.179	0.271	0.344	0.389	0 459	0.508	0.544
VAE Gtf.idf V 50	0.108	0.177	0.268	0.332	0.377	0.440	0.489	0.528
Ltf V 50	0.109	0.177	0.267	0.331	0.377	0.447	0.501	0.540
VAE Ltf V 50	0.111	0.175	0.259	0.318	0.364	0.425	0.476	0.511

Table 3.4 ROUGE-1 of graph-based summary on our own corpus with different vocabulary and summary size

Model	10%	15%	20%	25%	30%	35%	40%	45%
Gtf.idf V 1000	0.138	0.203	0.291	0.361	0.406	0.465	0.521	0.569
Gtf.idf V 500	0.145	0.220	0.298	0.374	0.442	0.491	0.532	0.587
Gtf.idf V 300	0.116	0.189	0.285	0.367	0.423	0.479	0.536	0.598
Gtf.idf V 50	0.117	0.179	0.263	0.331	0.406	0.465	0.527	0.563
VAE Gtf.idf V 1000	0.129	0.1970	0.293	0.367	0.431	0.490	**0.573**	**0.617**
VAE Gtf.idf V 500	**0.158**	**0.232**	**0.335**	**0.405**	**0.459**	0.498	0.559	0.608
VAE Gtf.idf V 300	0.132	0.197	0.274	0.352	0.413	0.486	0.538	0.582
VAE Gtf.idf V 50	0.108	0.180	0.273	0.344	0.411	0.476	0.538	0.579
Ltf V 50	0.135	0.201	0.299	0.388	0.445	**0.505**	0.554	0.598
VAE Ltf V 50	0.092	0.147	0.227	0.313	0.357	0.421	0.477	0.521

VAE (*Gtf.idf* and *V = 1000*) gives better results than basic *Gtf.idf* representation with the same vocabulary size. When the summary size has 30%, the recall results of *Gtf.idf* with 1000 and 500 vocabulary sizes are 0.379 and 0.377 respectively, which are improved by VAE (*Gtf.idf*) to 0.403 and 0.381. We can note that the enriched representation of the input term space enables the unsupervised VAE to better learn a richer latent space with reduced dimensions, achieving better summaries.

3.4.1.2 Evaluation Results on Our Own Dataset

Using graph-based and query-based techniques on our own dataset, the evaluation results (Table 3.4) show that, in most cases, the summarization performance is improved by VAE (*Gtf.idf and V = 500*), except when the summary size is 35%, the best recall is achieved by *Ltf* with 50 vocabularies. However, when the summary size is large (40 and 45%), the best performance is obtained by VAE (*Gtf.idf and V= 1000*). Table 3.5 shows that the VAE outperforms query-based summarization in all cases. The best result is obtained by VAE with 500 vocabularies.

3.4.1.3 Performances Comparison with Other Techniques

For further evaluation of the effectiveness of our model, we compare the performance of the proposed method with other representative ones on the two available datasets. To achieve this, we developed five summarization systems and adapt them to Arabic documents. The first baseline system is based on *TF.ISF* feature in order to compute the relevance of each sentence. In this system, the score of each sentence is computed as follows:

$$score(S_i) = \frac{\sum_{wj \in S_i} TF.ISF(w_j)}{rootCount(S_i)} \tag{3.16}$$

Where $TF.ISF(w_j)$ is the term frequency/inverse sentence frequency of the root w_j*; and* $rootCount(S_i)$ is the number of root in the sentence.

In the second system, we used the unsupervised AE model to train our datasets and build the semantic latent space. The AE architecture has a hidden layer with 20 units used to represent the input in the semantic concept space. The third system is a topic-based summarization system which is based on Latent semantic analysis (LSA) (Mashechkin et al., 2011). LSA is used to project the matrix obtained by the BOWs representation into the topic space, so the dimensionality of the matrix is reduced. The new matrix representing the topic space is utilized by a graph-based method to build a summary. The fourth and fifth systems are TextRank (Mihalcea and Tarau, 2004) and LexRank (Erkan and Radev, 2004) respectively. TextRank and LexRank are described in detail in section 1.5.6.

The evaluation results listed in Table 3.6 show that our algorithm outperforms the existing methods when the evaluation task is carried out on the EASC corpus. This result is valid for such graph-based and query-based algorithms. In Table 3.7, we provide the performance comparison on our own dataset. As we can see, the performance of our model is higher than other systems for both summarization techniques (graph and query). Therefore, we can state that our proposed algorithm can improve the summarization task giving better results in various cases.

As mentioned above, with the proposed method, we do not need to have labeled data, which consist in this case of a set of documents with human-generated summaries. These labeled data are very difficult to obtain, especially for Arabic, due to the luck in annotated summarization corpus designed for Arabic on the one hand and to the difficulty of manually creating summaries on the other hand. By contrast, with the emergence of the Internet and the digital world, unlabeled data become more widely available compared to labeled data. Thus, the availability of vast amounts of unlabeled data has made it imperative to adopt unsupervised learning framework in order to construct an automatic summarization model designed for Arabic documents. It is one of the strengths of the proposed method.

Table 3.5 ROUGE-1 of query-based summary on our own corpus with different vocabulary size

Model	10%	15%	20%	25%	30%	35%	40%	45%
Gtf.idf V 1000	0.149	0.218	0.307	0.376	0.419	0.461	0.520	0.560
Gtf.idf V 500	0.151	0.223	0.315	0.385	0.415	0.462	0.532	0.570
Gtf.idf V 300	0.150	0.233	0.320	0.392	0.433	0.470	0.523	0.567
Gtf.idf V 50	0.149	0.231	0.310	0.389	0.466	0.499	0.582	0.611
VAE Gtf.idf V 1000	**0.157**	0.238	0.338	0.411	0.479	0.526	0.577	0.608
VAE Gtf.idf V 500	0.150	**0.241**	**0.364**	**0.449**	**0.483**	**0.541**	**0.608**	**0.660**
VAE Gtf.idf V 300	0.154	0.235	0.312	0. 393	0.466	0.5311	0.601	0.654
VAE Gtf.idf V 50	0.156	0.243	0.344	0.403	0.449	0.4914	0.539	0.586
Ltf V 50	0.152	0.234	0.319	0.379	0.428	0.484	0.544	0.590
VAE Ltf V 50	0.155	**0.246**	0.328	0.385	0.435	0.473	0.521	0.583

Table 3.6 ROUGE-1 comparison with other methods on EASC corpus with different summary size

System	10%	20%	30%	40%
Graph-based VAE Gtf.idf V 1000	0.110	0.282	0.402	**0.529**
Query-based VAE Gtf.idf V 1000	**0.115**	**0.286**	**0.403**	0.526
Graph-based AE	0.079	0.212	0.321	0.430
LSA (Topic-based)	0.104	0.255	0.360	0.431
TextRank	0.112	0.278	0.382	0.501
LexRank	0.082	0.211	0.309	0.411
Baseline Tf.ISF	0.106	0.269	0.379	0.503

Table 3.7 ROUGE-1 comparison with other methods on our corpus with different summary size

System	10%	20%	30%	40%
Graph-based VAE Gtf.idf V = 500	**0.158**	0.335	0.459	0.559
Query-based VAE Gtf.idf V 500	0.150	**0.364**	**0.483**	**0.608**
Graph-based AE	0.096	0.233	0.370	0.483
LSA (Topic-based)	0.134	0.293	0.416	0.540
TextRank	0.156	0.331	0.452	0.566
LexRank	0.111	0.259	0.378	0.484
Baseline Tf.ISF	0.116	0.261	0.389	0.518

Table 3.8 displays an example of our system summary (VAE) compared with summaries extracted with the competitors. The input text is given in **Figure 1.3**. It is clear from this comparison that the summary generated by VAE is closer to human-generated summary than other models. Our method has 3 sentences existing in the human-generated summary, while the best result for the competitors is obtained by AE, LSA and TextRank which have 2 sentences existing in the human-generated summary. The bad result is obtained by LexRank and *Tf.ISF*.

Table 3.8 Comparison between reference summary, proposed method and competitors

<table>
<tr><th colspan="3">Human generated summary (Reference summary)</th></tr>
<tr><td colspan="3">- كشفت دراسة حديثة أجريت في تايوان أن أنواعا معينة من جراحات إنقاص الوزن قد تضعف العظام وتسبب هشاشتها وتزيد مخاطر الإصابة بالكسور.
- وتساعد جراحات إنقاص الوزن، مثل تحويل مسار المعدة، مرضى السمنة في فقدان الوزن من خلال تقليل كمية الطعام التي يمتصها الجسم.
- وقال كبير الباحثين في الدراسة الدكتور كو تشين هوانغ -وهو من كلية الطب في جامعة تايوان الوطنية في تايبيه - إن الجسم يُحرم من الكثير من العناصر الغذائية عندما يخضع المرضى لمثل هذه الجراحات.
- وذكروا أن بحثا سابقا أشار بالفعل إلى أن هذه الجراحات قد تزيد مخاطر كسور العظام.
- وأضاف "ثانيا يمكن أن يساعد التعرض للشمس وممارسة التمارين الرياضية في حمايتهم من هشاشة العظام، وأخيرا يجب عليهم مزاولة بعض تمارين التوازن لتجنب السقوط".</td></tr>
<tr><th colspan="3">Automatic summarization by proposed method and competitors</th></tr>
<tr><th>Model</th><th>Common sentences in the reference summary</th><th>System generated summary</th></tr>
<tr><td>VAE</td><td>3</td><td>- كشفت دراسة حديثة أجريت في تايوان أن أنواعا معينة من جراحات إنقاص الوزن قد تضعف العظام وتسبب هشاشتها وتزيد مخاطر الإصابة بالكسور.
- وقال كبير الباحثين في الدراسة الدكتور كو تشين هوانغ -وهو من كلية الطب في جامعة تايوان الوطنية في تايبيه- إن الجسم يُحرم من الكثير من العناصر الغذائية عندما يخضع المرضى لمثل هذه الجراحات.
- وبشكل عام، وجد الباحثون أن المرضى الذين خضعوا للجراحة زادت مخاطر إصابتهم بالكسور بنسبة 21% خلال السنوات الخمس التالية للجراحة.
- وأضاف "ثانيا يمكن أن يساعد التعرض للشمس وممارسة التمارين الرياضية في حمايتهم من هشاشة العظام، وأخيرا يجب عليهم مزاولة بعض تمارين التوازن لتجنب السقوط".</td></tr>
<tr><td>AE</td><td>2</td><td>- كشفت دراسة حديثة أجريت في تايوان أن أنواعا معينة من جراحات إنقاص الوزن قد تضعف العظام وتسبب هشاشتها وتزيد مخاطر الإصابة بالكسور.
- وذكروا أن بحثا سابقا أشار بالفعل إلى أن هذه الجراحات قد تزيد مخاطر كسور العظام.
- وبشكل عام، وجد الباحثون أن المرضى الذين خضعوا للجراحة زادت مخاطر إصابتهم بالكسور بنسبة 21% خلال السنوات الخمس التالية للجراحة.
- وقال هوانغ إن جراحات علاج السمنة يمكن أن تقلل احتمالات الإصابة بأمراض مثل السكري من النوع الثاني وارتفاع ضغط الدم.</td></tr>
<tr><td>LSA</td><td>2</td><td>- كشفت دراسة حديثة أجريت في تايوان أن أنواعا معينة من جراحات إنقاص الوزن قد تضعف العظام وتسبب هشاشتها وتزيد مخاطر الإصابة بالكسور.
- وقال كبير الباحثين في الدراسة الدكتور كو تشين هوانغ -وهو من كلية الطب في جامعة تايوان الوطنية في تايبيه- إن الجسم يُحرم من الكثير من العناصر الغذائية عندما يخضع المرضى لمثل هذه الجراحات.
- وقال هوانغ إن "العناصر الغذائية التي يُحرم منها الجسم في الغالب هي فيتامين "د" والكالسيوم والتي ترتبط بالإصابة بـهشاشة العظام، وربما توجد آليات أخرى ترتبط بالإصابة بالكسور".
- ومن خلال قاعدة بيانات التأمين الصحي الوطنية، تتبع الباحثون 2064 مريضا خضعوا لجراحات علاج السمنة في الفترة من 2001 إلى 2009، و5027 مريضا بالسمنة لم يخضعوا لهذه الجراحات.</td></tr>
<tr><td>TextRank</td><td>2</td><td>- كشفت دراسة حديثة أجريت في تايوان أن أنواعا معينة من جراحات إنقاص الوزن قد تضعف العظام وتسبب هشاشتها وتزيد مخاطر الإصابة بالكسور.
- وقال كبير الباحثين في الدراسة الدكتور كو تشين هوانغ -وهو من كلية الطب في جامعة تايوان الوطنية في تايبيه- إن الجسم يُحرم من الكثير من العناصر الغذائية عندما يخضع المرضى لمثل هذه الجراحات.
- وقال هوانغ إن "العناصر الغذائية التي يُحرم منها الجسم في الغالب هي فيتامين "د" والكالسيوم والتي ترتبط بالإصابة بـهشاشة العظام، وربما توجد آليات أخرى ترتبط بالإصابة بالكسور".
- وكتب هوانغ وزملاؤه في دورية الطب، أنه خلال العقد الماضي زادت جراحات علاج السمنة -وهي تقنية تستخدم إما في تصغير حجم المعدة وإما تحويل مسار أجزاء من القناة الهضمية - سبعة أمثال.</td></tr>
<tr><td>LexRank</td><td>1</td><td>- وقال هوانغ إن "العناصر الغذائية التي يُحرم منها الجسم في الغالب هي فيتامين "د" والكالسيوم والتي ترتبط بالإصابة بـهشاشة العظام، وربما توجد آليات أخرى ترتبط بالإصابة بالكسور".
- وبشكل عام، وجد الباحثون أن المرضى الذين خضعوا للجراحة زادت مخاطر إصابتهم بالكسور بنسبة 21% خلال السنوات الخمس التالية للجراحة.
- وقال هوانغ إن جراحات علاج السمنة يمكن أن تقلل احتمالات الإصابة بأمراض مثل السكري من النوع الثاني وارتفاع ضغط الدم.
- وأضاف "ثانيا يمكن أن يساعد التعرض للشمس وممارسة التمارين الرياضية في حمايتهم من هشاشة العظام، وأخيرا يجب عليهم مزاولة بعض تمارين التوازن لتجنب السقوط".</td></tr>
<tr><td>Tf.ISF</td><td>1</td><td>- وقال هوانغ إن "العناصر الغذائية التي يُحرم منها الجسم في الغالب هي فيتامين "د" والكالسيوم والتي ترتبط بالإصابة بـهشاشة العظام، وربما توجد آليات أخرى ترتبط بالإصابة بالكسور".
- وكتب هوانغ وزملاؤه في دورية الطب، أنه خلال العقد الماضي زادت جراحات علاج السمنة -وهي تقنية تستخدم إما في تصغير حجم المعدة وإما تحويل مسار أجزاء من القناة الهضمية - سبعة أمثال.
- ومن خلال قاعدة بيانات التأمين الصحي الوطنية، تتبع الباحثون 2064 مريضا خضعوا لجراحات علاج السمنة في الفترة من 2001 إلى 2009، و5027 مريضا بالسمنة لم يخضعوا لهذه الجراحات.
- وأضاف "ثانيا يمكن أن يساعد التعرض للشمس وممارسة التمارين الرياضية في حمايتهم من هشاشة العظام، وأخيرا يجب عليهم مزاولة بعض تمارين التوازن لتجنب السقوط".</td></tr>
</table>

3.4.1.4 Parameters Analysis

There are several parameters that affect the quality of the proposed model. For the parameters related to our model, such as learning rate and optimization algorithm, we follow the general setting for simplicity (Kingma and Welling, 2014). Other parameters such as the number of units in hidden layers and the number of epochs in the training stage have been taken into account. In our experimentation, the proposed model is composed of 2 hidden layers, 250 units in the first hidden layer and 20 in the second hidden layer. The last hidden layer with 20 units provide a 20-dimensional concept space which gives us a semantic representation of each sentence. Table 3.8 shows the results of the model according to the several structures. The model called VAE_250H1_20H2_25Eps designs a model with 250 units in the first hidden layer (H1), 20 units in the second hidden layer (H2) and 25 iterations (Epochs) through the whole training dataset (Eps). We change the number of layer H2 from 20 to 10 and 60, and the number of epochs from 25 to 5. From Table 3.8, we can see that the better results are achieved by the typical proposed model (VAE_250H1_20H2_25Eps). We conclude that the performance of the system decreases if the network parameters are not set properly.

Table 3.9 Performance comparison of typical model with different structures on EASC dataset

Model	10%	20%	30%	40%
VAE_250H1_20H2_25Eps	**0.110**	**0.282**	**0.402**	**0.529**
VAE_250H1_10H2_25Eps	0.108	0.281	0.393	0.512
VAE_250H1_60H2_25Eps	0.098	0.271	0.389	0.507
VAE_250H1_20H2_5Eps	0.096	0.237	0.345	0.456

3.5 Conclusion

In this chapter, we have presented a new Arabic summarization method based on unsupervised deep learning model. We have adopted the variational auto-encoder (VAE) to learn a compressed concept space from a high-dimensional input data in order to incorporate implicit semantic relations and automatically learn high level of features with an unsupervised technique. The deep architecture of our VAE is divided into two parts. First, the encoder is employed to map sentences from term vector space to latent semantic space. Then, the decoder, as an unsupervised data reconstruction, is used to conduct salience estimation, by reconstructing latent semantic space and observed term vectors.

The main contributions of this chapter are summarized as follows: (1) To the best of our knowledge, our work is the first attempt in using deep learning methods for automatic summarization of Arabic documents; (2) Drawing on the outstanding results achieved by auto-encoder models as an unsupervised learning method used in many tasks, a novel framework based on VAE is proposed in this work to extract relevant concepts from large features vectors. The proposed method demonstrates excellent extraction ability and better summary quality even compared with some extractive approaches. (3) We have investigated the ability of our proposed model on two summarization approaches: graph-based approach and query-based approach. We have found that our model improves the performances of both summarization techniques. (4) A series of experiments were performed on the EASC and our own datasets, using graph-based and query-based summarization approaches. We have ran our experimentations using global and local vocabularies with different length in order to construct word representations as the input of the VAE. The input word representation is constructed from both term frequency (*tf*) and *tf-idf* features. Comparing the results of global *tf-idf* with and without VAE demonstrates that the VAE provides a more discriminative feature space in which the semantic similarity measure is more accurate. More precisely, VAE outperforms significantly the *tf-idf* state-of-the-art term matching baselines. We assess how VAE handles a sparse word representation such as *tf-idf* and how the model works when using word representation based on local term frequency (*Ltf*) of a document-specific vocabulary. Comparison with other state-of-the-art approaches shows significant improvement in the propose approach. (v) Furthermore, our approach is completely unsupervised and does not need any annotated corpus for any stage of training, making our approach more suitable owing to the shortage of labeled summarization corpus designed for Arabic language.

The main disadvantages of our approach are: First, the training time of large datasets is very long. This is a common problem with all deep learning models. Second, finding optimal parameters for the network is a hard problem. In our work, a series of experiments have been carried out in order to find parameters that provide the best results. Nonetheless, these series of experiments are very time consuming.

In the next chapter, we propose several models for summarizing Arabic documents. We investigate many unsupervised feature learning techniques based on deep learning. Word embedding, Auto-encoder, extreme learning machine and ensemble learning techniques are some unsupervised learning models that we plan to study in the next chapter.

Chapter 4

Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning

4.1 Introduction

Sentence ranking is a key problem in all extractive summarization methods. Much research has been done to improve the quality of this process. Some works used statistical features (Luhn, 1958; Ferreira et al., 2013b; Ferreira et al., 2014) and some approaches are based on graphs (Erkan and Radev, 2004; Mihalcea and Tarau, 2004; Baralis et al., 2013), while others adopted supervised and unsupervised machine learning techniques (Fattah, 2014; Yang et al., 2014; Alguliyev et al., 2015). After investigating these works, we have found that they rely on bag-of-words (BOWs) approach in sentence representation. BOWs representation can cause two main problems. First, the system does not have enough observing data in the training stage. Thus, traditional systems use a sparse word representation as input (Yousefi-Azar and Hamey, 2017), which mean that many of the values are zero. Second, the semantic relationship between words is ignored. Moreover, it has been shown that distributed representation of words outperforms BOWs representation in capturing the semantics of the input text.

After investigating traditional Arabic text summarization systems, we have found that they suffer from the same problems. They are based on bag-of-words representation, which involve a sparse and high-dimensional input data. Thus, dimensionality reduction is greatly needed to increase the power of features discrimination.

In the previous chapter, we have introduced a deep learning based Arabic text summarization method, which uses the VAE to learn unsupervised features. The proposed model was trained on the traditional BOW representation, which is the TF.IDF matrix of the training corpus. In this chapter, we propose a further development of the previous work by investigating other neural networks models. The particularity of these models is that they are trained on a distributed representation of Arabic words, which we have built from a large Arabic corpus. The difference between this chapter and the previous are:

- In this chapter, in addition to the VAE, we adopt several neural networks models namely word embedding, Auto-encoder and Extreme learning machine.
- We build our word embedding model by training a large Arabic dataset taken from several sources

- We train our proposed models using word embedding representation instead of the traditional BOW representation used in the previous chapter.
- We propose new approaches based on ensemble learning techniques that average information provided from different models
- In addition to Arabic corpus, we evaluate the proposed models on English corpus in order to confirm that our approaches improve the automatic summarization of English documents

Deep neural networks have proven their ability to achieve excellent performance in many real-world NLP and computer vision applications. However, it still lacks attention in ATS, especially for Arabic. The key problem of traditional applications is that they involve high dimensional and sparse data, which makes it difficult to capture relevant information. One technique for overcoming these problems is learning features via dimensionality reduction. The aim is to discover the underlying low dimensional structure from the given high dimensional data.

We should note that document representation is an important phase in many machine learning algorithm used in the context of NLP. This phase allows the conversion of the text to be processed into numerical values, which are represented as input vectors to these kinds of algorithms. In ATS, BOW is the most frequently technique used to transform the original text into numerical vectors. In the BOW model, documents (or sentences) in the corpus are represented by a matrix of vectors in which each row represents the document (or sentence) and each column corresponds to a word generated from the vocabulary of the corpus. The value associated with each row and column relies on metrics based on word frequency. This approach, despite its simplicity, it suffers from two main problems. Firstly, it provides a sparse data in a high dimensional vector space, which impact negatively the performance of the classifier. Secondly, the semantic relation between different text units is ignored and not captured by the BOW representation.

In recent years, important new findings have been made to accomplish this transformation from documents to numerical vectors. Word Embedding (WE) is one of those techniques that allow such transformation. WE is another neural network technique that generates a much more compact word representation than a traditional Bag-of-Words (BOW) approach. It allow to represent words of a specific vocabulary with vectors in a low-dimensional space. This vector representation presents several advantages: i) it is amenable to be processed by machine learning and deep learning techniques; ii) it is a more powerful and effective representation which provide a dimensionality reduction; iii) it is a more expressive representation so it produces an efficient contextual similarity. Taking into account the context in which words appear in the corpus, an unsupervised learning algorithm is used to build the word embedding representation facilitating the understanding of syntactic and semantic meaning of those words and, therefore, improves the performance of many NLP tasks. Word2Vec is one of the well-known techniques used to produce WE. In recent years, this technique had paid special attention by the scientific community. It is based on a two-layer neural network whose input is a text corpus and the output is a set of numerical vectors representing each word in that corpus.

On the other hand, deep learning techniques (DL) have been successfully used as a base model for the representation of different kinds of data in a low dimensional vector space. DL is a particular machine learning approach whose main goal is to learn a high-level representation from lower-level representation. It has shown significant achievements in various areas, especially in computer vision (Wang et al., 2016; Donahue et al., 2017; Kahou et al., 2015; Li et al., 2017b), and audio processing (Lin et al., 2016; Li, Wang and Kot, 2017; Sun et al., 2017; Spille et al, 2018). Recently, DL techniques have achieved excellent results in NLP tasks (Er et al., 2016; Li et al., 2017a; Ayinde et al., 2017; Firat et al., 2017; Yousefi-Azar and Hamey, 2017).

The lack of labeled data used in training supervised models, make unsupervised neural networks techniques more suitable while unlabeled data are heavily available. For this purpose, many unsupervised deep learning models have been proposed in order to learn features from unlabeled data, therefore, the problem of a shortage in labeled data has become out of date. Examples of such models used in this work are Auto-Encoder (AE), Variational Auto-Encoder (VAE) and Extreme Learning Machine Auto-Encoder (ELM-AE).

In this chapter, we are seeking to enhance the quality of ATS by integrating unsupervised deep neural network techniques with word embedding approach. We adopt new concepts based on neural networks in order to build an effective representation of documents for Arabic summarization task. We propose several new models for Arabic text summarization. Those models are based on deep neural networks algorithms such as Word Embedding (WE), Extreme Learning Machine (ELM) and unsupervised deep learning. First, we develop a word embedding based text summarization, and we show that Word2Vec representation gives better results than traditional BOW representation. Second, we propose other models by combining word2vec and unsupervised feature learning methods in order to merge information from different sources. We show that unsupervised neural networks techniques using Word2Vec representation give better results than those learned from BOW representation.

We explore the sentence similarity measure based on hierarchical concept representations learned from different unsupervised models. The main goal is to predict concept importance and select accordingly the most important sentences to be included in the summary. We propose several models in order to compute the similarity measure between sentences. Firstly, we use the traditional BOW approach as the baseline model for the representation of documents with numerical vectors. Secondly, we use Sentence2Vec representation, which is based on the well-known Word2Vec representation. While Word2Vec represents each word as a vector, sentence2vec represents each sentence in the document as a vector in an embedded low-dimensional space. We compute the sentence2vec vector based on the average of all Word2Vec vectors in a sentence. Using this representation, a model of automatic text summarization is proposed in this work. Thirdly, we explore the unsupervised feature learning techniques, which aim to obtain a new representation of an input data in an abstract concept space. They learn a latent representation of the data by using unsupervised neural network techniques. In this work, we have developed two summarization frameworks based on the well-known unsupervised deep learning models called Auto-Encoder (AE) and Variational Auto-Encoder (VAE). Fourthly, Extreme Learning Machine (ELM), proposed by (Huang et al., 2006), has become a state-of-the-art learning framework (Liu et al, 2018) and has been successfully applied to computer vision (Cao et al., 2016) and bioinformatics (Lu et al., 2016). In this work, the unsupervised version of ELM, called ELM-Auto-Encoder (ELM-AE) (Kasun et al., 2013) has been proposed as a model of automatic text summarization. Lastly, we propose a combination of the main unsupervised feature learning approaches through several models, in which the information provided by many kinds of features are merged. In particular, we consider three kinds of ensemble learning methods, where several extracted features trained with several kinds of unsupervised neural networks are combined. The first ensemble combines BOW and word2vec using a majority voting technique. The second ensemble aggregates the information provided by the BOW approach and unsupervised neural networks. The third ensemble aggregates the information provided by Word2Vec and unsupervised neural networks models.

The main objective of this chapter is to evaluate the usefulness of Word2Vec and unsupervised feature learning models in Arabic documents summarization task. Our main goal is to show if the semantic representation offered by these models can improve the results of automatic summarization task performed by the traditional BOW approach. In order to show the complementarity of the proposed methods, we conduct our experiments on two different datasets publically available and designed specially to evaluate the quality of text summarization systems. The first dataset is a set of Arabic document collected from various Arabic newspapers. The summarization approach used for this dataset is based on a graph model. We build our word2vec model by training a large Arabic document corpus extracted from CNN, BBC and Wikipedia documents. The second dataset is a set of publicly available English emails. Two summarization approaches have been investigated for this dataset: graph-based and subject-based summarization approaches. We have used the existing word2vec model published by google. We show that the ensemble methods improve the quality of ATS, in particular the ensemble based on Word2vec approach gives better results. Finally, we perform different experiments to evaluate the performance of the investigated models. Results of statistical studies affirm that word embedding-based models outperform the summarization task compared to those based on BOW approach. In particular, ensemble learning technique with Word2Vec representation surpass all the investigated models. A statistical study on the results obtained by the proposed models shows the following:

1. Word2Vec approach provides significant improvements over the BOW approach.
2. The Word2Vec representation improves the results obtained by unsupervised deep learning models
3. The representation provided by unsupervised deep learning models improves significantly the results obtained when using the BOW approach.
4. The performance of the summarization system is improved when the networks are trained on Sentence2Vec vectors. This means that the combination of Word2Vec and neural networks gives better results than using neural networks with BOW representation.
5. The best results are obtained with the Ensemble of unsupervised deep neural network models that use sentence2vec representation as the input for training the combined models.

In the next sections, we describe the investigated models for automatic text summarization. The system evaluation and the experimental findings are detailed in section 4.3. The conclusion is presented in section 4.4.

4.2 Proposed models

In this section, we show the foundation of our proposed models designed for ATS. AE-based text summarization has already been proposed in previous works (Zhong et al., 2015); and it has proved to be effective, particularly with an ensemble technique (Yousefi-Azar and Hamey, 2017). VAE-based text summarization has also been proposed by Alami et al. (2018) and has been shown significant improvement in ATS for Arabic documents. Our proposed models are different from those of the previous studies. We use word2vec model as the input for training our models instead of the BOW representation used in the state-of-the-art. Moreover, we introduce a new model based on ELM-AE for the text summarization task. We investigate the impact of training the ELM-AE model using both BOW and word2vec approaches. Finally, we propose three new ensemble techniques that combine the results provided by different investigated models through a voting technique.

To the best of our knowledge, a hybrid approach in which unsupervised feature learning with neural networks (deep learning and ELM) and ensemble techniques are used for automatic text summarization has not been studied.

4.2.1 AE-based model

Recently, the use of unsupervised learning techniques has become very promising in many applications due to the increasing availability of unlabeled data. An auto-encoder (AE) (Figure 4.1) is a feed forward neural network which attempt to learn unsupervised data by reconstructing its input. A simple AE consists of 3 layers: an input layer *x*, hidden layer *z* and output layer *y* which is similar to the input *x*. The AE is trained to encode (compress) the input vector into a smaller hidden representation (concept space). Then, the compressed features (latent representation) are passed through the decoder trying to reconstruct (decode) its input. Back-propagation algorithm is used to train such network. The goal of training is to minimize the mean-square error between the input data *x* and its approximate reconstruction x̂. In the case where there is on hidden layer, the auto-encoder performs in two phases:

i) the encoder phase, which maps the input x to the concept space z (code, latent variables, or latent representation) by using the following function:

$$z = \sigma(Wx + b) = \sigma(\sum_{i,j} w_{ij} x_i + b_j) \tag{4.1}$$

Here, σ refers to an activation function such as sigmoid or rectified linear unit (ReLU). W is a weight matrix and b is a bias vector;

ii) After that, the decoder phase maps z to the reconstruction $\hat{x}$ of the same input x:

$$\hat{x} = \sigma'(W'z + b') = \sigma'(\sum_{i,j} w'_{ij} z_i + b_j) \tag{4.2}$$

After initializing the parameters weights of the AE with the appropriate values, the fine-tuning phase is performed to globally adjust the whole network parameters by applying back-propagation and gradient descent algorithm for optimal reconstruction. In this stage, the unsupervised learning algorithm is performed to minimize the reconstruction errors (loss function). For real-valued inputs, the loss function is represented by the Mean squared error given by:

$$\frac{1}{N}\sum_{i=1}^{N}(\widehat{x_i} - x_i)^2 \qquad (4.3)$$

where N is the size of the input vector, it represents the total number of items in the training data.

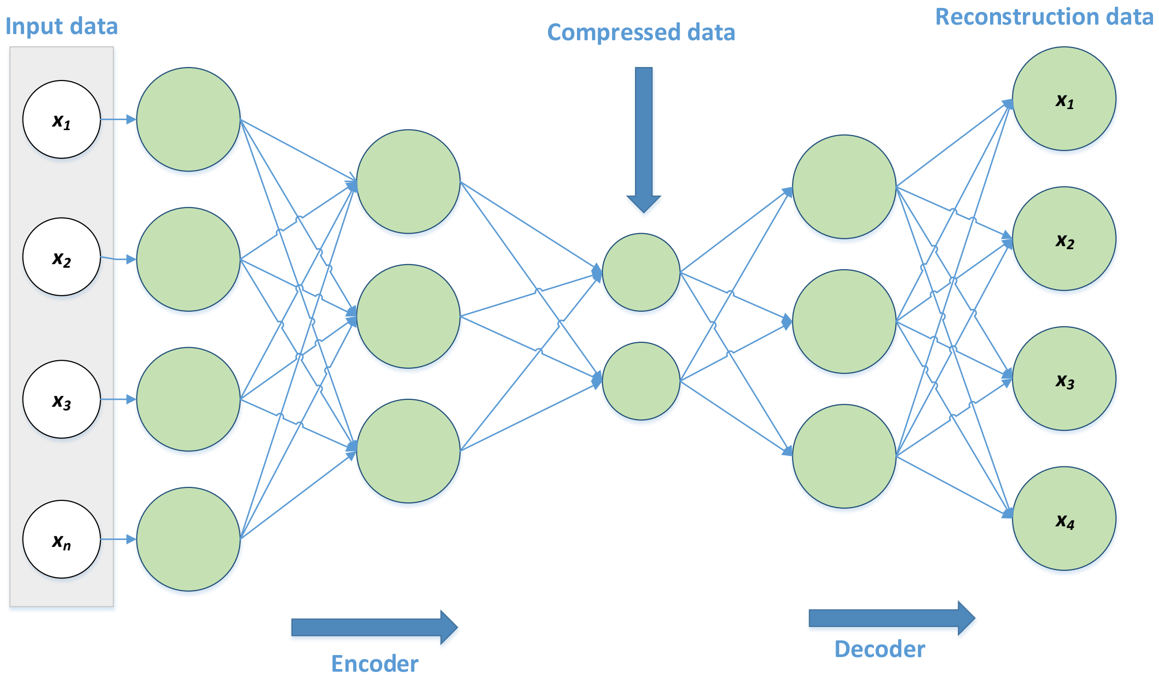


Figure 4.1 Topology of the Auto-Encoder.

4.2.2 VAE-based model

VAE (Kingma and Welling, 2014; Rezende et al., 2014) is a new generation of AEs, which benefits from the powerful of both neural network techniques and generative models. It represented by two networks: an encoder that maps the input data x to a latent representation and a decoder that decodes the latent representation z to the reconstruction $\hat{x}$ of the same input x:

$$z = Encoder(x) = q(z|x), \ \ \hat{x} = Decoder(z) = p(x|z). \qquad (4.4)$$

For more details on training VAEs, see section 3.3.2 of chapter 3.

4.2.3 ELM-AE based model

Given an input data point x, the output of the ELM network is given by a mapping function to M-dimensional ELM random feature space:

$$f_M(x) = \sum_{i=1}^{M}\beta_i h_i(x) = h(x)\beta \qquad (4.5)$$

Where $\beta = [\beta_1, \dots, \beta_M]^T$ is the output weight matrix between the hidden nodes and the output nodes, $\mathrm{h(x)} = [\mathrm{h_1(x)}, \dots, \mathrm{h_M(x)}]$ are the hidden node outputs for input x, and $\mathrm{h_i(x)}$ is the output of the i^{th} hidden node. Given N training samples $\{(x_i, t_i)\}_{i=1}^{N}$, the following learning problem is addressed by ELM:

$$H\beta = T \tag{4.6}$$

Where $[t_1, \dots, t_N]^T$ are target labels, and $H = [h^T(x_1), \dots, h^T(x_N)]^T$.

Algorithm 4.1. Extreme Learning Machine

```
Input: Training set $S = \{(x_i, t_i)\} | x_i \in R^n, t_i \in R^m, i = 1, 2, \dots, N$, activation function $g(x)$, number of neurons in hidden layer $M$;
Output: Weight matrix $\beta$;
1.  Initialize the input weight matrix $W$ and hidden layer bias $b$ with random values;
2.  Using the activation function g, calculate the hidden layer output matrix $H$ with:
        $H = g(Wx + b)$
3.  Calculate the network output weight matrix $\beta$ using Equation (4.8)
```

The output weights matrix β is calculated using the following formulas:

$$\beta = H^{\dagger}T \tag{4.7}$$

Where $H^{\dagger}$ is the Moore-Penrose generalized inverse (pseudoinverse) of the output matrix H.

Despite the evident advantages of ELM in generalization and training speed, it suffers from bad generalization performances. Deng et al. (2010) address this problem by proposing a new ELM model called Regularized Extreme Learning Machine (RELM), which aims to minimize the least squares estimation cost function by adding a regularization coefficient C as shown in the following formulation:

$$\beta = (\frac{1}{C} + H^T H)^{-1} H^T T \tag{4.8}$$

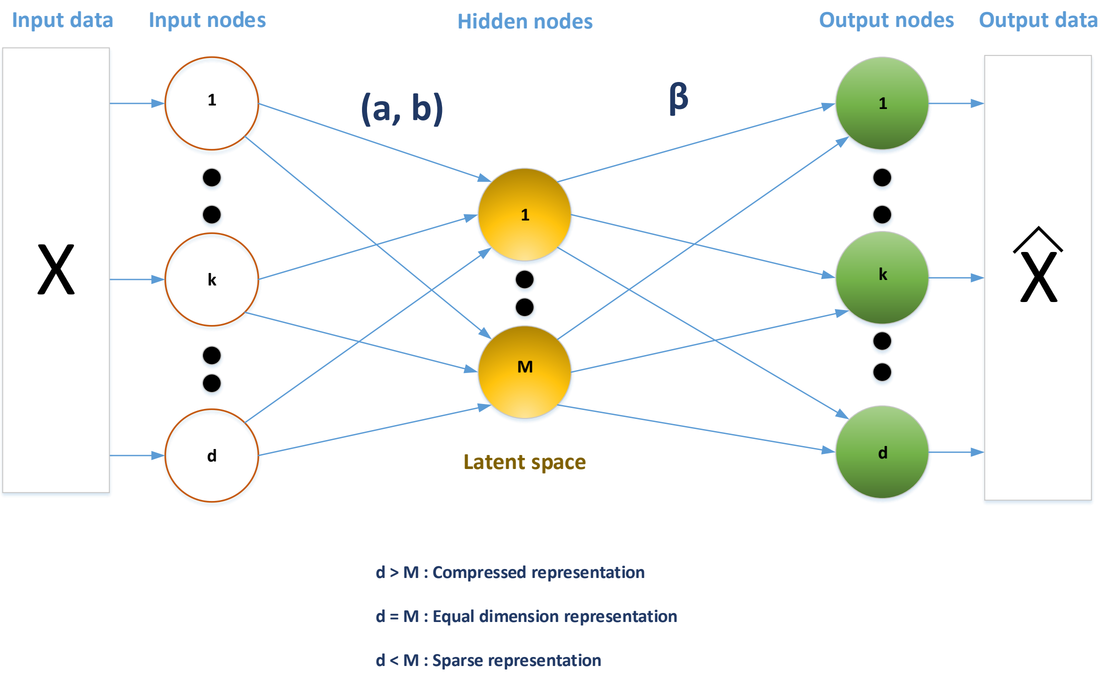

Input data
Input nodes
Hidden nodes
Output nodes
Output data
X
1
k
d
(a, b)
β
1
M
Latent space
1
k
d
$\hat{X}$
d > M : Compressed representation
d = M : Equal dimension representation
d < M : Sparse representation

Figure 4.2. ELM-AE model. The input X is the same as the output $\hat{X}$, (a, b) are the randomly generated hidden node parameters which are made orthogonal.

Algorithm (4.1) outlines the main steps of ELM. The basic version of ELM is designed to learn features from labeled data, while unlabeled data is much more widely available due to the digital transformation around the word. Unlabeled data need an unsupervised technique in order to learn, extract features and reduce the dimensionality of this data. With this rising need in mind and to address the challenge of training unsupervised tasks, a new unsupervised version of ELM called extreme learning machine auto-encoder (ELM-AE) was proposed by Kasun et al. (2013). Based on ELM, the ELM-AE is a neural network with a single hidden layer and the input data is the same as the output. The initial weights and biases of the hidden nodes are randomly generated and should be orthogonal. Figure 4.2 illustrates the network architecture of ELM-AE.

The process of training an ELM-AE is done in two main stages: encoder stage and decoder stage. In the first step (encoder stage), the input features are mapped into a M dimensional feature space in three different ways according to the size of d and M: 1) $d < M$, sparse architecture, which represents features from a lower dimensional input data space to a higher dimensional feature space; 2) $d > M$, compressed architecture, which represents features from a higher dimensional data space to a lower dimensional feature space; 3) $d = M$, equal dimension, which represents features from an input data space dimension equal to feature space dimension.

In this work, we are interested in the compressed architecture of ELM-AE. In this architecture, the random orthogonal weights and biases of hidden nodes map the input data x_i to the lower dimensional M space by using the following formula:

$$h(x_i) = g(a^T x_i + b) \tag{4.9}$$

$$a^T a = I, b^T b = 1 \tag{4.10}$$

Where $a = [a_1, \dots , a_M]$ are the orthogonal random weights, and $b = [b_1, \dots , b_M]$ are the orthogonal random biases between the input and hidden nodes. $h(x_i) \in R^M$ is the output vector of the hidden layer with respect to the input x_i; $g(.)$ is an activation function which can be sigmoid, Gaussian function or so on; I is an identity matrix of order M. We use the sigmoid function in the encoder stage of the ELM-AE.

In the second step (decoder stage), the output weights β are updated by minimizing the squared error objective. The following formula shows the mathematical model for training ELM-AE:

$$\min_{\beta \in R^{M \times d}} L_{ELM-AE} = \min_{\beta \in R^{M \times d}} L_{ELM-AE} \frac{1}{2} \|\beta\|^2 + \frac{C}{2} \|X - H\beta\|^2 \tag{4.11}$$

where C is a penalty coefficient on the training errors. It balances experiential risk and structural risk. By setting the gradient of L_{ELM-AE} to zero, we have:

$$\beta + CH^T(X - H\beta) = 0 \tag{4.12}$$

According to the above equation, the output weights β of an ELM-AE can be computed in three different ways:

- When the number of training samples N is larger than the number of hidden layer nodes M, output weights are calculated by Equation (4.13). This is a compressed ELM-AE representation.
- When the number of training samples N is smaller than the number of hidden layer nodes M, output weights are calculated by Equation (4.14). This is a sparse ELM-AE representation.
- For equal dimension ($N = M$), output weights can be expressed as Equation (4.15). This is an equal ELM-AE representation.

$$\beta = (\frac{I_M}{C} + H^T H)^{-1} H^T X \tag{4.13}$$

$$\beta = H^T (\frac{I_N}{C} + HH^T)^{-1} X \tag{4.14}$$

$$\beta = H^{-1} X \tag{4.15}$$

Where $I_k$is an identity matrix of dimension k.

The main focus of this chapter is to use the compressed data instead of real input data in the automatic summarization task. Dimensionality reduction is achieved by the unsupervised ELM-AE by projecting the input data X along the decoder stage. The new representation of the input X data in dimensional feature space n_h is given by the following formula:

$$X_{new} = X\beta^T \tag{4.16}$$

Thereafter, the original data (X) is replaced by the new generated data (X_{new}) in the summarization task. Algorithm (4.2) outlines the main steps of ELM-AE-based summarization model.

Algorithm 4.2. ELM-AE algorithm for summarization task

```
Input: input data $\{X\} = \{x_i\}_{i=1}^{N}$, the number of hidden neurons $M$, the penalty coefficient $C$
Output: transformed data $X_{new}$
1. Initialize the ELM-AE of $M$ hidden neurons with random orthogonal input weights and biases.
2. If $M < N$
           Calculate the output weights $\beta$ according to Equation (4.13)
   If $M > N$
           Calculate the output weights $\beta$ using Equation (4.14)
   If $M = N$
           Calculate the output weights $\beta$ using Equation (4.15)
3. Calculate the new data $X_{new}$ according to Equation (4.16)
4. Use $X_{new}$ in the summarization task instead of $X$
```

4.2.4 Sentence2Vec-based model

The most frequently method used to represent text in a vector form is the traditional bag-of-words (BOW) approach. This representation is based on a vocabulary existing in the corpus. In our case, we consider a sentence as a text unit and a document to be summarized as a set of sentences. Each word in the corpus is assigned with an *id* that represents its position in the dictionary. Let V represents the whole vocabulary in the corpus $V = \{w_1, w_2, \dots, w_n\}$, n is the size of the vocabulary V. Each sentence S is represented by a vector $S = \{f_1, f_2, \dots, f_n\}$ where f_i is the extracted feature of word w_i in the sentence S. There are multiple ways to compute f_i. It can be the frequency of word wi in the sentence S, or it can be one or zero depending on whether the word appears in the sentence or not. In our experiments, the value f_i represents the well-known *TF-IDF* measure, which represents the term frequency/inverse document frequency of a term.

The new approach for words representation provided by Word2Vec is an alternative of a BOW classical representation. Word2Vec (Mikolov et al., 2013) is an unsupervised learning method that aims to capture the semantic relationship between words based on their co-occurrence in documents of a specific corpus. The main idea of word2vec is to detect the context of words using deep learning approaches. There are two different learning models to produce the Word2vec representation: i) CBOW and Skipgram (Figures 4.3 and 4.4). In CBOW the goal is to predict a word giving its context (set of surrounding words), while in the Skipgram algorithm the goal is to predict the context of a given word. According to Mikolov et al. (2013), Skipgram works well with a small amounts of training data, whereas CBOW is much faster to train and the quality of representation is better for frequent words.

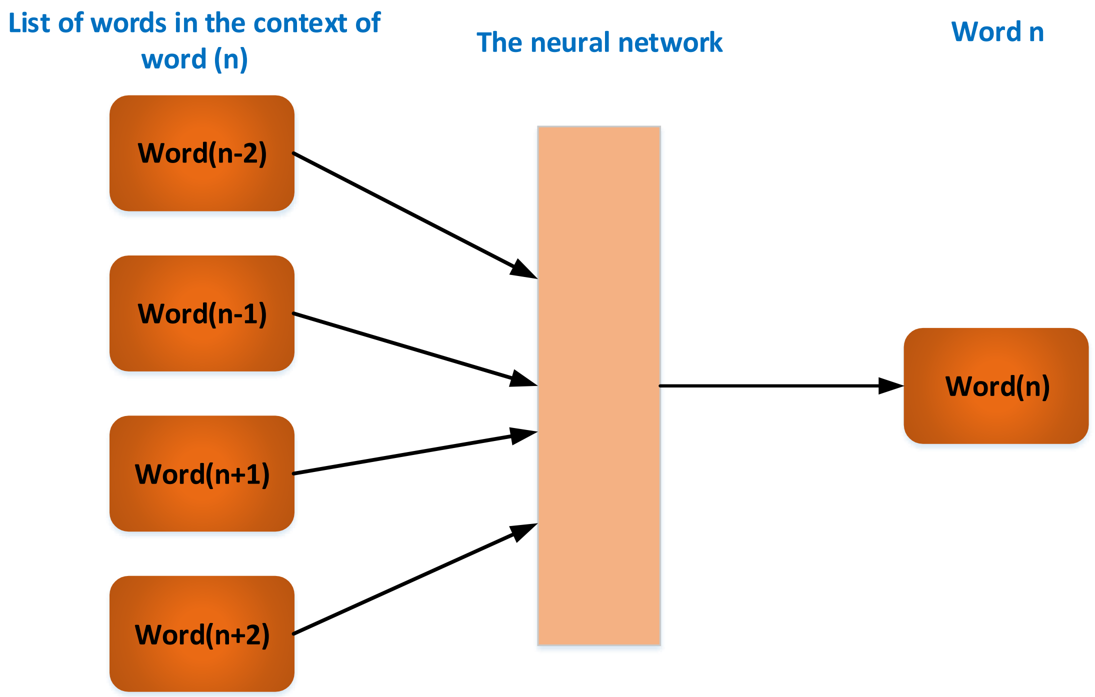


Figure 4.3 CBOW approach for word2vec

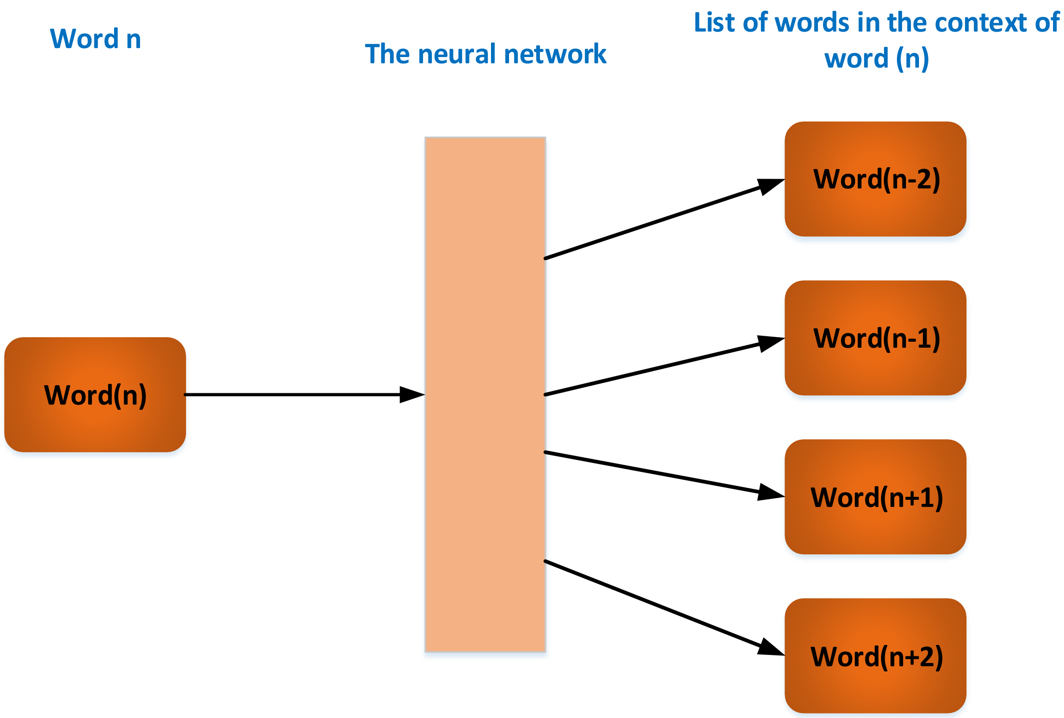


Figure 4.4 Skip-gram approach for Word2Vec

In order to build a Word2Vec representation of a given corpus, first a vocabulary based on the words in the corpus is constructed and, in order to avoid noise, only words that appear more times than a predefined threshold are considered, and then all documents are split into sentences and CBOW or Skip-gram algorithm is applied to learn word vector representation in a D-dimensional space. The output of Word2Vec is a set of vectors representing each word existing in a vocabulary of the trained corpus. After the training phase, words that are semantically close have vectors that are also close to each other. We have to note that in the preprocessing stage, the lemmatization task is not applied in order to allow the Word2Vec method to capture the semantic information of different word forms depending on the context.

In this work, sentence-level is explored in the ATS task. This require a method to generate a single vector representing the entire sentence from all word vectors existing in this sentence. While Word2Vec represents each word as vector, Sentence2Vec represents each sentence in the document by a vector in an embedded low-dimensional space. After testing several methods, the average of Word2Vec vectors of all the words in a sentence was chosen to compute Sentence2Vec vectors. The following formula is used to compute the vector of each sentence:

$$\overrightarrow{V_d} = \frac{\sum_{i=0}^{n} \overrightarrow{v_i}}{n} \tag{4.17}$$

4.2.5 Combination of Sentence2Vec and deep neural networks

In order to show the effectiveness of Sentence2Vec representation, we evaluate the three unsupervised neural networks by using Sentence2Vec matrix representation as the input for training the model:

- Sentence2Vec-based AE model (Sentence2Vec_AE): Sentence2Vec representation is used as the input of the AE instead of the BOW representation.
- Sentence2Vec-based VAE model (Sentence2Vec_VAE): Sentence2Vec representation is used as the input of the VAE instead of the BOW representation.
- Sentence2Vec-based ELM-AE model (Sentence2Vec_ELM-AE): Sentence2Vec representation is used as the input of the ELM-AE instead of the BOW representation.

4.2.6 Ensemble learning-based models

The proposed techniques distill an ensemble of models into a single model. In this chapter, we propose three ensemble learning techniques which aggregate the information provided by the features learned from different models. The first model aggregates the information provided by BOW and sentence2vec representation. The architecture of this model is shown in Figure 4.5. The second ensemble is based on BOW representation. It aggregates the information provided by BOW vectors and the features learned from AE, VAE and ELM-AE. Figure 4.6 illustrates the architecture of this model. The third Ensemble is based on Sentence2Vec representation. It aggregates the information provided by Sentence2Vec vectors and the features learned from AE, VAE and ELM-AE. Here, Sentence2Vec representation is used as the input of the learning models. Figure 4.7 illustrates the architecture of this model. In addition, we evaluated an ensemble composed with Sentence2Vec and BOW representation (Figure 4.8).

The document to be summarized is transformed into *TF-IDF* matrix (Feature extraction) in the first method and into sentence2vec for the second proposed method. The produced matrix is then used in order to train different models. The first model uses the produced matrix (BOW or Sentence2Vec representation) as the input of the summarization system. The second model uses the produced matrix in order to learn the features from VAE. The third model uses the produced matrix in order to learn the features from AE. The fourth model uses the produced matrix in order to learn the features from ELM-AE. The features learned from different models are used as the input of the summarization system. After that, the ranking obtained by different experiments is aggregated through an ensemble approach using the majority voting scheme in order to re-rank sentences and select the best ranked between them.

4.3 Experimental design and results

In order to have a thorough assessment of the proposed models, we perform several experiments on two publicly available datasets that are especially designed for summarization: Summarization and Keyword Extraction from Emails (SKE) (Loza et al., 2014); and The Essex Arabic Summaries Corpus (EASC) developed by El-Haj et al. 2010. We designed an experimental phase in which we compared the results of the summarization task using different document representation obtained with the proposed models. The dataset used in this work is described in section 1.9.1.

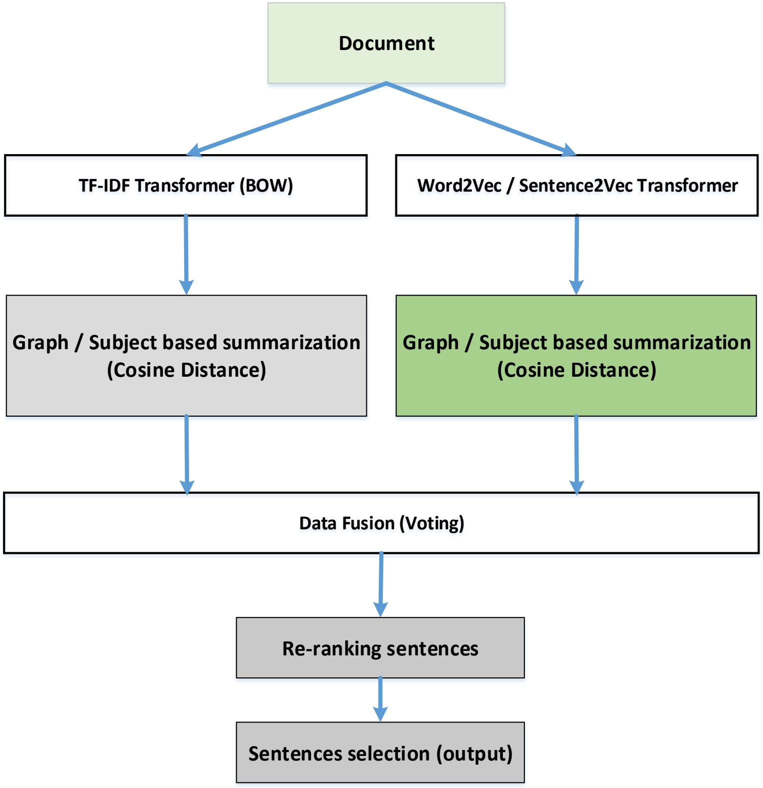

Document
TF-IDF Transformer (BOW)
Word2Vec / Sentence2Vec Transformer
Graph / Subject based summarization (Cosine Distance)
Graph / Subject based summarization (Cosine Distance)
Data Fusion (Voting)
Re-ranking sentences
Sentences selection (output)

Figure 4.5 The ensemble method combining BOW representation and word2vec/sentence2vec representation for text summarization.

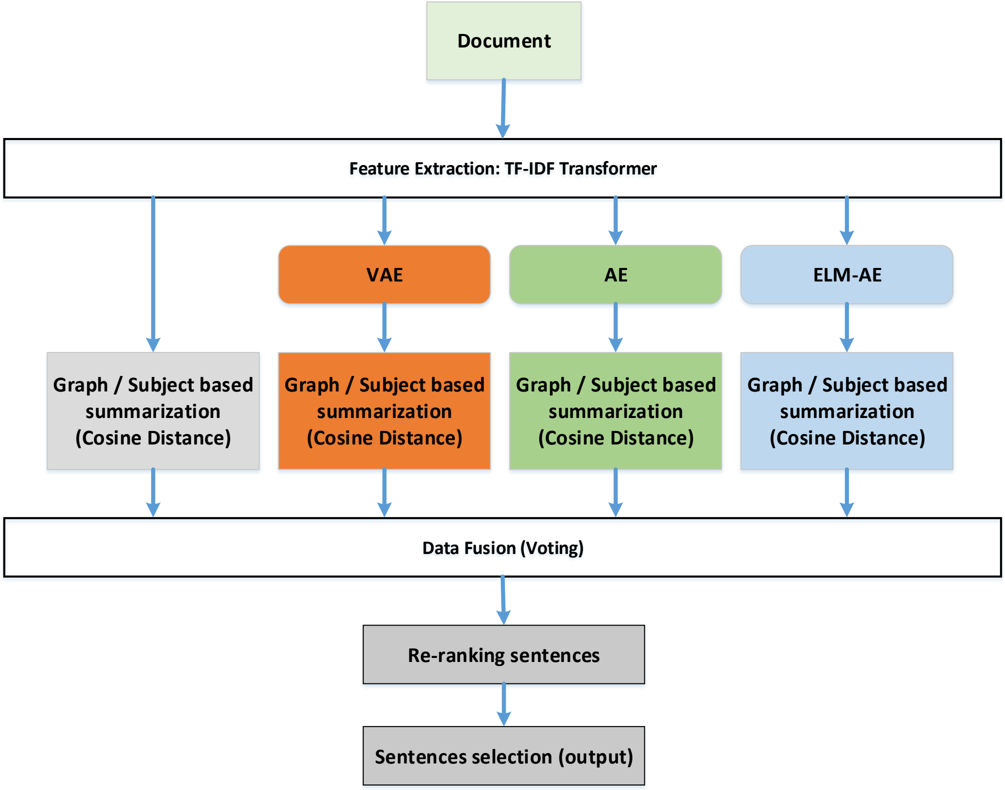

Document
Feature Extraction: TF-IDF Transformer
VAE
AE
ELM-AE
Graph / Subject based summarization (Cosine Distance)
Graph / Subject based summarization (Cosine Distance)
Graph / Subject based summarization (Cosine Distance)
Graph / Subject based summarization (Cosine Distance)
Data Fusion (Voting)
Re-ranking sentences
Sentences selection (output)

Figure 4.6 The ensemble of four models based on BOW representation for text summarization.

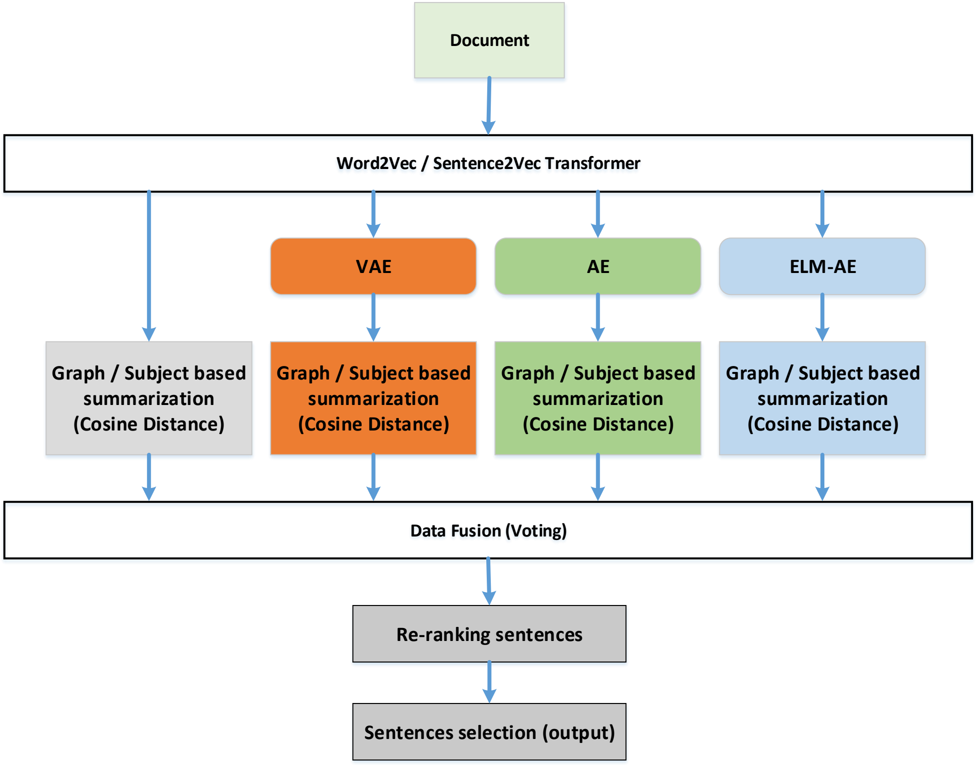

Document
Word2Vec / Sentence2Vec Transformer
VAE
AE
ELM-AE
Graph / Subject based summarization (Cosine Distance)
Graph / Subject based summarization (Cosine Distance)
Graph / Subject based summarization (Cosine Distance)
Graph / Subject based summarization (Cosine Distance)
Data Fusion (Voting)
Re-ranking sentences
Sentences selection (output)

Figure 4.7 The ensemble of four models based on word2vec/sentence2vec representation for text summarization.

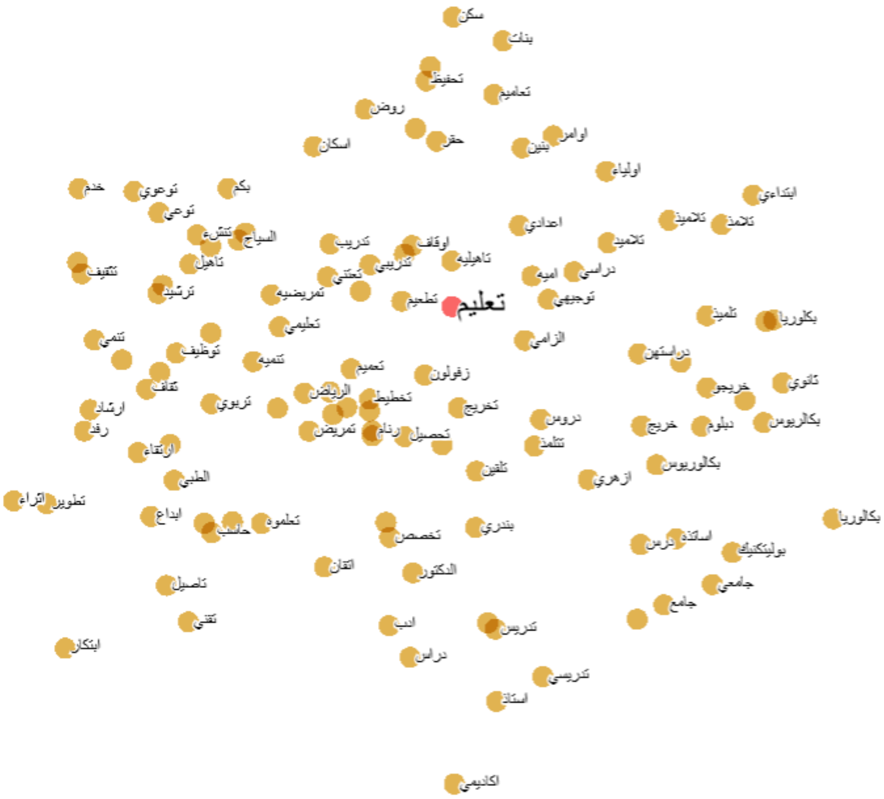


Figure 4.8 Sample of PCA projection of trained word2vec model

In the following sections we present how we obtained the word2vec model for Arabic and English datasets, the summarization techniques, the methods we compare with, implementation details and their results. To make reading easier, we provide for each model the following notation:

AE indicates the system based on the auto-encoder model. In our experimentation, the AE is composed of one hidden layer with 20 units. **VAE** indicates the system based on the variational auto-encoder model. In our experimentation, the VAE is composed of two hidden layer with 200 units in the first hidden layer and 20 units in the second layer. **ELM-AE** indicates the system based on the extreme learning machine auto-encoder model. In this work, the ELM-AE is composed of one hidden layer with 50 hidden units. **BOW_S2V** denotes the ensemble model combining BOW and sentence2vec models with a majority voting technique. **BOW_AE_VAE_ELM-AE** denotes the system based on the ensemble learning model trained on the BOW matrix representing the corpus. This model combines four summarization systems. The first system is the baseline summarization system which is based on the *TF.IDF* representation (BOW). The other systems are successively based on AE, VAE and ELM-AE models. **S2V_AE_VAE_ELM-AE** denotes the system based on the ensemble learning model trained on the sentence2vec matrix representing the corpus. This model combines four summarization systems. The first system uses sentence2vec matrix to build the summary. The other systems are successively based on AE, VAE and ELM-AE models which are trained on sentence2vec matrix representing the corpus.

4.3.1 Word2Vec model

To obtain the vector representation of Arabic words, a Skip-gram method has been chosen and trained on a large Arabic datasets composed by:

- Wikipedia corpus, which is the full database dump of Arabic articles freely provided by Wikipedia.
- CNN corpus (Saad and Ashour, 2010), which consists of 5,070 articles divided into 6 topics: Business, Entertainment, Middle East News, World News, Science and Technology, Sports. The dataset contains 2,241,348 words and 144,460 district keywords after removing stop-words.
- BBC corpus (Saad and Ashour, 2010), which consists of 4,763 articles divided into 7 topics: Middle East News, World News, Business and Economy, Sports, Science and Technology, Art and Culture, International Press. The dataset contains 1,860,786 words and 106,733 district keywords after removing stop-words.
- OSAC corpus (Saad and Ashour, 2010), which consists of 22,429 articles collected from multiple Arabic websites. The dataset is divided into 11 topics: Economics, History, Entertainment, Education and Family, Religious, Sports, Astronomy, Health, Law, Stories, and Cooking Recipes. The dataset contains about 18,183,511 words and 449,600 district keywords after removing stop-words.

Our Arabic word2vec model has been obtained using the Word2Vec implementation of Gensim python library. A vector of 200 dimensions has been generated for each word in the corpus. A sample of PCA projection of trained word2vec model is given in Figure 4.8.

The English word2vec model used in this work is freely provided to the community by Google through it Google's pre-trained vectors trained on part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases.

4.3.2 Summary generation

After training and fitting our models, each sentence is mapped into a concept space. Assuming we have a text document D with a set of sentences $S = \{s_1, s_2, \dots, s_m\}$, each sentence s_i is projected into a concept space by a mapping function given by a specific model. An abstract representation $\widehat{s_i}$ is produced and used in order to compute the similarity between two sentences using the cosine similarity metric (Equation (4.18)):

$$sim(S_i, S_j) = \frac{\widehat{S_i}.\widehat{S_j}}{\|\widehat{S_i}\|\|\widehat{S_j}\|} \tag{4.18}$$

Where $\widehat{S_i} = M(S_i)$ is the mapping function of the sentence S_i in the concept space of a specific model.

We build our summary based on the most relevant sentences in the document. This is known as extractive summarization or sentence ranking. Our proposed models rank the sentences based on their abstract representation in the concept space learned by the neural network. We investigate two extractive summarization techniques:

- **Graph-based summarization:** In graph-based summarization method, each sentence in the given document is represented by a node in the graph and the similarity between two sentences is represented by an edge between the correspondent nodes. The weight of each edge represent the similarity measure between two sentences. This similarity is calculated using the cosine similarity metric in the concept space as shown in Equation (4.19). In a graph-based summarization model, ranking sentences involving calculating the importance of a vertex within a graph, on the basis of the information elicited from the graph structure. PageRank algorithm was used to calculate a salient score for each vertex of the graph.
- **Query-based summarization:** In query-based summarization system, the score of each sentence is calculated according to it similarity to the given query using the following formula:

$$sim(S_i, Q) = \frac{\widehat{S_i}.\hat{Q}}{\|\widehat{S_i}\|\|\hat{Q}\|} \tag{4.19}$$

Where Q is the given query and S_i is the given sentence. $\hat{Q}$ and $\widehat{S_i}$are their mapping into the concept space. Sentences are ranked according to the highest score.

4.3.3 Results and discussion

4.3.3.1 Evaluation results on EASC dataset:

We investigate the performance of graph-based summarization system on EASC dataset using Rouge-1 recall. Table 4.1 shows the results in term of Rouge-1 recall with different summary length obtained by both BOW and Sentence2Vec approaches. It is clear from the obtained results that the new approach based on sentence2vec representation outperforms the classical approach based on BOW representation. Moreover, we can report that the proposed ensemble learning model (BOW_S2V) give good results compared to both models (BOW and Sentence2Vec). The ensemble model in this experimentation is built from the combination of the two models (BOW and Sentence2Vec) using majority voting technique. This leads us to conclude that the information contained in each vector are complementary to each other, and that is the reason why the combination achieves best results.

The results obtained by the proposed deep neural networks are exposed in Table 4.2. The unsupervised neural networks (AE, VAE and ELM-AE) are trained on both BOW and Sentence2Vec representation. We can see the difference between results obtained by the models trained on BOW representation and those trained on Sentence2Vec representation. For the models based on AE and ELM-AE, sentence2Vec representation give the best result compared to the BOW representation of the same model. For example, a Rouge-1 recall of obtained by the ELM-AE with 50 hidden layers in the latent space and a summary length of 20%, is 0.2473 when the model is trained on BOW representation and 0.2662 when the same model is trained on Sentence2Vec representation. These results confirm that the representation given by Sentence2Vec is more reliable and comprises more information as the one given by the traditional BOW representation. For VAE-based model, we can say that the results obtained by BOW and sentence2vec are close to each other.

Table 4.1 ROUGE-1 comparison between BOW approach and Sentence2Vec approach using graph-based summarization with EASC

	Summary length							
Model	10%	15%	20%	25%	30%	35%	40%	45%
BOW	0.0986	0.1667	0.2537	0.3254	0.3693	0.4382	0.4885	0.5176
Sentence2Vec	0.1299	0.2014	0.2924	0.3509	0.3953	0.4486	0.4969	0.5347
BOW_S2V	0.3185	0.3716	0.4305	0.4751	0.5064	0.5450	0.5798	0.6043

Table 4.2 ROUGE-1 recall of graph-based summarization with EASC using unsupervised neural network models trained on both BOW and Sentence2Vec representation

	BOW (TF-IDF)			Sentence2Vec		
Size	AE	VAE	ELM-AE	AE	VAE	ELM-AE
10%	0.0791	0.1101	0.0893	0.1024	0.1117	0.1120
15%	0.1357	0.1878	0.1636	0.1696	0.1797	0.1789
20%	0.2127	0.2825	0.2473	0.2484	0.2635	0.2662
25%	0.2762	0.3454	0.3094	0.3054	0.3291	0.3234
30%	0.3211	0.4021	0.3537	0.3515	0.3705	0.3658
35%	0.3753	0.4724	0.4128	0.4150	0.4328	0.4259
40%	0.4301	0.5298	0.4626	0.4652	0.4784	0.4746
45%	0.4663	0.5616	0.5009	0.5029	0.5141	0.5169

Table 4.3 shows the results obtained by the two proposed ensemble learning approaches. The first approach is a combination of sentence2vec model with unsupervised neural network models which are trained on sentence2vec representation. The second Ensemble technique is based on the combination of BOW model with neural network models trained on BOW representation. The ensemble model used in this experimentation is based on the majority

voting technique in order to obtain the final summary. The results show that the ensemble based on sentence2vec representation outperform the one based on BOW representation. For example, with a summary length of 20%, the rouge-1 result obtained by the former ensemble is 0.4444, while the rouge-1 result obtained by the latter is 0.3752. We conclude that sentence2vec, which is based on word2vec model, improves the quality of the final summary generated using the ensemble of deep neural networks models. We can also conclude that the ensemble learning model based on sentence2vec outperform all the proposed models and gives the better summary with significant improvement in the Rouge-1 recall for all the summary lengths.

Table 4.3 ROUGE-1 recall of graph-based summarization with EASC using Ensemble learning models

	Summary size							
Ensemble model	10%	15%	20%	25%	30%	35%	40%	45%
S2V_AE_VAE_ELM-AE	0.3265	0.3803	0.4444	0.4855	0.5147	0.5573	0.5910	0.6209
BOW_AE_VAE_ELM-AE	0.2812	0.3244	0.3752	0.4215	0.4672	0.5070	0.5579	0.5928

4.3.3.2 Evaluation results on SKE dataset

Graph-based summarization with SKE

Rouge-2 results of graph-based summarization with SKE dataset are presented in Table 4.4. The summary size is denoted by the variable n, which indicates the number of sentences extracted by the system. These results confirm those exposed in table 1, which mean that the summarization task is outperformed when using word2vec as a document representation model. Also, we can notice that the combination of BOW and sentence2vec through an ensemble model with the majority voting technique outperforms the BOW approach but not the sentence2vec approach. The result obtained by this ensemble approach is between the two models.

Table 4.4 ROUGE-2 recall of graph-based summarization using English corpus SKE

	Summary size (number of sentences)				
Model	n=1	n=2	n=3	n=4	n=5
BOW	0.1417	0.2679	0.3792	0.4651	0.5460
Sentence2Vec	0.1646	0.3012	0.4214	0.5136	0.5902
Ensemble: BOW_S2V	0.1556	0.2861	0.4083	0.4969	0.5771

Table 4.5 ROUGE-2 recall of graph-based summarization using English corpus (SKE)

	BOW (TF-IDF)			Sentence2Vec		
Size	AE	VAE	ELM-AE	AE	VAE	ELM-AE
n=1	0.0451	0.1035	0.1135	0.1492	0.1334	0.1441
n=2	0.0943	0.2013	0.2074	0.2812	0.2591	0.2671
n=3	0.1538	0.3068	0.3002	0.4045	0.3623	0.3725
n=4	0.2190	0.3966	0.3842	0.5122	0.4575	0.4677
n=5	0.2940	0.4868	0.4485	0.5957	0.5387	0.5453

The same applies to other unsupervised neural models proposed in this work. As noted in Table 4.5, all the proposed models, AE, VAE and ELM-AE give better results when using sentence2vec representation as the input of the network. We can conclude that the relevant information is expressed by sentence2vec.

Table 4.6 shows the Rouge 2 recall of graph-based summarization with SKE using Ensemble learning models. We can note that the summarization of SKE dataset using only sentence2vec gives better results than the summarization using the proposed Ensemble technique. This is inconsistent with what we have found previously in Table 4.3 (summarization with EASC). We note that we have used the same configuration of the network when using the both representation (BOW and sentence2vec). For this and in order to confirm or reverse the strength of our proposed ensemble methods, we choose another configuration when using sentence2vec representation.

To show the strength of our proposed ensemble methods, we perform an experiment with the following configuration of the neural network models: the AE is composed of 250 hidden units. We build the VAE with 250 units in the first hidden layer and 250 in the second hidden layer.

Table 4.6 ROUGE-2 recall of graph-based summarization with SKE using Ensemble learning models

	Summary size (number of sentences)				
Ensemble model	n=1	n=2	n=3	n=4	n=5
S2V_AE_ VAE_ELM-AE	0.1637	0.2931	0.4169	0.5097	0.5909
BOW_AE_VAE_ELM-AE	0.1061	0.2064	0.3133	0.3989	0.4739
S2V_AE_VAE_ELM-AE_250	0.1702	0.3074	0.4269	0.5235	0.6040

The ELM-AE is based on 250 hidden units in the latent space. The result obtained with this configuration is exposed in Table 4.6 (**S2V_AE_VAE_ELM-AE_250**). The performance of the proposed ensemble technique is outperformed by this new configuration and it achieves better results than other models. The particularity of this configuration is that the dimensionality of latent spaces is higher than the first ensemble.

Query-based summarization with SKE

In this section, we consider the query-oriented summarization task with the English SKE dataset. The email subject is considered as the query text. Table 4.7 presents the Rouge-2 recall of the BOW approach (*tf-idf* baseline) and two of the proposed approaches using subject-oriented summarization: sentence2Vec and ensemble method combining BOW and sentence2vec with majority voting technique.

Table 4.7 ROUGE-2 recall of Subject-oriented summarization with SKE using Sentence2Vec model and an Ensemble learning of BOW and Sentence2Vec

	Summary size (number of sentences)				
Model	n=1	n=2	n=3	n=4	n=5
BOW	0.0994	0.2038	0.3107	0.4038	0.4867
Sentence2Vec	0.1138	0.2274	0.3373	0.4404	0.5373
Ensemble: BOW_S2V	0.1060	0.2143	0.3190	0.4223	0.5079

It is clear from the exposed results in Table 4.7, that the new approach based on sentence2vec representation outperforms the classical approach based on BOW representation. Moreover, we note that the new ensemble learning model (**BOW_S2V**) performs well compared to BOW model but badly compared to sentense2vec. This leads us to conclude that, in the case where we use the English SKE dataset, the information provided by BOW representation decreases the quality of the summary provided by sentence2vec.

Table 4.8 shows the results obtained by the adopted neural networks models trained on both BOW and Sentence2Vec representation. According to these results, the models based on AE and VAE give the best result when they are trained on Sentence2Vec representation. By analyzing these results, we prove that Sentence2Vec representation is more reliable and contains more information as the traditional BOW representation. Regarding the model based on ELM-AE, we note that the results obtained by BOW are better than the results obtained by Sentence2Vec.

To confirm the strength of our proposed ensemble method, we performed an experiment of subject-oriented summarization with SKE using the same configuration described in the previous section. The results obtained with this configuration are exposed in Table 4.9 (**S2V_AE_VAE_ELM-AE_250**). We show that the performance of the proposed ensemble technique is outperformed by this new configuration and it achieves better results than other models.

Table 4.8 ROUGE-2 recall of Subject-oriented summarization with SKE using unsupervised neural network models trained with BOW vectors and Sentence2Vec vectors.

	BOW (TF-IDF)			Sentence2Vec		
Size	AE	VAE	ELM-AE	AE	VAE	ELM-AE
n=1	0.0642	0.1035	0.1063	0.1093	0.1088	0.0979
n=2	0.1287	0.2013	0.1991	0.2131	0.1999	0.1912
n=3	0.2003	0.3068	0.2986	0.3207	0.3121	0.2859
n=4	0.2664	0.3966	0.3894	0.4183	0.4056	0.3763
n=5	0.3393	0.4868	0.4754	0.5101	0.4904	0.4697

Table 4.9 ROUGE-2 recall of subject-oriented summarization with SKE using Ensemble learning models

Ensemble model	Summary size (number of sentences)				
	n=1	n=2	n=3	n=4	n=5
S2V_AE_VAE_ELM-AE	0.1162	0.2336	0.3459	0.4511	0.5464
BOW_AE_VAE_ELM-AE	0.1030	0.1976	0.3052	0.3996	0.4829
S2V_AE_VAE_ELM-AE_250	0.1185	0.2342	0.3514	0.4503	0.5386

4.3.3.3 Comparison with existing methods

In order to assess the adequacy and efficiency of the approaches proposed in this chapter, we compare their performances with some existing methods. For Arabic EASC dataset, we developed two summarization systems. The first system is TextRank (Mihalcea and Tarau, 2004) which is similar to the baseline graph-based BOW representation investigated in previous section (see table 1). The difference is in the similarity measure between sentences. In TextRank the similarity is measured based on the content overlap between the given sentences, while our baseline uses cosine similarity measure of *TF-IDF* vectors. TextRank is described in detail in section 1.5.6 of chapter 1. The second system is a topic-based summarization system which is based on Latent semantic analysis (LSA) (Mashechkin et al., 2011). LSA is used for dimensionality reduction and for creating a vector representation of a document (or sentence) in a latent space using singular value decomposition (SVD) on the *tf-idf* vectors. In order to perform the summarization task with LSA, we project the matrix obtained by the BOW representation into a latent space. The produced matrix is used to compute the semantic similarity between sentences and the summary is produced by a graph-based model.

To simplify the comparison, we show only the models trained with sentence2vec, since an initial comparison of the proposed models with those trained on BOW is already reported in section 4.3.3.1. The evaluation results shown in Table 4.10 prove that our algorithm outperforms the existing methods when the evaluation task is carried out on the EASC corpus. This result is valid for all the proposed models. Therefore, we can confirm that our proposed models can improve the summarization task giving better results in various cases.

The best result of the competitors is obtained by the graph-based VAE proposed by Alami et al. (2018), which use the VAE as an unsupervised learning model. The Rouge-1 measure obtained by this method is 0.402 when the summary size is 30%, while in our experiences, the best Rouge-1 result is 0.5147 obtained by the ensemble **S2V_AE_VAE_ELM-AE**. Other proposed models outperform the summarization task compared to the competitors, except the model based on sentence2vec and AE (Sentence2Vec_AE), which gives the lower results of the proposed models (0.3515 of Rouge-1). The Rouge-1 of all the proposed ensemble models are better than others. These results clearly indicate that when the information is provided from several sources (different models), the system generates an effective and meaningful summary.

Table 4.10 ROUGE-1 comparison with other methods on EASC corpus using graph-based model and different summary size.

	Summary size			
Method	10%	20%	30%	40%
BOW (TF-IDF)	0.0986	0.2537	0.3693	0.4885
LSA (Topic-based)	0.1045	0.2559	0.3608	0.4312
TextRank	0.1197	0.2819	0.3892	0.5014
Graph-based VAE (Alami et al., 2018)	0.1101	0.2825	0.4021	0.5298
Sentence2Vec	0.1299	0.2924	0.3953	0.4969
Sentence2Vec_AE	0.1024	0.2484	0.3515	0.4652
Sentence2Vec_VAE	0.1117	0.2635	0.3705	0.4746
Sentence2Vec_ELM-AE	0.1120	0.2662	0.3658	0.4746
Ensemble: BOW_S2V	0.3185	0.4305	0.5064	0.5798
Ensemble: BOW_AE_VAE_ELM-AE	0.2812	0.3752	0.4672	0.5579
Ensemble : S2V_AE_VAE_ELM-AE	0.3265	0.4444	0.5147	0.5910

For English dataset, we compare our methods with the results published by (Youssef et al., 2017). To the best of our knowledge, Youssef et al. (2017) is the only work that evaluate the summarization system using SKE dataset. The authors proposed a new summarization method using an unsupervised deep learning method based on auto-encoder. They introduced an Ensemble Noisy Auto-Encoder (ENAE) in which the summarization is produced by a same model and a same input, but with different added noise. The final summary is then generated using a majority voting technique. They compared their results with unsupervised and supervised models reported for BC3 dataset (Ulrich, Murray, and Carenini, 2008)

For unsupervised models, they found that their approach exceeds the best unsupervised systems existing in the stat of the art which are: a graph-based model (Hatori et al., 2011), MEAD (Radev et al., 2004) and ClueWordSummerizer (Ulrich et al., 2009). For supervised models, Ltf-ENAE (Gaussian) outperforms the supervised methods based on SVM, ME (lex) and BAG (lex-lc). Furthermore, the best supervised techniques reported in Ulrich et al. (2009), which are Bagging and Gaussian process perform better than Ltf-ENAE (Gaussian) model.

In this chapter, we compare our best model with that propped by Youssef et al. (2017) (Ltf-ENAE (Gaussian)). Table 4.11 shows that all the proposed models based on sentence2vec representation outperforms the state-of-art method. The best performances are achieved by the ensemble method combining sentence2vec and unsupervised neural networks. On the other hand, the bad results are obtained by the ensemble of neural models based on BOW representation. This shows that the BOW approach decreases the performances of the summarization system. However, Word2Vec approach increases the performances of the summarization system, especially when it is used as the input of the ensemble of unsupervised neural network models composed of AE, VAE and ELM-AE.

Table 4.11 ROUGE-2 comparison between the proposed methods and others on SKE dataset using graph-based model and different summary size.

	Summary size (number of sentences)				
Method	n=1	n=2	n=3	n=4	n=5
Ltf-ENAE (Youssef et al., 2017)	0.1370	0.2471	0.3510	0.4325	0.5031
Sentence2Vec	0.1646	0.3012	0.4214	0.5136	0.5902
Ensemble: BOW_S2V	0.1556	0.2861	0.4083	0.4969	0.5771
Sentence2Vec_VAE	0.1334	0.2591	0.3623	0.4575	0.5387
Sentence2Vec_AE	0.1492	0.2812	0.4045	0.5122	0.5957
Sentence2Vec_ELM-AE	0.1441	0.2671	0.3725	0.4677	0.5453
Ensemble S2V_AE_VAE_ELM-AE	0.1637	0.2931	0.4169	0.5097	0.5909
Ensemble BOW_AE_VAE_ELM-AE	0.1061	0.2064	0.3133	0.3989	0.4739
Ensemble S2V_AE_VAE_ELM-AE_250	0.1702	0.3074	0.4269	0.5235	0.6040

By this work, and based on the experimental outcomes, we can confirm that using unsupervised neural networks and word embedding contribute to the improvement of automatic summarization task, especially when they are combined in an ensemble technique. The proposed approach is able to generate summaries that are close to what the human produces, by ranking and selecting the most important sentences that express various ideas conveyed by the original text. Several reasons are behind this improvement. First, the proposed models can express the implicit semantic relations by building a low-dimensional concept space, where the semantic relationships between different textual units are identified. Second, they automatically learn high-level features from data by unsupervised feature learning instead of using feature extraction tools or domain expertise. Third, the proposed approach deal with a shortage in manual annotated data (texts with their summaries produced by human experts), which are required to create powerful systems based on supervised deep learning algorithms. In the context of automatic text summarization, labelled data are few and very hard to obtain, while unlabeled data are widely available for learning meaningful representations.

4.4 Conclusion

In this chapter, we have introduced several unsupervised learning algorithms based on neural networks for automatic text summarization. These algorithms are performed based on vector representation of words. In recent years, the increased strength of deep learning methods especially for leaning unsupervised tasks makes this representation more powerful and more relevant than the classical BOW representation through word2vec. On the other hand, ensemble learnings technique usually produces more accurate results than a single model. We have proposed several models in order to address the summarization task. In order to train these models, we have used two type of vector representations built from two kind of approaches: BOW and Word2vec approach. The goal is to demonstrate the benefits of the information provided by word2vec and the proposed models trained on sentence2vec vectors. The summarization task is significantly improved with the combination of these models through an Ensemble method with a voting technique. For unsupervised learning models, we have used the AE, VAE and ELM-AE in order to learn the latent semantic representation of documents.

We have experimented with a two kind of dataset designed to evaluated the summarization task in English and Arabic languages. The results confirm that sentence2vec encapsulates relevant information and achieve better result than representation based on BOW approach. Also, we show that Ensemble method based on unsupervised neural network models trained on Sentence2Vec representation outperforms significantly the performances of the summarization task and obtains the best accuracy for both English and Arabic datasets.

Chapter 5

Enhancing Arabic text summarization by document clustering and topic modeling

5.1 Introduction

Most existing Arabic text summarization methods do not consider the context or domain to which the document belongs. We assume that the summarization system is more efficient if it is able to detect the context of the text to be summarized. For example, an effective summarizer for biomedical text should be able to identify and extract important biomedical concepts.

Topic is the subject of the document, i.e., what the document is about. The topic space represents a set of identified topics in the given corpus. Topic modeling is a type of statistical modeling for discovering the abstract topics that occur in a collection of documents. It provides us with methods to organize, understand and summarize large collections of textual information. There are several methods allowing the identification of topics existing a dataset. Latent Dirichlet Allocation (LDA) is an example of topic model and it is used to classify text in a document to a particular topic. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions.

Document clustering is the process of document dataset grouping that refers to the similarity of document data patterns into a cluster. Meanwhile those document without similarity will be grouped into another clusters. K-means is one of the well-known cluster algorithm and frequently used to resolve clustering problem by grouping a certain number of k cluster, where the number k has been defined previously.

One of the most important steps in text summarization is document representation. For further processing, text needs to be converted into numerical values. This conversion consists of building a set of vectors representing each document. For that, traditional Arabic summarization system use the term frequency (TF), inverse document frequency (IDF) or term frequency-inverse document frequency (TF.IDF) feature. A summarizer is said to be more relevant, if it contains more fruitful and relevant compact representation of large text collections. More powerful document representation approaches have been advanced. Recently, Word embedding and neural networks are the most widely used to improve the quality of several applications. In addition, topic modeling is a probabilistic approach allowing a representation of a document in a topic space according to themes and subjects circulated in the dataset. We assume that combining both of them using ensemble learning techniques can improve significantly the performance of Arabic summarization task. We do not need any domain-specific knowledge, but we will try to automatically learn the context of each document to be summarized.

In this chapter, we propose new Arabic summarization approaches based on topic modeling and unsupervised neural networks namely Auto-Encode (AE), Variationnal Auto-Encode (VAE) and Extreme Learning Machine Auto_Encode (ELM-AE). In addition, we have adopted several ensemble learning methods combining different models. The neural networks and ensemble models proposed in this work are used to learn the unsupervised features and extract latent concepts in the abstract concept space from real-valued input data. Furthermore, instead of using the traditional document representation matrix, we investigated the relevance of document representation in the topic space on the Arabic text summarization task. In our proposed methods, we split our dataset into different clusters, and for each cluster we identify the topic terms that represent the subject of the documents collection. The document vectors are generated based on the relationships between document sentences and a set of identified topics. Finally the document matrix is formed by concatenating the generated document vectors.

In order to rank sentences, we have adopted a graph-based summarization technique and two sentence selection strategies are investigated. The first one is to simply select the top-ranked sentences to form the summary. The second selection technique consists of using an adapted Maximal Marginal Relevance (MMR) algorithm (Carbonell and Goldstein, 1998) is applied to eliminate unneeded information and enhance the quality of the final summary. Indeed, information redundancy and diversity is a typical problem in automatic text summarization and especially for Arabic documents.

In the first stage of the proposed approaches, we proceed to document preprocessing, which consists of splitting, normalizing and removing stop-words. The second stage consists of clustering the dataset. The purpose of this stage is to group similar documents in the same cluster. The third stage consists of identifying the topic of each cluster. Thus, LDA is applied on each cluster to build the cluster topics and terms belonging on that cluster. The fourth stage is the document analysis, which consists of preprocessing (removing stop-words, normalization, stemming…) and creating a matrix representation of each document in the topic space. The importance of each sentence with respect to the topic terms is computed and represented as the input matrix to the proposed summarizers. Next, several unsupervised deep learning and ensemble learning models are used to learn unsupervised features by training the input matrix built from the sentence/topics representation. The learned features are used to compute the semantic similarity between sentences. The similarity matrix is then used as the input of the graph model to rank each sentences. Subsequently, the weighted ranking algorithm PageRank (Brin and Page, 1998) is executed on the graph to produce relevant score for each sentence in the input text. Then, the summary is built from the higher ranked sentences while respecting the desired summary length. Finally, the final summary document is generated by identifying and removing duplicate sentences which are similar to each other.

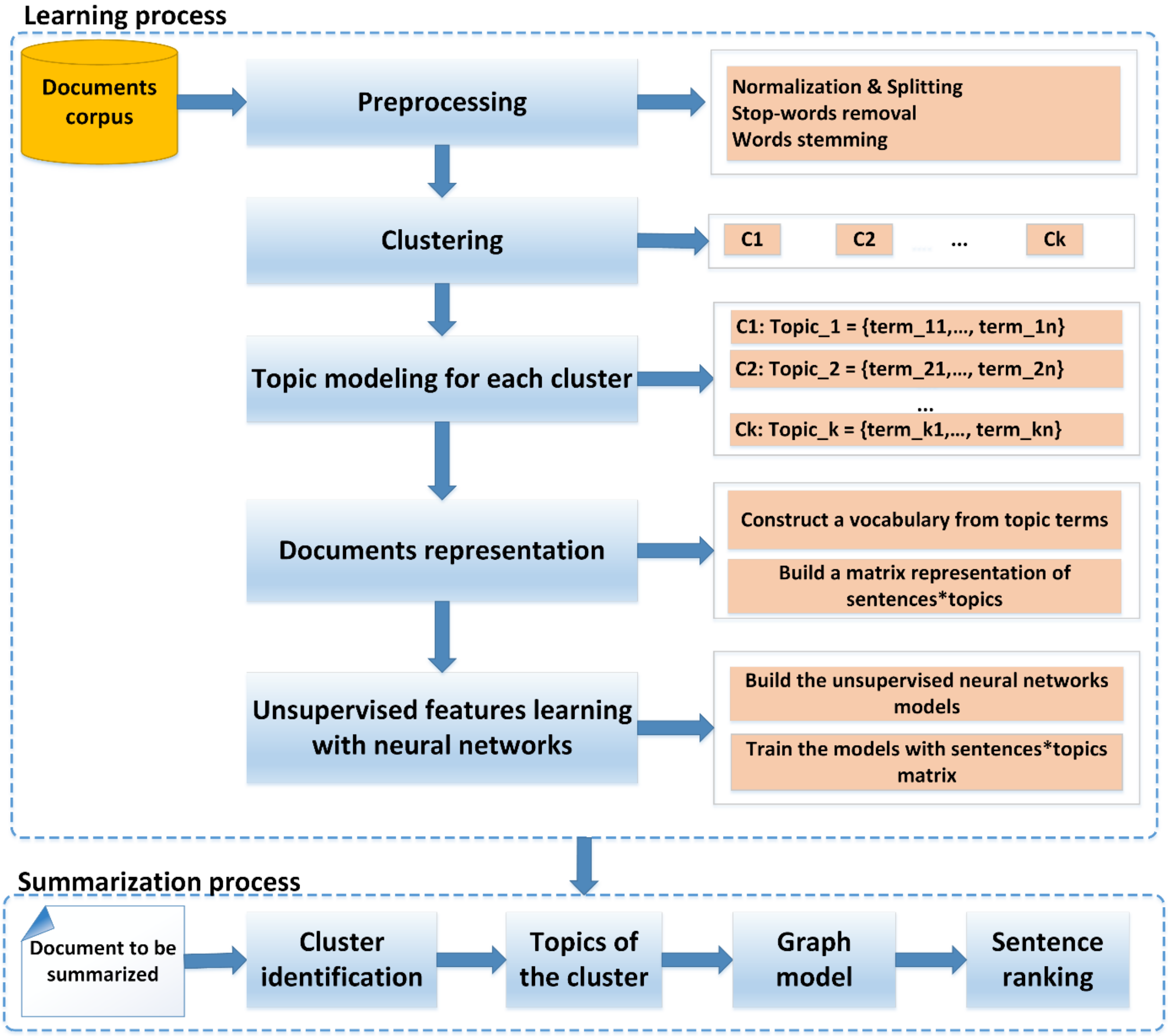

Learning process
Documents corpus
Preprocessing
Normalization & Splitting
Stop-words removal
Words stemming
Clustering
C1
C2
...
Ck
Topic modeling for each cluster
C1: Topic_1 = {term_11,..., term_1n}
C2: Topic_2 = {term_21,..., term_2n}
...
Ck: Topic_k = {term_k1,..., term_kn}
Documents representation
Construct a vocabulary from topic terms
Build a matrix representation of sentences*topics
Unsupervised features learning with neural networks
Build the unsupervised neural networks models
Train the models with sentences*topics matrix
Summarization process
Document to be summarized
Cluster identification
Topics of the cluster
Graph model
Sentence ranking

Figure 5.1 The main steps of the proposed Arabic Summarization system

5.2 Proposed Arabic text summarization systems

5.2.1 General architecture

Figure 5.1 illustrates the general architecture of our proposed approach. The input of the system is an Arabic corpus with a large number of text documents. The proposed system is composed of two key processes: learning process and summarization process. The learning process consists of dividing the large corpus into several groups, which are similar according to the subjects covered. The purpose is to learn features from documents that belong to the same topics in order to build an accurate representation of the document to be summarized. This process is performed in five major stages. The first stage is document preprocessing where each document is prepared for the next stage by removing stop words, splitting, normalization and stemming the input text. The second stage is document clustering where the clustering technique is applied on the documents collection to create several documents clusters. The purpose of this stage is to group similar documents for making them ready for summarization and ensures that all the similar set of documents participates as a group in the summarization process. In the third stage, topic modeling with Latent Dirichlet Allocation (LDA) technique (Blei et al., 2003) is applied on each cluster of text documents in order to generate the cluster topics and terms belonging to each cluster. The fourth stage is the document vectors formation. A matrix representation of each document is formed based on the topic space of the specific document cluster. Each sentence in the document is represented by a row and each column represents a topic belong to the document cluster. The topic terms generated for text clusters are taken as the column matrix. In the fifth stage, unsupervised deep learning algorithm is applied on the resulted matrix in order to learn the abstract features of the original representation. In the second process (summarization process), the summarization of any text document is performed by identifying the cluster in which the document belongs, extracting the topics of the cluster and ranking the document sentences. In this work, we investigated the graph-based summarization approach based on the PageRank algorithm. The latent matrix representation in the concept space built by the unsupervised neural network models is used as the input of the ranking process. In the following section, we explain in the detail each phase of the two processes.

5.2.2 Preprocessing phase

The preprocessing phase consists of cleaning the source documents, as well as splitting and tokenizing the sentences. In our system, the sentence is the extraction unit and the term is considered as a scoring unit. We implement this phase in three steps:

5.2.2.1 Tokenization

The Tokenization process consists of dividing the text into tokens. The input text is normalized through two steps: first, all punctuations, non-letters and diacritics are removed, secondly some characters are replaced by the normalized ones (أ, آ, and إ with ا and last ى with ي and last ة with ه). In our system, and depending on the datasets used, we consider the character "." as a sentences separator and the character " " (space) as a word separator. This consideration makes the splitting process easy in order to segment the text document into sentences and each sentence into words.

5.2.2.2 Stop words removal

Stop-words are very common words with a mainly structural function; they are recurrently used in a text, carry little meaning and their function is syntactic only. They do not indicate the subject matter and do not add any value to the content of their documents. In Arabic, words like (هو, هذا, الذي, هي) are frequent in sentences; but with little significance in the implication of a document. These words can be deleted from the text to help identify the most meaningful words in the summarization process. There is no typical list of stop-words specific to the Arabic text; this is why we simply use, in this work, a list of 168 words proposed by (Khoja, 2001).

5.2.2.3 Root extraction

Words in Arabic are generally derived from a root, which is indeed a base for diverse words with a somehow related meaning. A set of derivations representing a same area can be constructed by adding suffixes to the root. Identifying a root of an Arabic word (stemming) helps in its grammatical variations mapping to the instances of the same term. This amounts to saying that multi-derivations of the Arabic wording structures make the semantic representation of the text possible. The quality and performances of a text summarization task may be positively impacted by an adequate representation of Arabic text. Moreover, since words sharing the same root have a semantic relation, using this root in features selection can improve the accuracy of similarity measure and frequency analysis in Arabic text because the words in a text can have more than one occurrence, but in different forms.

5.2.3 Clustering phase

Clustering is a process of creating groups of similar objects. Data, which consists in our case a large documents collection, are portioned in an unsupervised manner such that documents that are similar to each other are grouped in the same group. Document clustering consists of grouping similar documents in the same cluster or group. In this work, we have adopted the k-mean clustering algorithm which uses statistical features like *TF.IDF*, cosine similarity, etc. to create k number of clusters from k disjoint partitions. It is a simple and a very popular unsupervised clustering technique.

Though it suffers from the initial selection bias, k-means is a prominent unsupervised learning algorithm used to cluster unlabeled data, part of it is attributed to its convenient implementation and to its efficiency. Our clustering method is detailed in Algorithm 5.1.

Algorithm 5.1 Our proposed clustering algorithm

```
Input: k: the number of clusters, D: a dataset containing n documents.
Output: A set of k clusters
Unsupervised feature extraction:
1) From the input dataset, build a TF.IDF matrix of a vocabulary of 10000 most frequency words
   in the dataset
2) Build an unsupervised ELM model (ELM-AE) with 10000 units in the input layer and 200 units
   in the hidden layer
3) Train the ELM-AE using the TF.IDF matrix built in the first step. The output of the ELM-AE
   is a new matrix representing the abstract features learned by the ELM-AE model. We note C
   the new matrix representation of the input data in the new concept space. Now, each document
   is represented by a 200-dimensional vector in the concept space.
k-mean clustering:
4) Arbitrary choose k objects from C as the initial cluster centers;
Repeat:
5) Assign each object to the cluster to which the object is the most similar, based on the mean value
   of the objects in the cluster;
6) Update the cluster means, i.e., calculate the mean value of the object for each cluster;
7) Until no change;
```

As illustrated in Figure 5.2 the algorithm operates in four major stages:

- Preprocessing stage: is performed in order to clean the source documents, as well as to split and tokenize the sentences.
- Features extraction: generally, the most used features for clustering are *TF.IDF*.
- Unsupervised features learning: this phase aims to build a new representation of the original data that is more suitable to accomplish some specific tasks such as clustering. In this work, the clustering task is performed by the ELM-AE algorithm (Kasun et al., 2013), which is an unsupervised feature learning model that builds a low dimensional concept space to represent abstract features from our initial unlabeled data. ELM-AE learns the abstract

representation of the input data, which consists of the *TF.IDF* matrix representation of the corpus. These features could be seen as latent features as they are a distributional representation of our sparse data in a low dimensional space thus are more efficient than *TF.IDF*.

- K-means clustering: this phase consists of creating k number of clusters from k disjoint partitions. The algorithm is performed on the latent matrix representation generated from the previous stage. The output is a set of k cluster with their appropriate text documents. The k-means algorithm is performed in three steps:
  - Initialization: First, k random samples are selected from the datasets as centers of the clusters denoted the centroids,
  - Assignment: The prevailing samples are assigned to the most proximate centroid using equation (5.1). In this step, we use the Euclidean distance denoted L_2 to assign our samples to each cluster. Each data point *x* is assigned to a cluster c_i based on:

$$argmin_{c_i \in C} dist(c_i, x)^2 \tag{5.1}$$

where $dist(.)$ is the Euclidean distance denoted (L_2). C is a set of initial clusters.

  - Update: Finally, the new centroid of each cluster is determined. The mean is computed taking into account all the samples belonging to a specific cluster in order to re-compute the centroids according to the following formula:

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i \tag{5.2}$$

Where S_i is a set of data point assigned to the cluster i.

This process is repeated until the convergence of our cost function defined by the equation (5.3) given that *E* is the square error sum, p a sample and mi the mean of the cluster *Ci*.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 \tag{5.3}$$

In general, there is no method for determining the exact value of k *(number of clusters)*, but an accurate estimate can be obtained using many techniques. One of these techniques consists of using the mean distance between data points and their cluster centroid parameterized by *k* in order to compare the results across different values of *k*. The optimal value of *k* is the elbow point of our plot characterized by a sudden decrease in the y-axis. Another technique for choosing *k*, consists of using cross-validation by splitting the training data into a train dataset and validation dataset or by using 10 folds technique. We could also use other techniques for hyperparameters optimization such as grid search, random search or Bayesian optimization.

In this work, to find the number of clusters in the dataset used in the experimentation, we have run the *K*-means clustering algorithm for a range of multiple values and compare the results obtained for each value. The value of *k* has been chosen so that our proposed approaches give the better performance.

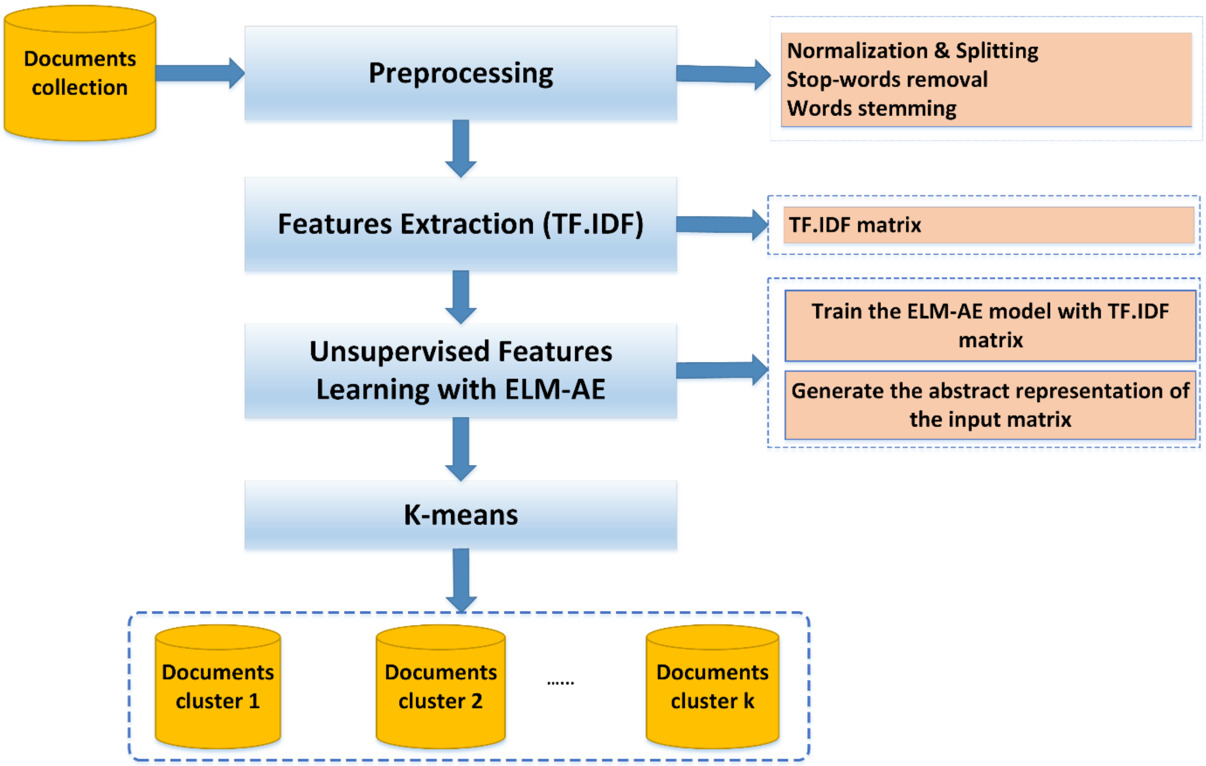

Documents collection
Preprocessing
Normalization & Splitting
Stop-words removal
Words stemming
Features Extraction (TF.IDF)
TF.IDF matrix
Unsupervised Features Learning with ELM-AE
Train the ELM-AE model with TF.IDF matrix
Generate the abstract representation of the input matrix
K-means
Documents cluster 1
Documents cluster 2
......
Documents cluster k

Figure 5.2 Architecture of the clustering algorithm

5.2.4 Topic modeling for document representation

Document representation is an important step in any NLP application. In order to perform any machine learning algorithm or statistical technique on any form of text, documents need to be converted into numerical values or vector representation for further automatic processing. This numeric representation should depict significant characteristics of the text. There are many such techniques, such as, term-frequency, Term Frequency-Inverse Document Frequency (TF-IDF), word co-occurrence matrix, word2vec and GloVe. TF-IDF (Term Frequency-Inverse Document Frequency) is a very common algorithm to transform text into a meaningful representation of numbers. This technique is widely used to extract features across various NLP applications especially for Arabic documents.

Topic modeling is a type of statistical modeling for discovering the abstract topics that occur in a collection of documents. The topic space is related to a set of topics composed of the most important terms describing the dataset. Latent Dirichlet Allocation (LDA) is an example of topic model and it is used to classify text in a document to a particular topic. LDA was proposed by Blei et al. (2003) for document representation. It builds a topic per document model and words per topic model, modeled as Dirichlet distributions. Blei et al. (2003) applied the LDA technique in the evaluation of document model and they showed that LDA outperformed other latent topic models especially the probabilistic LSA (Chien and Wu, 2008). LDA was also used to construct the LDA language model for speech recognition (Chien and Chueh, 2008). In this work, we have adopted the LDA in order to improve the Arabic text summarization task.

Assume we have a dataset of M documents with a total of N words, V vocabulary and T latent topics. For each document d, each word w in d is associated with a hidden variable z, which represents the latent topic. The variable z is sampled from a multinomial distribution with parameter θ indicating the probability of latent topic. The prior density of multinomial parameter θ is given by a Dirichlet distribution with hyperparameter α. The topic language model is designed by the $T \times V$ parameter matrix $\beta = \{\beta_{tw}\}$. To estimate the LDA parameters $\{\alpha, \beta\}$, a marginal likelihood $p(w|\alpha, \beta)$ is maximized from a set of text documents $w = \{w_{dn}\}$.

Figure 5.3 illustrates the probabilistic graphical model of LDA. The LDA has different parameters:

- A word w belonging to a fixed vocabulary with size V. The word w can be considered as a vector which all components are null except for the component in the position i, which is the index of w in the vocabulary V.
- A document d with N words, $d = \{w_1, \dots, w_N\}$
- A corpus which is a collection of D documents, $D = \{d_1, \dots, d_D\}$
- Variable $z_{d,n}$ which indicates the probability to assign the latent topic t to the word $w_{d,n}$ (n^{th} word in document d)
- The parameter θ_d which represents the topic distribution for the document d
- β_t is the parameter of Dirichlet distribution of topic t
- α is the parameter of Dirichlet prior the per-document topic distribution.

As the figure makes clear, there are three levels to the LDA representation. The parameters α and β are corpus-level parameters, assumed to be sampled once in the process of generating a corpus. The variables θ are document-level variables, sampled once per document. Finally, the variables z and w are word-level variables and are sampled once for each word in each document.

The only observed data in the LDA model are words $w_{d,n}$. The other variables (θ and $z_{d,n}$) are hidden, and need to be learned. Given the parameters α and β, the role of the inference model is to determine the hidden variables θ and $z_n$of a document d, given the list of words w_n in the document. The main inference methods for LDA are sampling methods (notably Gibbs sampling) and variational methods (particularly mean-field methods, which can be done in batch or online). The EM algorithm is used by the inference model in order to estimate the parameters α and β.

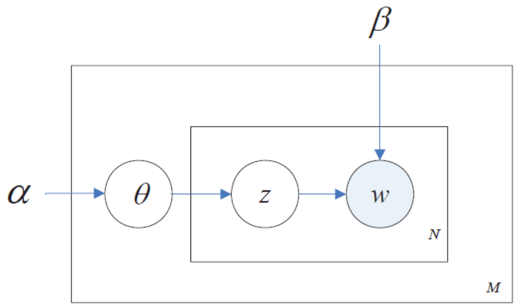


Figure 5.3 Graphical model for LDA

The LDA algorithm is divided into two major phases: initialization and training. In the initialization phase, a topic is assigned to each word in each document, according to a Dirichlet distribution on a set of T topics. The generative process of LDA for a document d can be expressed as follow:

- Choose $\theta \sim Dirichlet(\alpha)$
- For each word $w_n$do:
    - Choose a topic $z_n \sim Multimodal(\theta)$
    - Choose a topic $w_n \sim Multimodal(\beta_t)$, with $t = z_n$

This generates a first topic-model, which consists of subjects present in the documents and the words defining those subjects. This topic-model is very unlikely because it is randomly generated.

In the training phase, we seek to improve the topic-model randomly generated in the initialization phase. For this, the topic of each word in each document in the corpus is updated. The training phase tries to improve the topic-model randomly generated in the initialization phase. For this, in each document, we take each word and update the theme to which it is linked. This new theme is the one that would have the highest probability of generating it in this document. It is therefore assumed that all themes are correct except for the word in question. More precisely: for each word (w) of each document (d), two things are calculated for each topic (t):

- $P(t \mid d)$: the probability that the document d is assigned to the topic t.
- $P(w \mid t)$: the probability that the topic t in the corpus is assigned to the word w.

We then choose the new topic t with the probability $P(t \mid d) * P(w|t)$. This corresponds to the probability that the topic t generates the word w in the document d.

By repeating the previous steps a large number of times, the assignments stabilize. The topics present in each document is obtained by counting each representation of a topic (assigned to the words of the document). The words associated with each topic are obtained by counting the words associated with them in the corpus.

After training the LDA model, a latent topic space is built. The number of topics is much lower than the number words in the vocabulary. The content of each document has a semantic relationship with the extracted topics. For this, sentences can be expressed based on the extracted topics. In the learned topic space given by the LDA model, words, sentences and documents can be expressed as a uniform expression. For a word w_i, we can express it as a vector in the topic space, in which the value for each topic is the probability of the topic, given w_i. That is $L(w_i) = (P(z_1|w_i), P(z_2|w_i), \dots, P(z_t|w_i))$. According to the Bayes formula:

$$P(z_i|w_i) = P(z_i)P(w_i|z_i)/P(w_i) \quad (5.4)$$

We can get $P(w|z_i)$ from parameter β. $P(w_i)$ can be calculated through a simple statistic processes.

$P(w_i) = count(w_i)/N$, where N is the number of words in the vocabulary.

Therefore, the probability of the topic, given w_i, can be calculated. We can express a word as a vector of topics. We can get the topic vector of a sentence $S = \{w_1, w_2, \dots, w_n\}$ by calculating the average of the topic vectors of all words in S using the following formula:

$$L(S) = L(w_1, w_2, \dots, w_n) = \left(\frac{\sum_{i=1}^{n} P(z_1|w_i)}{n}, \frac{\sum_{i=1}^{n} P(z_2|w_i)}{n}, \dots, \frac{\sum_{i=1}^{n} P(z_t|w_i)}{n}\right) \quad (5.5)$$

Where $L(S)$ indicates the latent topic vector of the sentence S.

Then the cosine distance is used to measure the similarity of any two sentences. The similarity of two sentences is defined as:

$$sim(S_1, S_2) = COSINE(L(S_1), L(S_2)) \quad (5.6)$$

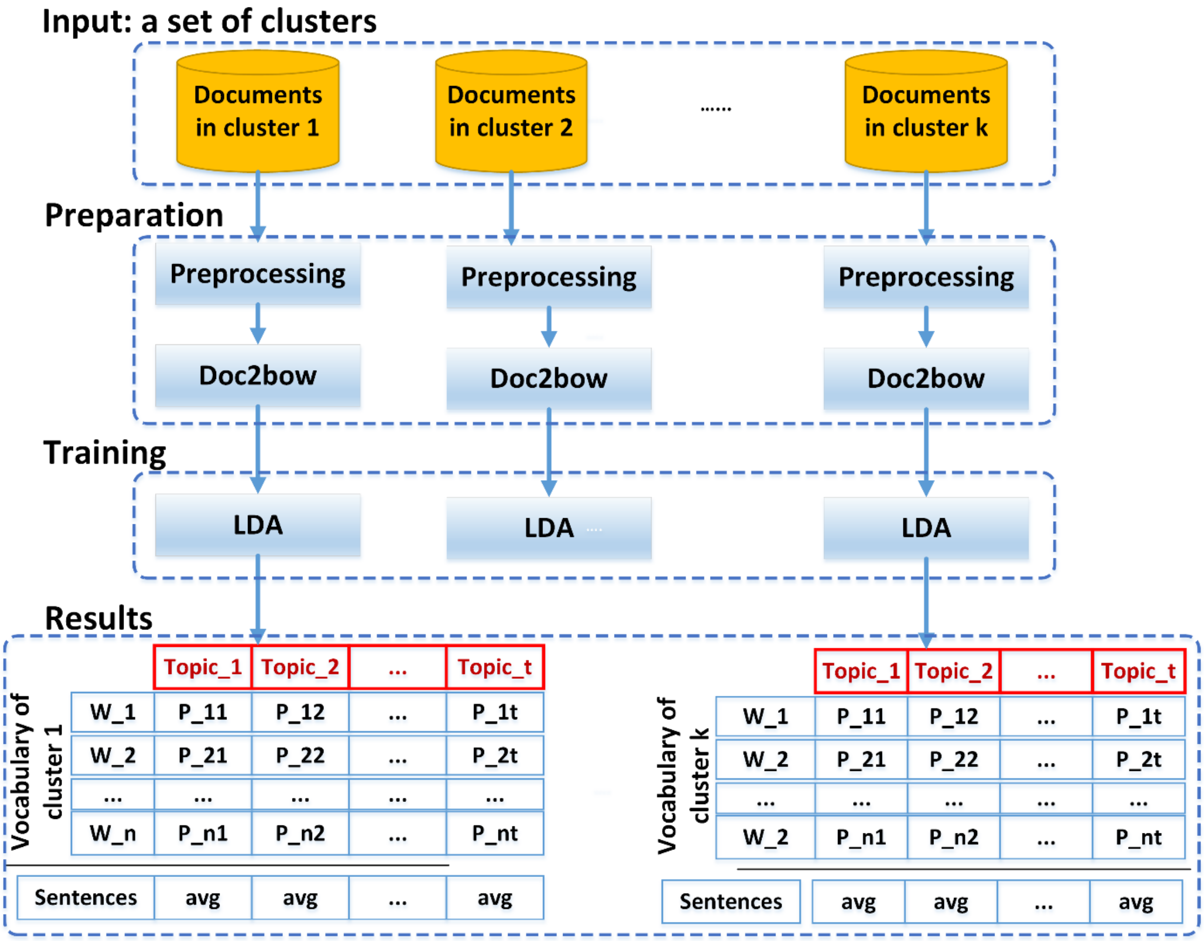


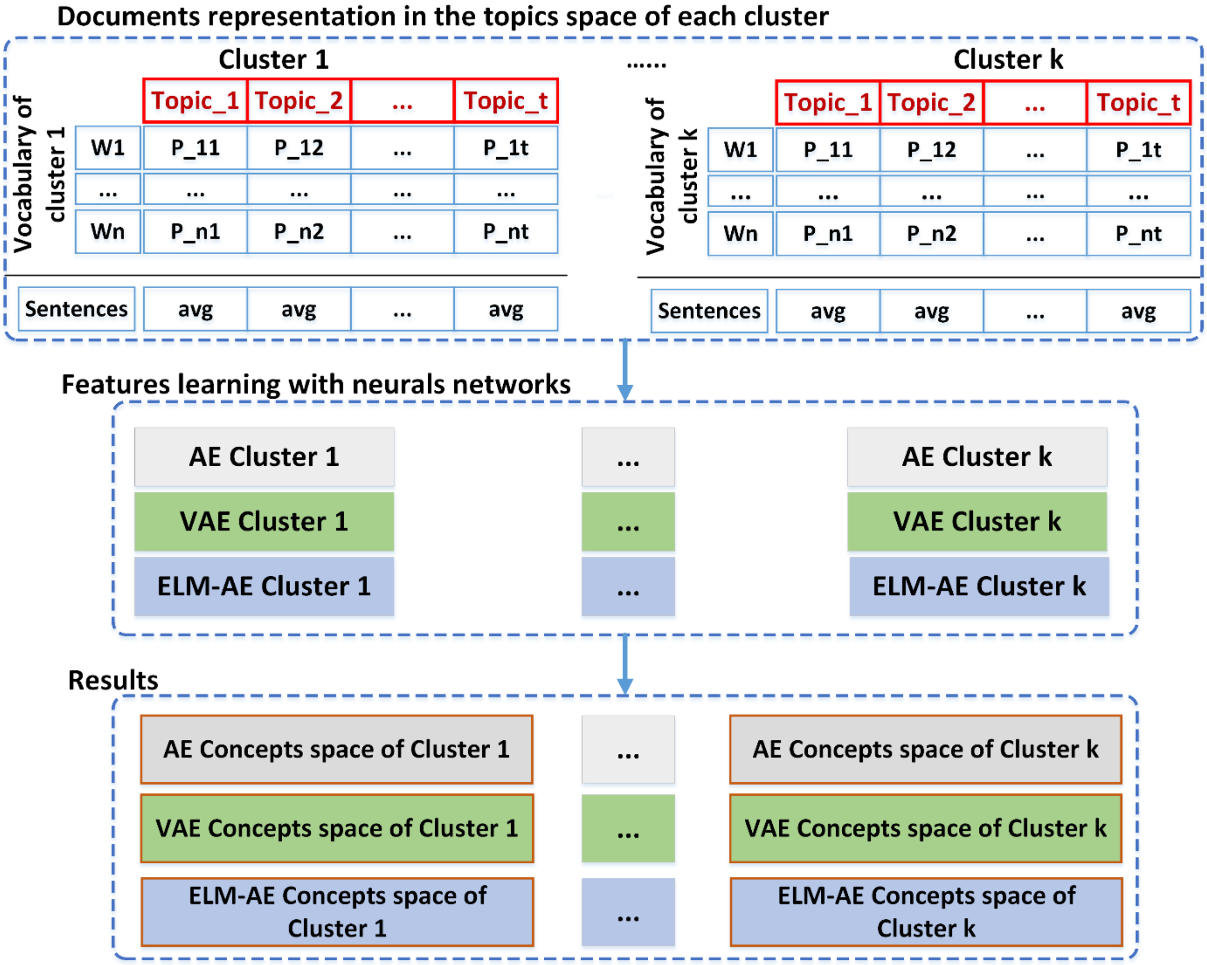
Figure 5.4 Topic modeling and document representation

Figure 5.4 Topic modeling and document representationillustrates the main steps to generate document representation for each cluster. Each set of text documents in each cluster identified in the previous stage is preprocessed and transformed into Doc2bow representation, which is used as the input of the LDA algorithm in order to generate topics for each cluster and represent each word and sentence according to these topics.

After training the LDA model on each documents cluster, a latent topic layer is built for each cluster. The number of topics is much lower than that of words. The topics here have deep relationship with the content of the document. So it is fit for being the foundation of the sentence expression. In the topic space, we can express the word, sentence or document as a uniform expression. For example, equation (5.5) is used to express the sentence as a vector in the topic space.

5.2.5 Unsupervised feature learning

Figure 5.5 shows the main steps of this stage. Each representation built from the previous stage is used as the input of several unsupervised neural networks models, namely Auto-Encoder (AE), Variational Auto-Encoder (VAE) and Extreme Learning Machine Auto-Encoder (ELM-AE). After training these models, a concept space of each cluster/model is constructed representing the abstract representation of documents belonging to that cluster.



Documents representation in the topics space of each cluster
Cluster 1
......
Cluster k
Vocabulary of cluster 1
Topic_1
Topic_2
...
Topic_t
W1
P_11
P_12
P_1t
Wn
P_n1
P_n2
P_nt
Sentences
avg
Vocabulary of cluster k
Features learning with neurals networks
AE Cluster 1
VAE Cluster 1
ELM-AE Cluster 1
AE Cluster k
VAE Cluster k
ELM-AE Cluster k
Results
AE Concepts space of Cluster 1
VAE Concepts space of Cluster 1
ELM-AE Concepts space of Cluster 1
AE Concepts space of Cluster k
VAE Concepts space of Cluster k
ELM-AE Concepts space of Cluster k

Figure 5.5 Features learning of each cluster using neural networks models

5.2.6 Summarization process

Figure 5.6 shows how the summarization task is performed by the proposed system. First, for each document to be summarized, we identify it cluster C_i based on the previous stages (training process). After that, based on the trained LDA built from the training process, we identify the main topics related to the cluster C_i. Thus the topic space of the cluster C_i is constructed. Next, we build a matrix representation, which is a document representation model according to this topic space using equation (5.5). Then, the produced matrix is projected into the concept space of each neural networks model (section 5.2.5) in order to build a latent representation of the input document, which is a smaller representation that is more efficient and contains more accurate semantic information. Finally, we use a graph-based summarization technique with a redundancy elimination algorithm in order to generate an efficient and consistency summary. In the following section, we explain in details this summarization technique.

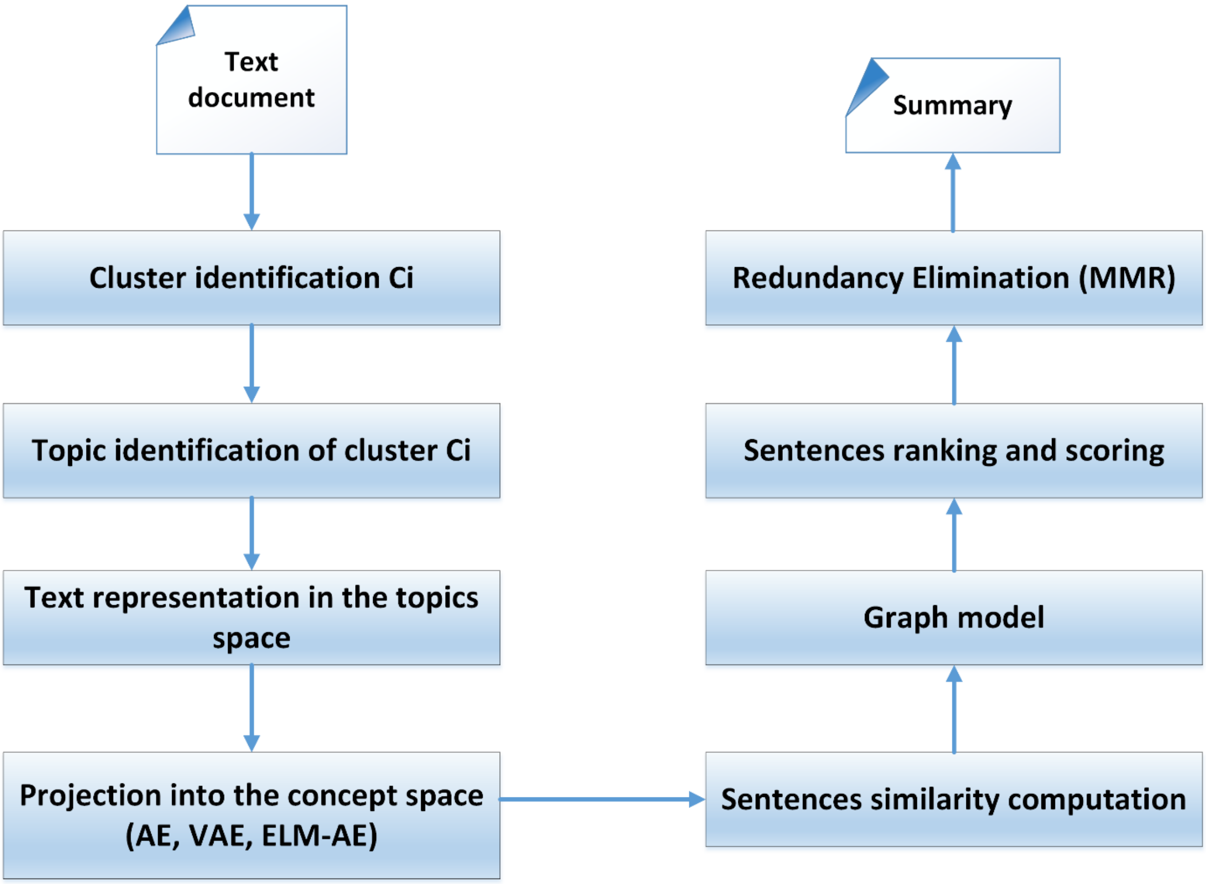

Text document
Cluster identification Ci
Topic identification of cluster Ci
Text representation in the topics space
Projection into the concept space (AE, VAE, ELM-AE)
Sentences similarity computation
Graph model
Sentences ranking and scoring
Redundancy Elimination (MMR)
Summary

Figure 5.6 The summarization process

5.2.7 Sentence ranking

We build our summary based on the most relevant sentences in the document. This is known as extractive summarization or sentence ranking. Our proposed models rank the sentences based on their abstract representation in the concept space learned by the neural network. We investigate the graph-based summarization technique. In graph-based summarization method, we convert the input text document into a graph format. To draw the graph, we need to find textual units that best describe the task of automatic summarization and consider them as nodes of the graph. Then, we need to identify relations that connect those units. In this work, we consider the sentences of the input Arabic document as a text unit and the similarity between those sentences as a relation between them.

The system we have put forward relies on the graph model. An undirected weighted graph *G = (N, E)* is built in which sentences are represented by a set of nodes *N* and the relation between each sentence is represented by the edge that connects the two correspondent vertices. The edge between the pair of sentence is created if this measure exceeds a predefined threshold. The weight of the edge represents the degree of the semantic similarity between the two sentences.

In our graph-based summarization system, the document to be summarized is split into a set of sentences $S = \{s_1, s_2, \dots, s_m\}$. Each sentence S_i is represented by a Node N_i in the graph. The semantic similarity measure between two sentences S_i and S_j is represented by the weight w_{ij} of the edge between node N_i and node N_j.

After training our models, each of them provides a specific mapping function that project each sentence s_i into a concept space in order to provide its latent representation in a low-dimensional space. This new representation is used to calculate the semantic similarity between two sentences according to the equation (5.7).

$$w_{ij} = sim(S_i, S_j) = \frac{\widehat{S_i}.\widehat{S_j}}{\|\widehat{S_i}\|\|\widehat{S_j}\|} \tag{5.7}$$

Where S_i and S_j are the given sentences. $\widehat{S_\imath}$ and $\widehat{S_j}$are their mapping into the concept space of a specific model.

PageRank algorithm (Brin and Page, 1998) was used to calculate a salient score for each vertex of the graph. PageRank is a very popular link analysis algorithm that was developed as a method for Web link analysis. It determines the importance of a vertex within a directed graph, on the basis of the information elicited from the graph structure. In our case, the key intuition is that a sentence should be highly ranked if it is recommended by many other highly ranked sentences. PageRank can as well be used on undirected graph. In this respect, the output-degree and the input-degree for a node are equal. In our case, *In(Ni)* is equal to *Out(Ni)* since the graph is undirected. Equation (5.8) provides the score of a node *Ni*, where *adj (Ni)* is the set of vertices adjacent to *Ni*, w_{ij} is the weight of the edge between node *Ni* and node *Nj*, and *d* is a damping factor that can be set between *0* and *1*. The factor *d* has the role of incorporating into the model the probability of moving randomly from a given node to another in the graph. This factor is often set to *0.85* (Mihalcea and Tarau, 2004).

$$PR(N_i) = (1 - d) + \ d \ * \sum_{N_j \in adj(N_i)} w_{ij} \frac{PR(N_j)}{\sum_{N_k \in adj(N_j)} w_{jk}} \tag{5.8}$$

We apply equation (5.8) iteratively on a weighed graph *G* to compute *PR*. First, all nodes are assigned an initial score of *1* and then equation (5.8) is applied to bring the scores difference between iterations below a threshold of *0.001* for all vertices. The salient score of each sentence S_i corresponds to the weight of its corresponding vertex N_i referred by $PR(N_i)$. When they correspond to vertices with higher scores, these sentences become important, salient to the document and have strong ties with others sentences.

5.2.8 Redundancy elimination and summary generation

Summary generation is the final step of our system. It consists of eliminating redundancy from the best scored sentences obtained by the equation (5.8). In this way, we are sure that our final generated summary covers a diversity of most information contained in the original input document. In this step, and after carrying out the ranking process, each sentence has its salient score $score(S_i)$. Simply and as other graph-based summarization systems, we can choose to include in the final summary (depending on the summary size) the sentences with the higher scores. However, this will create redundancy in the summary, since many similar sentences that represent the same meaning in the document have similar score, so they can be included together in the summary. Also, the remaining ideas of the document may not be identified and relevant information of the document may be overlooked and does not appear in the final summary. That is why the adapted version of MMR is used to re-rank and select appropriate sentences to include into the summary without redundancy. MMR is an iterative method for content selection. In the case of automatic summarization task, it iteratively chooses the best sentence to insert in the summary according to two characteristics:

- Relevant: That is, the sentence must be highly relevant to the content of the text. So, the sentence with the higher ranking score will be considered.
- Novel: which means that the sentence must be minimally redundant with the summary, so the similarity between the sentence and other previously selected sentences in the summary needs to be low.

Algorithm 5.2. Ranking and generating the summary via maximizing marginal relevance

```
Input: set of sentences R, score of each sentence, semantic similarity matrix, summary size n
Output: set of summary sentences S,
S←∅
for n=1, …, n do
$maxPos = argmax_{i:\, s_i \epsilon R \setminus S} \left[\lambda * score(s_i) - (1-\lambda) * max_{s_j \epsilon S} * sim(s_i, s_j)\right]$
    S←S U R(maxPos)
    R←R\ R(maxPos)
end for
return S
```

As shown in Algorithm 5.2, the sentence is incorporated if it is highly ranked and its similarity to any existing sentence in the summary must not be very high. First, the sentence with the highest rank is added to the summary S and removed from the ranked list *R*. The next sentence with the highest re-ranked score from equation (5.9) is selected from the ranked list. It is then deleted from the ranked list and added to the summary. The same process is repeated until the summary attains the predefined length. The MMR method works according to the following equation:

$$MMR = argmax_{s_i \epsilon R \setminus S} \left[\lambda * score(s_i) - (1-\lambda) * max_{s_j \epsilon S} * sim(s_i, s_j)\right] \quad (5.9)$$

Where *R* is a set of sentences; *S* is a set of summary sentences; λ is a tuning factor between the importance of a sentence and its significance to formerly chosen sentences; $score(S_i)$ is the initial ranking score for sentence S_i and $sim(S_i, S_j)$ is the cosine similarity measure between S_i and S_j.

5.2.9 Proposed models

In this work, we have adopted three basic document representation models from which other models are created. The basic document representation models are noted as follow:

- **BOW:** is the traditional document representation model which is built from the $TF.IDF$ matrix of each document. The rows of the matrix represent the document sentences and the columns represent the set of words of the summarization corpus. The set of words are chosen from the 1000 high-frequency words in the vocabulary.
- **Sent2Topic_prob:** this is the first representation model based on the topic space. The document to be summarized is represented by a matrix in which the rows represent the sentences and the columns represent the set of topics belonging to the cluster of the document. Each row can be expressed as a vector in the topic space, in which the value for each topic (i.e. the value in the matrix) is the probability of the topic, given a set of words w_i of the given sentence. This probability is calculated by Equation (5.6).
- **Sent2Topic_w2v:** this is the second representation model that we have built from the identified topic space. The difference between this model and Sent2Topic_prob is that the values of the matrix representation is calculated based on the word2vec semantic similarity between the given sentence and each topic of the document cluster.

Other models are built from the combination of the basic models explained above and unsupervised neural networks models. They can be divided into simple models and ensemble learning models:

Simple models

Simple models use one document representation in the summarization task. The information used to rank sentences is provided from a unique unsupervised feature leaning algorithm, which are based on different unsupervised neural networks models namely the deep learning AE (Hinton and Salakhutdinov, 2006), the deep learning VAE (Kingma and Welling, 2014), and the neural network ELM-AE (Kasun et al., 2013). Each of those models is an unsupervised feature learning algorithm, which constructs a low-dimensional concept space to represent abstract features from unlabeled data. To simplify the reading and understanding of this work, we provide for each model the following notation:

- **AE_BOW** indicates the system based on the auto-encoder model. In this case, the AE is trained on the BOW representation.
- **AE_Sent2Topic_prob:** indicates the system based on the auto-encoder model. In this case, the AE is trained on the **Sent2Topic_prob** representation.
- **AE_Sent2Topic_w2v:** indicates the system based on the auto-encoder model. In this case, the AE is trained on the **Sent2Topic_w2v** representation.

- **VAE_BOW** indicates the system based on the variational auto-encoder model. In this case, the VAE is trained on the BOW representation.
- **VAE_ Sent2Topic_prob** indicates the system based on the variational auto-encoder model. In this case, the VAE is trained on the **Sent2Topic_prob** representation.
- **VAE_ Sent2Topic_w2v** indicates the system based on the variational auto-encoder model. In this case, the VAE is trained on the **Sent2Topic_w2v** representation.
- **ELM-AE_BOW** indicates the system based on the extreme learning machine auto-encoder model. In this case, the ELM-AE is trained on the BOW representation.
- **ELM-AE_ Sent2Topic_prob** indicates the system based on the extreme learning machine auto-encoder model. In this case, the ELM-AE is trained on the **Sent2Topic_prob** representation.
- **ELM-AE_Sent2Topic_w2v** indicates the system based on the extreme learning machine auto-encoder model. In this case, the ELM-AE is trained on the **Sent2Topic_w2v** representation.

Ensemble learning models

Ensemble methods use multiple models, mostly classifiers, which are combined to solve a particular problem. These techniques distill an ensemble of models into a single model in order to aggregate the information provided from different sources. The first source is the BOW representation of the original document. The second source is the features vector obtained by the projection of the document in the topic space. The third, fourth and fifth sources are the features learned by the AE, VAE and ELM-AE, respectively. In this work, we have adopted the following ensemble learning models:

- **Ensemble BOW_Sent2Topic_prob:** designs the combination of BOW and Sent2Topic_prob models
- **Ensemble BOW_Sent2Topic_w2v:** designs the combination of BOW and Sent2Topic_w2v models
- **Ensemble NN_Sent2Topic_prob** denotes the ensemble learning model combining unsupervised neural networks (AE, VAE and ELM-AE) and Sent2Topic_prob representation. In this ensemble learning model, AE, VAE and ELM-AE are trained on Sent2Topic_prob matrix and noted **AE_Sent2Topic_prob**, **VAE_ Sent2Topic_prob** and **ELM-AE_ Sent2Topic_prob,** respectively.
- **Ensemble NN_Sent2Topic_w2v** denotes the ensemble learning model combining unsupervised neural networks (AE, VAE and ELM-AE) and Sent2Topic_w2v representation. In this ensemble learning model, AE, VAE and ELM-AE are trained on Sent2Topic_w2v matrix and noted **AE_Sent2Topic_w2v**, **VAE_ Sent2Topic_w2v** and **ELM-AE_ Sent2Topic_w2v,** respectively.

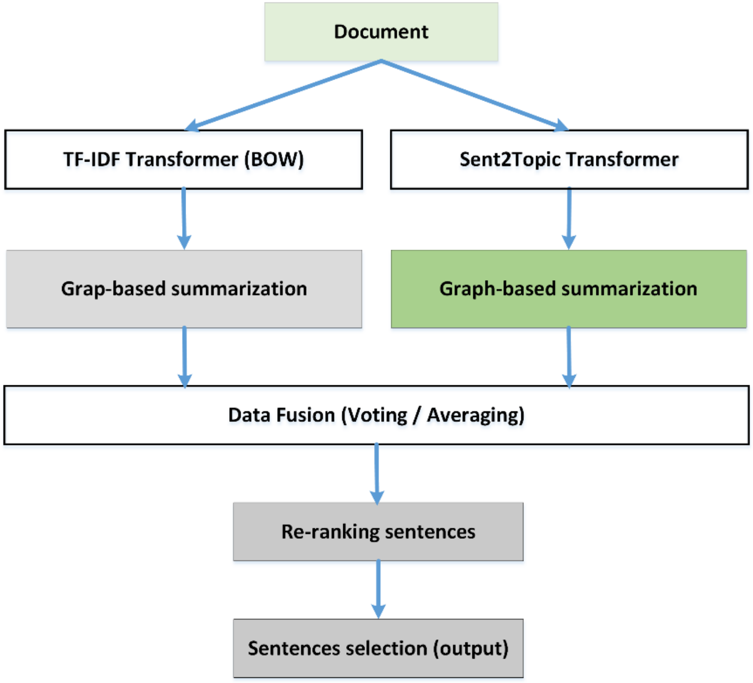


Figure 5.7 The ensemble method combining two models: topic representation and BOW representation for text summarization

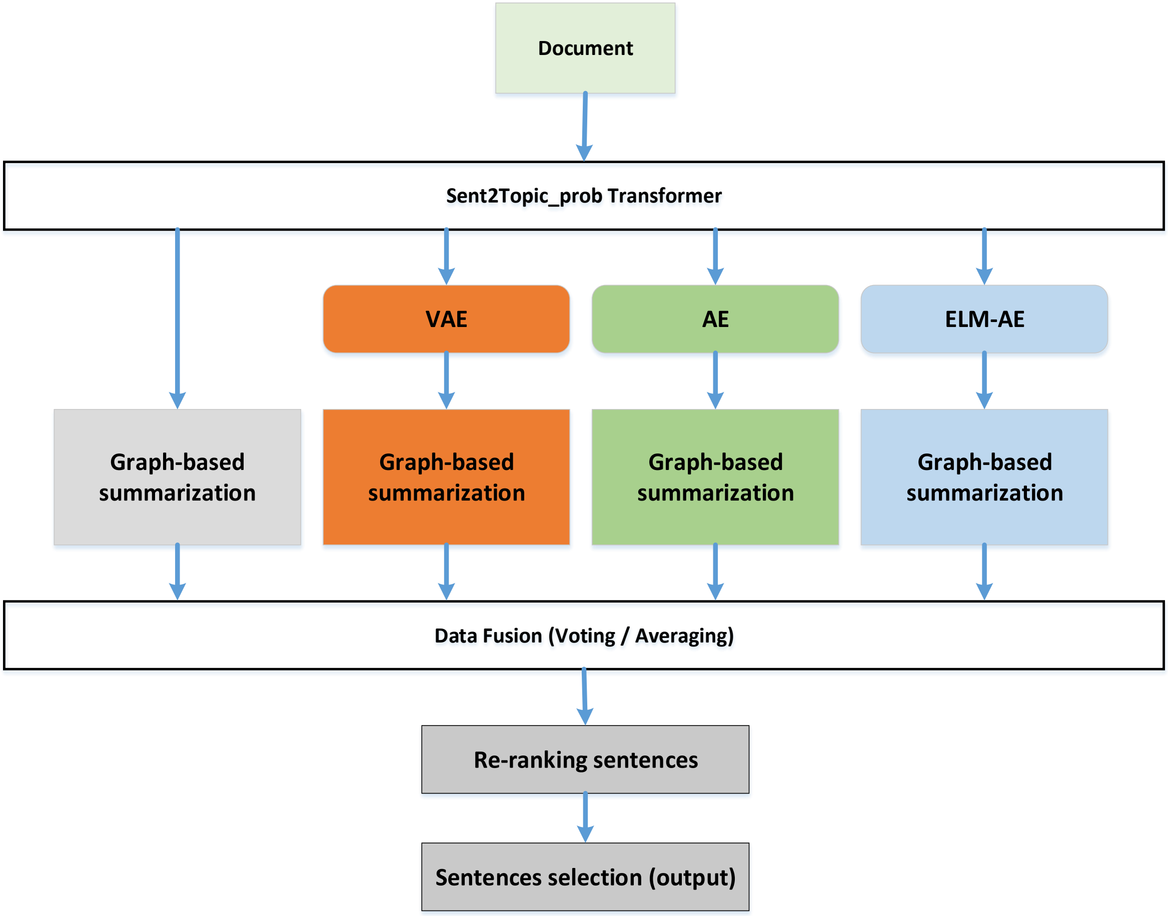

Document
Sent2Topic_prob Transformer
VAE
AE
ELM-AE
Graph-based summarization
Graph-based summarization
Graph-based summarization
Graph-based summarization
Data Fusion (Voting / Averaging)
Re-ranking sentences
Sentences selection (output)

Figure 5.8 The ensemble method combining four models: topic representation, AE, VAE and ELM-AE

The ensemble learning technique used in this work provides the reliable result using averaging and majority voting technique. In the majority-based model, the majority of the combined models are used as the final prediction. In the averaging-based model, the average of predictions from all the models is computed and used to provide the final prediction. An example of two ensemble learning models are presented in Figure 5.7 and Figure 5.8.

5.3 Experimental design and results

5.3.1 Training dataset

Training is an essential phase for building a powerful machine learning systems. In this work, we used CNN and BBC corpus (Saad and Ashour, 2010) for training our proposed models in three phases:

- In the clustering phase we train the ELM-AE model and k-mean algorithms to generate different clusters for our dataset. For the best performance in the experiments, we have chosen the number of clusters k=10
- In topic identification phase, we train the LDA model that gives us better topics for each cluster. For the best performance in the experiments, we have chosen the number of topics in each cluster equal to 250 topics.
- In the summarization phase, we train our proposed neural networks models used in this work for unsupervised feature learning. For each model, we build a concept space that gives a latent representation for the input data. As explained above three kind of unsupervised neural networks are used: AE, VAE and ELM-AE.

5.3.2 Word2Vec model

In order to compute the semantic similarity between sentences and topic terms, we have built a distributed vector representation of Arabic words, which is the Arabic word2vec model used to construct the **Sent2Topic_w2v** representation. To obtain the Arabic werd2vec model, we have adopted the Skip-gram method which is trained on a large Arabic datasets composed by:

- OSAC corpus (Saad and Ashour, 2010), which consists of 22,429 articles collected from multiple Arabic websites. The dataset is divided into 11 topics: Economics, History, Entertainment, Education and Family, Religious, Sports, Astronomy, Health, Law, Stories, and Cooking Recipes. The dataset contains about 18,183,511 words and 449,600 district keywords after removing stop-words.
- BBC corpus (Saad and Ashour, 2010), which consists of 4,763 articles divided into 7 topics: Middle East News, World News, Business and Economy, Sports, Science and Technology, Art and Culture, International Press. The dataset contains 1,860,786 words and 106,733 district keywords after removing stop-words.
- CNN corpus (Saad and Ashour, 2010), which consists of 5,070 articles divided into 6 topics: Business, Entertainment, Middle East News, World News, Science and Technology, Sports. The dataset contains 2,241,348 words and 144,460 district keywords after removing stop-words.
- Wikipedia corpus, which is the full database dump of Arabic articles freely provided by Wikipedia.

Our Arabic word2vec model has been obtained using the Word2Vec implementation of Gensim python library. A vector of 200 dimensions has been generated for each word in the corpus. Experiment setup

5.3.3 Experiment setup

In order to have a thorough assessment of the proposed models, we perform several experiments on the EASC dataset that is especially designed for summarization. We designed an experimental phase in which we compared the results of the summarization task using different document representation models. As mentioned above, we have adopted three basic document representation models: BOW, Sent2Topic_prob and Sent2Topic_w2v.

In addition, other models are built based on the combination of the basic models and unsupervised neural networks models. Several architectures of these models are tested and evaluated by changing the network parameters. For each model we have chosen the best parameters that perform efficiently the summarization task and give us the best results. In our experimentation, the AE is composed of one hidden layer with 20 units that represent the concept space. The VAE is composed of two hidden layer with 200 units in the first hidden layer and 20 units in the second layer. The ELM-AE is composed of one hidden layer with 50 hidden units.

5.3.4 Results and discussion

We investigate the performance of graph-based summarization of different proposed models on EASC dataset by calculating the Rouge-1 and Rouge-2 using different compression ratio (CR).

5.3.4.1 Experiments without using redundancy-removal component

In the first experiment (1), we ran our algorithm on the EASC dataset using different document representation models: Sent2Topic_prob, Sent2Topic_w2v, and the baseline BOW representation, which is built from the *TF.IDF* matrix of each document. Table 5.1 summarizes the results this experiment. By analyzing the results of this experiment, we shown that the proposed approaches based on topic modeling outperforms the baseline approach based on BOW representation. For all summary sizes (compression ratios), graph-based Arabic summarization using Sent2Topic_prob representation and Sent2Topic_prob, which is based on the distributed word2vec model, are better than the baseline BOW representation, which is based on real *TF.IDF* matrix. This shows that the projection of sentences in the topic space gives a better representation and provides relevant information about the input Arabic document. We can also conclude from Table 5.1, when comparing the ROUGE-1 results of the 10% of CR with the results of the 40% of CR that the recall decreases when the compression ratio goes down because the co-occurrence between candidate summary and gold summary increases.

In the second experiment (2), we exposed the results of different Arabic summarization methods based on unsupervised neural networks: AE, VAE and ELM-AE. As explained above, we used the three document representation models as the input for training the proposed models: BOW, Sent2Topic_prob and Sent2Topic_w2v. The results of this experiment are presented in Table 5.2.

Table 5.1 ROUGE-1 measure of graph-based Arabic summarization system using different document representation models

	Compression ratio (CR)						
Models	10%	15%	20%	25%	30%	35%	40%
BOW	0.2881	0.3578	0.4348	0.4862	0.5362	0.5868	0.6264
Sent2Topic_prob	0.3169	0.3897	0.4687	0.5252	0.5659	0.6184	0.6625
Sent2Topic_w2v	0.3141	0.3797	0.4637	0.5186	0.5615	0.6108	0.6594

Table 5.2 ROUGE-1 measure of graph-based Arabic summarization system using unsupervised neural networks trained on different document representation models

	CR						
Neural Networks Models	10%	15%	20%	25%	30%	35%	40%
AE-based models							
AE_BOW	0.1936	0.2512	0.3125	0.3631	0.4075	0.4631	0.5118
AE_ Sent2Topic_prob	0.2569	0.3209	0.3893	0.4518	0.5001	0.5612	0.6162
AE_ Sent2Topic_w2v	0.3022	0.3689	0.4482	0.5020	0.5420	0.5979	0.6442
VAE-based models							
VAE_BOW	0.2688	0.3312	0.4097	0.4691	0.5125	0.5637	0.6057
VAE_Sent2Topic_prob	0.3004	0.3578	0.4298	0.4909	0.5370	0.5848	0.6276
VAE_Sent2Topic_w2v	0.3047	0.3692	0.4519	0.5093	0.5606	0.6239	0.6760
ELM-AE-based models							
ELM-AE_BOW	0.2431	0.2970	0.3622	0.4101	0.4480	0.4887	0.5372
ELM-AE_Sent2Topic_prob	0.2978	0.3631	0.4464	0.5078	0.5468	0.6017	0.6454
ELM-AE_Sent2Topic_w2v	0.2969	0.3620	0.4555	0.5154	0.5586	0.6086	0.6541

Table 5.3 ROUGE-1 measure of the proposed ensemble learning models

	CR						
Ensemble models	10%	15%	20%	25%	30%	35%	40%
BOW_Sent2Topic_prob	0.4592	0.5059	0.5591	0.5978	0.6335	0.6746	0.7079
VAE_Sent2Topic_prob	0.4697	0.5122	0.5641	0.6115	0.6440	0.6780	0.7115
BOW_Sent2Topic_w2v	0.4645	0.5100	0.5641	0.6035	0.6391	0.6754	0.7021
NN_Sent2Topic_prob	0.4688	0.5186	0.5702	0.6195	0.6543	0.6848	0.7121
NN_Sent2Topic_w2v	0.4672	0.5138	0.5675	0.6143	0.6434	0.6808	0.7167

As reported in Alami et al. (2018), Table 5.2 shows that the VAE_BOW provides the best results compared to AE_BOW and ELM-AE_BOW. In addition, ELM-AE gives better results than other models when Sent2Topic_prob representation is used to train our neural networks. However, the results of the AE are better with Sent2Topic_w2v than other matrix representations. We also shown that the proposed models are effectives when they are trained on Sentence2Topic and Sent2Topic_w2v than classical BOW representation.

According to the results exposed in Table 5.2, we found that among the proposed models trained on BOW, the VAE is the best unsupervised feature learning algorithm that gives a better summary. The VAE, in this case, use the Sent2Topic_w2v matrix representation as the input of the model.

By analyzing the results exposed in Table 5.2, we prove that the representation model based the topic space is more reliable and contains more information as the one given by the traditional BOW representation.

In the third experiment (3), we reported the evaluation results of different Arabic summarization systems based on the proposed Ensemble learning models. The results of this experiment are presented in Table 5.3. By analyzing the results of the experiment (3), we can report that the best result is achieved by the proposed ensemble learning model NN_Sent2Topic_prob which is built from the combination of four models using majority voting technique: Sent2Topic_prob representation with the AE, VAE and ELM-AE which are the unsupervised neural networks trained by using the matrix formed by the Sent2Topic_prob representation. This leads us to conclude that the information contained in each vector among the four document representation models are complementary to each other, and that is the reason why the combination get a better results.

5.3.4.2 Experiments with redundancy-removal component

Alami et al., 2015 shows that removing redundancy is an important part of Arabic text summarization. The quality of the final summary is significantly improved by adopting a redundancy-removal component. For further improvement of our proposed models, we have adopted an adapted version of the Maximal Marginal Relevance (MMR) algorithm for redundancy elimination and information diversity (for more information refer to section 5.2.8).

It is worth noting that in experiments (1), (2) and (3), we applied the proposed Arabic summarization approaches without using any redundancy elimination technique. In the following experiments, we applied the MMR technique on the ranking result obtained by the graph model. As a result for applying MMR algorithm, there is no redundant information and more information related to the content of the document is included in the final generated summary.

In the experiment (4), the Arabic summarization is performed using the proposed documents representation models and MMR for redundancy elimination. The difference between experiment (1) and this one is in the sentence selection phase. In experiment (1), sentences are selected according to their ranking score obtained by the application of the proposed model. However, in the experiment (4), after calculating the initial rank of each sentence by the proposed model, the MMR algorithm is applied in order to re-rank the sentences and avoid redundant information, which consists of eliminating unneeded sentences that are similar to already selected sentences.

By comparing the results exposed in Table 5.1 and Table 5.5, it can be noticed that eliminating redundancies enhances the quality of the final summary. The experiment (4) achieved higher values of Rouge-1 recall than experiment (1) for all the summary sizes. At compression ratio 15%, the Rouge-1 recall achieved by experiment (1) is 0.3897, while the Rouge-1 recall achieved by experiment (4) is 0.4937. The recall is increased by 0.104. We conclude that the summary quality was improved when the MMR was applied to the proposed model.

The results of the experiment (5) are presented in Table 5.5. It is clear that even the models based on the proposed neural networks give better results when using the MMR technique. The proposed models evaluated in the experiment (5) achieved higher values of Rouge-1 compared to the same models evaluated in the experiment (2).

Table 5.4 ROUGE-1 recall of the proposed Arabic summarization systems with MMR using different document representation models and summary sizes

	CR						
Models	10%	15%	20%	25%	30%	35%	40%
BOW	0.4164	0.4685	0.5395	0.5887	0.6243	0.6676	0.7056
Sent2Topic_prob	0.4328	0.4937	0.5619	0.6094	0.6448	0.6962	0.7336
Sent2Topic_w2v	0.4117	0.4696	0.5329	0.5901	0.6285	0.6802	0.7248

Table 5.5 ROUGE-1 recall of the proposed Arabic summarization systems with MMR using different unsupervised neural networks models trained on different document representation models

	CR						
Neural Networks Models	10%	15%	20%	25%	30%	35%	40%
AE-based models							
AE_BOW	0.2931	0.3484	0.4063	0.4516	0.4847	0.5413	0.5835
AE_ Sent2Topic_prob	0.3849	0.4441	0.5031	0.5533	0.5978	0.6489	0.6913
AE_ Sent2Topic_w2v	0.4130	0.4709	0.5362	0.5822	0.6217	0.6661	0.7136
VAE-based models							
VAE_BOW	0.3614	0.4243	0.4916	0.5435	0.5874	0.6262	0.6632
VAE_Sent2Topic_prob	0.4043	0.4597	0.5231	0.5720	0.6114	0.6571	0.6994
VAE_Sent2Topic_w2v	0.4294	0.4958	0.5604	0.6170	0.6594	0.7013	0.7388
ELM-AE-based models							
ELM-AE_BOW	0.3458	0.3986	0.4552	0.4934	0.5272	0.5758	0.6118
ELM-AE_Sent2Topic_prob	0.4208	0.4819	0.5463	0.5924	0.6277	0.6765	0.7124
ELM-AE_Sent2Topic_w2v	0.4075	0.4658	0.5360	0.5894	0.6258	0.6716	0.7127

Table 5.6 ROUGE-1 recall of the proposed ensemble learning models with MMR

	Compression ratio						
Ensemble Models	10%	15%	20%	25%	30%	35%	40%
Majority voting technique							
BOW_Sent2Topic_prob	0.5339	0.5769	0.6295	0.6719	0.7002	0.7303	0.7596
BOW_Sent2Topic_w2v	0.5319	0.5693	0.6274	0.6658	0.6927	0.7297	0.7636
NN_Sent2Topic_prob	0.5487	0.5946	0.6404	0.6758	0.6966	0.7315	0.7631
NN_Sent2Topic_w2v	0.5202	0.5697	0.6224	0.6634	0.6950	0.7340	0.7705
Averaging technique							
BOW_Sent2Topic_prob	0.4274	0.4882	0.5613	0.6081	0.6479	0.6899	0.7348
BOW_Sent2Topic_w2v	0.4080	0.4619	0.5305	0.5756	0.6197	0.6698	0.7138
NN_Sent2Topic_prob	0.4184	0.4733	0.5406	0.5890	0.6274	0.6686	0.7141
NN_Sent2Topic_w2v	0.4080	0.4659	0.5325	0.5852	0.6228	0.6698	0.7178

As shown in the second experiment (2), the VAE based on the Sent2Topic_w2v matrix representation is the best unsupervised feature learning algorithm that gives a better summary.

ELM-AE works better than other models when Sent2Topic_prob representation is used to train the proposed neural networks. However, the best results achieved by the AE is obtained when the input training matrix is Sent2Topic_w2v representation. We also shown that the proposed models are effectives when they are trained on Sentence2Topic_prob and Sent2Topic_w2v than classical BOW representation.

In the experiment (6), we investigated the performance of the proposed ensemble learning models using MMR algorithm. Each ensemble model is composed by two or four models. To aggregate the information provided from different model, we investigated two kind of ensemble techniques: majority voting and averaging technique. The results of this experiment are presented in Table 5.6.

By analyzing the results of experiment (6), we can report that the best result in term of Rouge-1 recall is achieved by the proposed ensemble learning model NN_Sent2Topic_prob which is built from the combination of four models using majority voting technique: Sent2Topic_prob representation, AE, VAE and ELM-AE. The unsupervised neural networks are trained using the matrix built from the Sent2Topic_prob representation.

Considering the previous results obtained by the same models in the experiment (3) (Table 5.3), it is obvious that the proposed ensemble approaches using MMR techniques outperform the same models without using MMR. In addition, the ensemble models based on averaging technique give a reasonable recall measure but they do not match the performance of those based on majority voting technique. A further comparison between these two ensemble techniques using F-measure metric is given in Table 5.7.

5.3.5 Comparison with other methods

The aim of the experiments (7) and (8) is to evaluate the performance of the proposed approaches against other existing Arabic summarization approaches. Our proposed methods were compared with a set of the baseline approaches already evaluated in the works of Alami et al. (2018) and Al-Radaideh and Bataineh (2018), since they used the same evaluation metrics and dataset. The second system is Query-based VAE.

In the experiment (7) we compared the performance of the proposed approach that achieved better result in terms of Rouge-1 recall with the results published in Alami et al. (2018), where many summarization systems have been evaluated. The first system, which is a Graph-based VAE was proposed by Alami et al. (2018). The authors introduced a graph-based Arabic summarization system based on the unsupervised deep learning algorithm VAE. The second system introduced by the same authors is a Query-based VAE, which used the input query to rank the sentences according to their semantic similarity. The semantic similarity is calculated using the concept space produced by the VAE.

The third system is LSA-based summarization approach**.** It is based on the Latent semantic analysis (LSA) (Mashechkin and Petrovskiy, 2011) algorithm to extract features from the input BOW representation and represent them in a contextual and low-dimensional concept space. The extracted features are used in a graph model to rank sentences according to the PageRank algorithm (Brin and Page, 1998).

The fourth system is TextRank (Mihalcea and Tarau, 2004). TextRank is a graph-based ranking model used for both automatic text summarization and key-words extraction. It is based on PageRank (Brin and Page, 1998) algorithm in order to rank the graph elements that better describe the text. In the summarization task, each sentence is represented by a node in the graph and the edge between two nodes represents the similarity relation that is measured as a content overlap between the given sentences. The weight of each edge indicates the importance of a relationship. Sentences are ranked based on their scores and those that have very high score are chosen.

The fifth system is a simple Arabic text summarizer based on TF.ISF feature. The summary is generated from the highest scored sentences. The score of each sentence is computed as follows:

$$score(S_i) = \frac{\sum_{wj \in S_i} TF.ISF(w_j)}{rootCount(S_i)} \tag{5.10}$$

Where $TF.ISF(w_j)$ is the term frequency / inverse sentence frequency of the root w_j*; and* $rootCount(S_i)$ is the number of root in the sentence.

Table 5.7 draws a comparison between our system (Ensemble NN_Sent2Topic_prob) and competitors in term of Rouge-1 recall. We can easily notice that our system has the highest value of Rouge-1 score, and outperforms all the other systems whether with then use of redundancy elimination technique or not. At compression ratio 30%, the best Rouge-1 result obtained by the other systems was reported by the query-based VAE approach (Alami et al. (2018) with 0.403. The second good result was reported by the graph-based VAE introduced by Alami et al. (2018) with 0.4021 of Rouge-1 recall at 30% of summary size. Whereas, in our experiment, Rouge-1 score of the proposed method is 0.6966 when applying the MMR technique and 0.6543 when MMR was not applied. In terms of Rouge-1 recall, the performance of the proposed approach is increased by 0.2936 with the use of MMR and by 0.2513 without the use of MMR. This amounts to saying that our algorithm achieves better results compared to the state-of-the-art and enhances the performance Arabic summarization systems.

Table 5.7 Rouge-1 recall comparison of the proposed approach against other methods on EASC corpus with different summary size

	CR (%)			
Methods	10%	20%	30%	40%
Proposed approaches without MMR	0.4688	0.5702	0.6543	0.7121
Proposed approaches with MMR	0.5487	0.6404	0.6966	0.7631
Graph-based VAE (Alami et al., 2018)	0.1101	0.2825	0.4021	0.5298
Query-based VAE (Alami et al., 2018)	0.115	0.286	0.403	0.526
LSA (Topic-based)	0.1045	0.2559	0.3608	0.4312
TextRank	0.1197	0.2819	0.3892	0.5014
Baseline Tf.ISF	0.106	0.269	0.379	0.503

From the results presented in Table 5.7, it is obvious that our method outweighs all other methods thanks to the fact that our system can spot the relationships between sentences using their projection in a topic space. These relationships cannot be identified by the reference systems used in the experimentation. In addition, these results clearly indicate that when the information is provided from several sources (different models), the system generates an effective and meaningful summary.

Experiment (8) presents comparisons between the proposed approaches and some other Arabic summarization systems namely Al-Radaideh and Bataineh (2018), Al-Khawaldeh and Samawi (2015) (LCEAS), Oufaida et al. (2014) and Al-Omour (2012). A primary comparison between these methods has been reported by Al-Radaideh and Bataineh (2018), where the summary of each document in the EASC corpus was generated using two different summary sizes 25% and 40%. The evaluation was made based on Rouge-1 and Rouge-2 metrics.

Table 5.8 shows a comparison of our proposed approaches with these techniques by calculating the average of Rouge-1 recall and Rouge-1 F-measure. By analyzing these results, we can noticed the following: (i) in terms of Rouge-1 recall, the best result was achieved by the proposed Ensemble NN_Sent2Topic_prob with majority voting technique for both summary sizes 25% and 40%; (ii) all the proposed approaches outperforms the other summarization systems in terms of Rouge-1 recall; (iii) in terms of F-measure, the best result is achieved by the proposed VAE_SentTopic_w2v with 0.5433 at CR 25% and 0.5452 at CR 40%; (iv) at compression ratio 25%, all the proposed ensemble models based on averaging technique outperform the other competitors for both Rouge-1 recall and F-measure. However the results of F-measure obtained by majority voting technique are not satisfactory for both CR 25% and 40%; (v) at compression ratio 40% the second best result classed after the proposed VAE_SentTopic_w2v is achieved by Al-Radaideh and Bataineh (2018) with 0.542.

Table 5.9 shows a comparison between the proposed approaches against the competitors using Rouge-2 recall and F-measure for both summary sizes 25% and 40%. After analyzing these results, we noticed that the best result is reported by the proposed VAE_Sentence2Topic_w2v which outperforms the other Arabic summarization systems in terms of Rouge-2 recall and F-measure. In addition the results obtained by the proposed ensemble models with averaging technique are better than the results obtained by the competitors. Except when a summary size is 40%, the F-measure of the system proposed by Al-Radaideh and Bataineh (2018) is ranked second after the proposed VAE_Sentence2Topic_w2v and close to the proposed Ensemble BOW_Sent2Topic_prob.

By this work, we can also confirm that various reasons account for the difficulty to compare the proposed approach to other existing systems. Firstly, unlike English, there is no approved benchmark reference for Arabic language against which to assess our approach in Arabic text summarization. Hence, the comparison of the performance of the proposed approaches is intricate given that a different dataset and different evaluation measures are used in each work. Dissimilarly, benchmarking in English can rely on DUC human generated summaries. Moreover, the community working on Arabic text summarization is still quite small. Add to this, lexical, syntactic, and semantic ambiguity are higher in Arabic because of the complexity of the language as far as spelling, vocabulary and morphology are concerned.

As mentioned above, with the proposed approaches, we do not need to have labeled data, which consist in this case of a set of documents with human-generated summaries. These labeled data are very difficult to obtain, especially for Arabic, due to the luck in annotated summarization corpus designed for Arabic on the one hand and to the difficulty of manually creating summaries on the other hand. By contrast, with the emergence of the Internet and the digital world, unlabeled data become more widely available compared to labeled data. Thus, the availability of vast amounts of unlabeled data has made it imperative to adopt unsupervised learning framework in order to construct an automatic summarization model designed for Arabic documents. It is one of the strengths of the proposed method.

Table 5.8 Rouge-1 comparison of the proposed approach against other methods on EASC corpus with 25% and 40% of compression ratio

	CR=25%		CR=40%	
Proposed Methods	Recall	F-measure	Recall	F-measure
VAE_Sent2Topic_w2v	0.6170	**0.5433**	0.7388	**0.5452**
Proposed Ensemble learning with majority voting technique				
Ensemble BOW_Sent2Topic_prob	0.6719	0.2115	0.7596	0.1674
Ensemble BOW_Sent2Topic_w2v	0.6658	0.2113	**0.7636**	0.1679
Ensemble NN_Sent2Topic_prob	**0.6758**	0.2104	**0.7631**	0.1669
Ensemble NN_Sent2Topic_w2v	0.6634	0.2104	0.7705	0.1698
Proposed Ensemble learning with averaging technique				
Ensemble BOW_Sent2Topic_prob	0.6081	0.5037	0.7348	0.5124
Ensemble BOW_Sent2Topic_w2v	0.5756	0.4986	0.7138	0.5141
Ensemble NN_Sent2Topic_prob	0.5890	0.4938	0.7140	0.4986
Ensemble NN_Sent2Topic_w2v	0.5852	0.4949	0.7178	0.5032
Competitors				
Al-Radaideh and Bataineh (2018)	0.395	0.476	0.588	0.542
Oufaida et al. (2014)	0.420	0.370	-	-
Al-Omour (2012)	0.324	0.411	0.449	0.485

Table 5.9 Rouge-2 metrics of the proposed approaches and competitors

	CR=25%		CR=40%	
Proposed Methods	Recall	F-measure	Recall	F-measure
VAE_Sentence2Topic_w2v	0.5026	**0.4435**	0.6310	**0.4617**
Proposed Ensemble learning with averaging technique				
Ensemble BOW_Sent2Topic_prob	0.4589	0.3844	0.5992	0.4207
Ensemble BOW_Sent2Topic_w2v	0.4229	0.3702	0.5738	0.4150
Ensemble NN_Sent2Topic_prob	0.4403	0.3740	0.5718	0.4028
Ensemble NN_Sent2Topic_w2v	0.4361	0.3744	0.5767	0.4067
Competitors				
Al-Radaideh and Bataineh (2018)	0.334	0.372	0.465	0.422
Oufaida et al. (2014)	-	-	0.290	0.260
Al-Khawaldeh F, Samawi (2015)	-	-	0.270	0.28

5.4 Conclusion

In this chapter, we proposed a new Arabic summarization method based on clustering, topic modeling unsupervised neural networks, ensemble learning models which aggregates the information provided from the topic space and other representation models. A big collection of Arabic document is used to perform document clustering using ELM-AE algorithm and k-mean technique. For each cluster, the LDA algorithm is used to identify the topic space belonging to each cluster. A numerical document representation is then formed based on the topic space. This new representation is used as the input of the graph-based summarization technique. We have experimented with the EASC dataset designed to evaluate the summarization task for Arabic languages. The results confirm that sentence representation in a topic space encapsulates relevant information and achieves better result than representation based on BOW approach. Also, we show that Ensemble methods that aggregate information from different models using majority voting and averaging techniques outperforms significantly the performances of the summarization task and obtains the best accuracy compared the state-of-the-art in Arabic document summarization.

Conclusion and Perspectives

In this PhD thesis, we have investigated and improved automatic summarization with a particular focus on Arabic text summarization.

In chapter 1, we have presented a detailed study of general automatic text summarization systems and approaches. Then, we have focused our study on existing works and approaches designed for Arabic text summarization. Based on the literature review established in this chapter, several challenges have been identified and addressed. First, we concluded that, the research works carried out on this area have experienced lately strong progress especially for English. However, researches in Arabic text summarization are very few and are still in their beginning. Second, we concluded that most of existing Arabic summarization systems do not consider the semantic relationships among textual units (words, phrases, sentences, etc...) and redundant informations are repeated in the final summary. Third, we found that most of the existing works are based on traditional bag-of-words representation model, which involves a sparse and high-dimensional input data and ignores semantic relationships between words and sentences. Therefore, capturing relevant information in a document to be summarized by these systems is a complex task. Fourth, traditional Arabic summarization approaches do not consider the themes and topics existing in the processed documents. These themes and topics, if they are identified and considered, can be helpful in extracting important information from the original document, so the quality of the summarization task can be improved.

Considering these limitations and shortcomings, we have proposed several contributions in this thesis work in order to improve the automatic summarization task of Arabic documents.

Our first contribution is presented in chapter 2, in which we propose a new graph-based Arabic summarization system that combines statistical and semantic analysis. The proposed approach utilizes ontology hierarchical structure and relations to provide a more accurate similarity measurement between terms in order to improve the quality of the summary. The proposed method is based on a two-dimensional graph model that makes uses statistical and semantic similarities. The statistical similarity is based on the content overlap between two sentences, while the semantic similarity is computed using the semantic information extracted from a lexical database whose use enables our system to apply reasoning by measuring semantic distance between real human concepts. The weighted ranking algorithm PageRank is performed on the graph to produce significant score for all document sentences. In addition, we have addressed the redundancy and information diversity issues by using an adapted version of Maximal Marginal Relevance method. Experimental results on EASC and our own datasets showed the effectiveness of our proposed approach over existing summarization systems. We have also investigated the effect of the stemming process in Arabic summarization task. Stemming is a process of reducing inflected words to their stem or root from a generally written word form. This process is used in many text mining application as a feature selection technique. Therefore, we have evaluated the impact of three different Arabic stemmers (i.e. Khoja, Larekey and Alkhalil’s stemmer) on Arabic text summarization performance. The evaluation of the proposed system, with the three different stemmers and without stemming, on the dataset used shows that the best performance was achieved by Khoja stemmer in term of recall, precision and F1-measure. The evaluation also shows that the performances of the proposed system are significantly improved by applying the stemming process in the pre-processing stage.

In chapter 3, we detailed our second contribution, which consists of adopting an unsupervised deep learning algorithm for summarizing Arabic documents. We have proposed to use a variational auto-encoder (VAE) model to learn a new feature space from a high-dimensional input data. We have explored several input representations such as term frequency (*tf*), *tf-idf* and both local and global vocabularies. All sentences are ranked according to the latent representation produced by the VAE. We have investigate the impact of using VAE with two summarization approaches, graph-based and query-based approaches. Experiments on two benchmark datasets specifically designed for ATS shown that the VAE using *tf-idf* representation of global vocabularies clearly provides a more discriminative feature space and improves the recall of other models. Experiment results confirm that the proposed method leads to better performance than most of the state-of-the-art extractive summarization approaches for both graph-based and query-based summarization approaches.

Chapter 4 was devoted to our third contribution, which consists of adopting several deep neural networks models for Arabic summarization task. Deep neural networks have proven their ability to achieve excellent performance in many real-world Natural Language Processing and computer vision applications. However, it still lacks attention in Automatic Text Summarization (ATS). The aim is to discover the underlying low dimensional structure from the given high dimensional data. On the other hand, word embedding is another neural network technique that generates a much more compact word representation than a traditional Bag-of-Words (BOW) approach. The aim of this chapter is to enhance the quality of ATS by integrating unsupervised deep neural network techniques with word embedding approach. First, we have built our Arabic word embedding model by training a large Arabic datasets with word2vec method. Second, we have shown that word2vec-based text summarization gives better results than traditional BOW representation. Third, we have proposed other models by combining word2vec and unsupervised feature learning methods in order to merge information from different sources. We have shown that unsupervised neural networks techniques using word2vec representation give better results than those learned from BOW representation. Fourth, we have also propose three ensemble techniques. The first ensemble combines BOW and word2vec using the majority voting technique. The second ensemble aggregates the information provided by the BOW approach and unsupervised neural networks. The third ensemble aggregates the information provided by word2vec and unsupervised neural networks. We have shown that the ensemble methods improve the quality of ATS, in particular the ensemble based on Word2vec approach gives better results. Finally, we have performed different experiments to evaluate the performance of the investigated models. We have used two kind of datasets that are publically available for evaluating ATS task for English and Arabic documents. Results of statistical studies affirm that word embedding-based models outperform the summarization task compared to those based on BOW approach. In particular, ensemble learning technique with word2vec representation surpass all the investigated models.

In Chapter 5, we detailed our fourth contribution, which consists of enhancing of the previous best results by finding a good representation of Arabic documents. For this purpose, we have proposed new approaches for summarizing a large Arabic documents using unsupervised deep learning and topic representation instead of traditional bag-of-words representation. The summarization task is performed in four major stages. First, a new Arabic document clustering technique using Extreme learning machine is performed on a large text collection in order to group each document in a specific cluster. Second, topic modeling using LDA is applied on documents collection in order to identify topics present in each cluster. Third, each document is represented in a topic space by a matrix where rows represents the document sentences and columns represent the cluster topics. The generated matrix is then trained using different unsupervised neural networks and ensemble learning algorithms in order to build an abstract representation of the input in the concept space. The resulted matrix in the concept space is a latent representation of the original document. This matrix is used to build a graph representation of the document. The weighted ranking algorithm PageRank is performed on the graph to produce significant score for all document sentences. We have improved the proposed approaches by adoption the MMR algorithm for eliminating redundancies and diversifying informations to include in the final summary. Experimental results on EASC showed the effectiveness of our proposed approaches over existing summarization systems.

The promising results found out in this thesis work open several ways for further developments on the field of Arabic summarization systems:

- The first direction is to extend the dataset used in the evaluation process by developing a large corpus of documents with their manual summaries. This will give more value to the proposed approaches.
- A second direction is to consider more language-specific features, such as part-of speech, co-reference and anaphora resolution, which are open research fields in Arabic NLP. Semantic features can also be incorporated by using other knowledge resources, such as Wikipedia and other large corpus in addition to Arabic WordNet, which is not a complete solution to compute the semantic similarity between two words, because of its limit of the overall concept coverage.
- We intend to evaluate the performance of the proposed approaches on a specific domain, such as online reviews, biomedical texts and online tweets. While, online data (Arabic tweets, hotels reviews… etc.) is heavily available on the internet, applying the proposed approaches on training and testing this data can improve the summarization performance in a specific domain.
- In future work, we intend to incorporate more unsupervised deep learning models such as stacked auto-encoders, attention auto-encoder, Restricted Boltzmann machine and the unsupervised version of convolutional neural network. In addition, supervised approaches can help in the improvement of Arabic summarization task. The issue with supervised approaches is that they need annotated corpus to be trained and fine-tuned. We can address this problem by developing a large corpus containing documents taking from the well-known Arabic websites, and generating human-summaries from each documents.
- Several improvements can be made to this work by investigating other summarization approaches such as clustering technique and dynamic programming.

- As of the time of writing, researches in Arabic abstractive summarization are not yet available. Abstractive summarization consists of understanding the main concepts in the original document and presenting them in a shorter document. It requires human knowledge, statistical methods and linguistic methods. Whereas abstractive summarization needs heavy machinery for language generation and is not easy to implement or stretch to larger domains, simple extraction of sentences has yielded positive results in large-scale applications, namely in multi-document summarization. The summarization category addressed in this thesis work is extractive, which involves basic NLP tools to generate the final summary. Automatic processing of Arabic suffers from the lack of resources and natural language generation tools. Thus, it is difficult for researches to address deeply this field. Therefore, we can start tackling Arabic abstractive summarization field by developing tools and resources that can generate a correct sequence of sentences. Among those tools, we exemplify Arabic lexicons, ontologies, a man-developed knowledge and language models.

Appendix A

Résumé détaillé de la thèse en Français

A.1 Introduction

Un résumé est défini comme une forme réduite du contenu d’un texte, d’un document, qui consiste à reproduire de manière succincte les idées les plus pertinentes. Hovy (2005) a défini un résumé comme étant un texte abrégé qui représente l’information la plus pertinente contenue dans un ou plusieurs textes. Cette forme abrégée ne devra pas dépasser la moitié du texte original. Le but ainsi d’un résumé automatique de texte est de produire par la machine une représentation réduite et abrégée d'un ou de plusieurs documents. Le texte ainsi réduit doit rester fidèle aux informations et idées du texte original. En plus des données textuelles, le résumé automatique peut être appliqué à tous les types de données, tels que la parole, les documents multimédias, les images, les vidéos et même toute sorte de combinaison de ces types.

Avec la croissance rapide d'Internet et la multiplicité des supports de stockage de masse, la quantité des documents électroniques et de données textuelles est devenue énorme y compris celles en langue Arabe. Étant donné que les utilisateurs ne peuvent pas gérer manuellement ce volume important de données, et leur manipulation représente une tache fastidieuse, il est devenu nécessaire de trouver un moyen pour extraire et accéder à l’information pertinente de façon rapide permettant aux utilisateurs de gagner le temps et réduire le coût via des méthodes d’analyse automatique. De telles méthodes devraient éviter de parcourir manuellement un grand nombre de documents et faciliter l’accès aux informations pertinentes afin de décider rapidement s’ils présentent un intérêt ou non. Le Résumé Automatique des Textes (RAT) vient pour résoudre ce problème, il permet un gain considérable de temps et de productivité mais aussi une aide à la prise de décision. L’être humain n’a pas besoin de lire la totalité du document pour décider s’il contient l’information rechercher ou pas, en plus la nécessité d’accéder rapidement au contenue. Ainsi, le besoin du RAT, qui est devenu une tâche importante dans le domaine du Traitement Automatique du Langage Naturel (TALN), se fait progressivement sentir pour des raisons de réduction des coûts pouvant résulter de cette automatisation. Le RAT peut aussi être utilisé comme phase préalable pour améliorer la performance d’autres tâches du TALN telles que le clustering, la classification, l’indexation, l’extraction de mots-clés, etc. A ce jour, le RAT est un domaine dynamique en pleine croissance qui présente de nombreux challenges.

Au cours des dernières décennies, la recherche dans le domaine de RAT a suscité un intérêt croissant de la part de la communauté de recherche, reflétée par une croissance exponentielle de la littérature scientifique. Par contre, la littérature traitant le RAT Arabes est relativement petite et les études dans ce domaine ont commencé à apparaître seulement au cours de la dernière décennie (Douzidia and Lapalme, 2004; El-Haj et al., 2011a; Al-Saleh and Menai, 2015). De plus, les systèmes de RAT Arabes n’ont pas atteint le même niveau de maturité et de fiabilité que ceux réalisés pour l'Anglais à cause du manque d'un corpus standard d'évaluation, et des outils fiables d'analyse automatique de l'Arabe.

Dans ce travail de thèse, nous nous intéressons au domaine du RAT en langue Arabe. Nous analysons les travaux les plus importants proposées pour le RAT et en particulier ceux désignés aux textes Arabes. Nous nous intéressons à l'amélioration de ces travaux en proposant des nouvelles méthodes permettant de traiter certains problèmes et limitations identifiées dans la littérature. Dans le cadre de ce travail de thèse, nous avons apporté les contributions suivantes :

1. Dans la première contribution, nous avons proposé une nouvelle méthode de RAT Arabes qui consiste à modéliser le texte à résumer sous forme de graphe bidimensionnel combinant un score statistique et un sémantique. Les nœuds du graphe représentent les phrases du texte et les arcs sont étiquetés par des scores statistiques et sémantiques relatifs au degré de similarité entre chaque paire de phrases. De plus, nous avons intégré un algorithme d'élimination de la redondance pour diversifier les informations qui constituent le résumé final et améliorer davantage la performance de la méthode proposée. Nous avons également montré l'intérêt de la phase de prétraitement (stemming) sur la performance du RAT Arabes.

2. La deuxième contribution consiste à présenter une nouvelle méthode de RAT Arabes basée sur le variationnal auto-encoder (VAE) qui est un modèle d'apprentissage profond non supervisé utilisé pour l'apprentissage des caractéristiques latentes à partir d'un ensemble de documents textuels Arabes. Ainsi, pour chaque phrase, le VAE génère une représentation abstraite qui est exploitée pour classer les phrases du texte et extraire celles les plus pertinentes selon deux techniques : une technique basée sur le modèle graphique et une autre basée sur la similarité par rapport à une requête. Cette méthode permet, d'une part, la réduction de la dimensionnalité, et d'autre part, l'amélioration du processus d'extraction des phrases pertinentes.

3. La troisième contribution consiste à explorer plusieurs modèles dans le cadre du RAT Arabes. Nous adoptons la représentation distribuée des mots (Word2vec) comme données d'entrainement de plusieurs modèles de réseaux de neurones non supervisés. Les nouvelles représentations obtenues des phrases sont utilisées pour calculer la similarité entre les paires des phrases afin de modéliser le texte à résumer sous forme de graphe. Nous proposons aussi de nouvelles méthodes basées sur l'apprentissage ensembliste pour améliorer la performance des méthodes de RAT Arabes. Nous présentons également plusieurs expérimentations pour évaluer la qualité des modèles proposés sur deux types de corpus (Arabe et Anglais).

4. Dans la quatrième contribution, nous décrivons une représentation plus riche des textes Arabes en adoptant les techniques de clustering et de modélisation thématique. Pour cela, nous utilisons la machine d'apprentissage extrême pour regrouper les textes en plusieurs clusters. Ensuite, pour chaque cluster, nous appliquons la méthode d'allocation de Dirichlet latente pour identifier l'espace des sujets associé à chaque cluster. Puis, nous utilisons la représentation des textes dans l'espace thématique de chaque cluster comme données d'entrainement des réseaux de neurones non supervisés et des techniques ensemblistes pour l'apprentissage de nouvelles représentations abstraites des phrases. Cette nouvelle

représentation est exploitée pour modéliser le texte à résumer sous forme de graphe afin de classer les phrases selon leur pertinence.

A.2 Synthèse des méthodes existantes

Le tableau A.1, illustre les approches les plus connues dans le cadre du RAT Arabes. En analysant ces travaux, nous avons conclus ce qui suit :

- La plupart des méthodes proposées pour le RAT Arabes reposent sur des approches statistiques (ou numériques). La caractéristique principale de ces approches est qu'elles reposent sur les mots existants dans le document. Ainsi, l'un des inconvénients évidents est la négligence des relations sémantiques entre les mots. Le système est toujours limité aux mots explicitement mentionnés dans le texte d'origine. Par exemple, si le système ne trouve pas les relations entre des termes tels que «بترول» et «نفط», il traitera ces deux mots séparément en tant que deux termes non liés, ce qui affectera négativement leur importance dans le texte à traiter. Les résumés automatiques basés sur les approches statistiques sont également affectés par les mêmes limitations en matière de détection de concept. Par exemple, avec des expressions telles que « استخراج النفط», « انتاج النفط», « استخراج البترول » et « انتاج البترول», le système devrait pouvoir comprendre que ces expressions font référence au même concept. La relation entre les différents concepts détectés dans le document analysé n'est pas exploitée dans les approches statistiques. La capacité à détecter une telle relation entre les termes et les concepts d'un texte nécessite des connaissances supplémentaires, externes au texte analysé, ainsi qu'un module d'analyse pour apprendre les relations sémantiques entre les différentes unités textuelles du document.

- Les approches basées sur l'apprentissage automatique supervisé telles que (Boudabous et al., 2010; Sobh et al., 2007; Belkebir and Guessoum, 2015; El-Fishawy et al., 2014; Fattah et al., 2009), l'apprentissage est une étape décisive pour améliorer la précision du système. Par conséquent, dans ce type d'approche, tous les mots qui apparaissent dans les documents de test mais pas dans les documents d'apprentissage sont ignorés et aucune nouvelle information, en dehors de ce qui est déjà disponible dans les documents de teste, n'est prise en compte. En plus, ces types d'approches basées sur l'apprentissage supervisé nécessite une base d'apprentissage composée par un grand nombre de documents annotés (paires de documents / résumés) pour apprendre la fonction de prédiction. Dans le cadre de la langue Arabe, il n'existe pas de corpus approuvé pour effectuer adéquatement l'apprentissage supervisé. Par conséquent, les méthodes proposées dans la littérature qui adoptent ce type d'apprentissage, utilisent deux manières pour apprendre la fonction de prédiction. Soit elles se basent sur un corpus limité (seulement quelque dizaine de documents annotés) développé par les auteurs. Ce qui influence négativement la qualité de l'apprentissage automatique qui nécessite par défaut une grande quantité de données annotées. Soit elles utilisent la traduction automatique pour traduire les corpus disponibles en Anglais vers la langue Arabe. Ainsi, les méthodes proposées utilisent ces exemples traduits en Arabe dans leur phase d'apprentissage. Cette manière peut aussi influencer la qualité du RAT Arabes au cas où la traduction automatique n'est pas fiable.

- La majorité des méthodes développées pour l'Arabe n'abordent pas la problématique de redondance et de diversité des informations dans le résumé finale. C'est un problème majeur pour les systèmes de RAT en général et en particulier ceux développés pour l'Arabe. Ainsi, deux phrases ayant une signification similaire représentant les mêmes idées peuvent être incluses dans le résumé si leur score est élevé, et par conséquent, d'autres phrases portant des idées différentes seront exclues vu la taille limite du résumé à produire.

- Le classement des phrases est l'un des étapes clés dans toutes les méthodes de résumé par extraction. De nombreuses recherches ont été menées pour améliorer la qualité de ce processus. Certains travaux ont utilisé des caractéristiques statistiques (Luhn, 1958; Douzidia and Lapalme, 2004; Haboush et al., 2012; Ferreira et al., 2013b; Ferreira et al., 2014) et certaines approches sont basées sur les modèles graphiques (Erkan and Radev, 2004; Mihalcea and Tarau, 2004; Baralis et al., 2013).), tandis que d'autres ont adopté des techniques d'apprentissage automatique supervisées et non supervisées (Sobh et al., 2007; Fattah et al., 2009 ; Boudabous et al., 2010; Fattah, 2014; Yang et al., 2014; Alguliyev et al., 2015). Après avoir étudié ces méthodes, nous avons constaté qu'elles s'appuient sur une approche par sac de mot (Bag-of-Words ou BOW) pour la représentation des documents sous forme numérique. La représentation BOW peut causer deux problèmes majeurs. Premièrement, le système ne dispose pas de suffisamment de données d'observation dans la phase d'apprentissage. Ainsi, les systèmes se basant sur cette approche utilisent une représentation creuse avec des données insuffisantes qui ne portent pas suffisamment d'information (Yousefi-Azar and Hamey, 2017). Deuxièmement, les relations sémantiques entre les mots sont ignorées. De plus, il a été démontré que la représentation distribuée des mots surpasse celle en BOW dans l'identification de la sémantique dans les textes.

- Certains travaux sur le RAT Arabes ont adopté les techniques d'apprentissage automatique comme (Boudabous et al., 2010; Sobh et al., 2007; Belkebir and Guessoum, 2015; El-Fishawy et al., 2014; and Fattah et al.,2009). Cependant, les algorithmes d'apprentissage profond (ou DL) et les réseaux de neurones n'ont pas été suffisamment étudiés dans le cadre du traitement automatique de la langue Arabe, en particulier le RAT. Ces techniques ont prouvé leur efficacité dans plusieurs domaines. Ils ont été utilisés avec succès dans les applications de vision par ordinateur et de TALN, notamment le RAT (Yousefi-Azar and Hamey, 2017). Jusqu'à présent, et d'après la littérature étudiée, les travaux qui adoptent l'apprentissage profond ou les réseaux de neurones pour le RAT Arabes sont rares ou absents. Fattah and Ren (2009) est le seul travail que nous avons trouvé dans la littérature qui utilise des réseaux de neurones supervisés, mais avec un corpus d'entrainement très réduit composé seulement de 100 documents Arabes pour la phase d'apprentissage. Ce qui influence négativement les résultats obtenus par ces modèles, qui nécessitent des corpus d'entrainement très larges pour un apprentissage efficace. Dans notre travail de thèse, nous proposons plusieurs approches utilisant l'apprentissage profond et les réseaux de neurones non supervisés. La raison pour laquelle nous avons adopté l'apprentissage non supervisé, c'est qu'il n'existe pas de corpus standard dédié à la langue Arabe avec un nombre important de documents annotés pour l'apprentissage de la tâche du RAT, alors que les documents non annotés sont largement disponibles sur le web.

- Les travaux existants ne prennent pas en compte le contexte des documents à résumer. Nous supposons que la tâche de résumé peut être améliorée si nous prenons en considération les concepts clés présentés dans le texte. Pour cela, dans notre travail de thèse, nous essayons d'améliorer les modèles basés sur l'apprentissage profond et les réseaux de neurone pour améliorer la qualité des résumés générés en adoptant la technique de clustering et la modélisation thématique pour modéliser le texte à résumer avec une représentation numérique adéquate afin d'améliorer la performance de l'apprentissage automatique des modèles proposés.

Table A.0.1 Les principales approches existantes pour le RAT en Arabe

Référence	**Approche**	**Techniques utilisée**	**Jeux de test**	**Evaluation**
Douzidia and Lapalme (2004)	Numérique	Position de phrase, fréquence des termes, mots du titre et mots de repère	Corpus DUC-2004 traduit en Arabe	ROUGE
Al-Sanie (2005)	Symbolique	RST	Corpus développé par l'auteur	Précision
Sobh et al. (2007)	Numérique	Apprentissage automatique: classification naïve bayésienne, programmation génétique	Corpus développé par les auteurs	Rappel, précision, and F1-measure
Fattah and Ren (2009)	Numérique	Apprentissage automatique: Réseaux de neurones, probabilistes, réseaux de neurones à propagation avant, Modèle de mélange Gaussian Régression mathématique, Programmation génétique	Corpus développé par les auteurs et le corpus DUC-2001 traduit en Arabe	ROUGE-1, Rappel et précision,
Boudabous et al. (2010)	Numérique	Apprentissage automatique avec les machines à vecteurs de support (SVM)	Corpus développé par les auteurs	F1-measure Rappel, précision
El-Haj et al. (2011a)	Numérique	Requête et Concept	Corpus développé par les auteurs	Evaluation manuelle
El-Haj et al. (2011b)	Numérique	Clustering	Corpus DUC-2002 traduit en Arabe	Précision, rappel, ROUGE-1
Azmi and Al-Thanyyan (2012)	Hybride	Caractéristiques statistique, RST	Corpus développé par les auteurs	ROUGE, rappel, précision, et F1-measure
Haboush et al. (2012)	Numérique	Fréquence des racines des mots	Corpus développé par les auteurs	Rappel et précision
Ibrahim and Elghazaly (2013)	Hybride	RST SVM	Corpus développé par les auteurs	Précision
El-Fishawy et al. (2014)	Numérique	Similarité entre les tweets. Apprentissage automatique: Arbre de décision avec la régression linéaire	Corpus développé par les auteurs	F1-measure, Normalized Discounted Cumulative Gain
Oufaida et al. (2014)	Numérique	Minimal-redundancy maximal-relevance (mRMR)	EASC TAC 2011 MultiLing Pilot	ROUGE-1 et ROUGE-2
Belkebir and Guessoum (2015)	Numérique	Apprentissage automatique: SVM et AdaBoost	Corpus développé par les auteurs	F1-measure

A.3 Méthodes proposées

A.3.1 Méthode proposée basée sur les graphes pour le résumé automatique des textes Arabes.

Dans le chapitre 2, nous avons présenté une nouvelle méthode basée sur la théorie des graphes pour le résumé automatique des textes Arabes. Cette méthode adresse deux principaux défis qui se posent dans le contexte de la langue Arabe : i) la prise en considération des relations sémantiques entre les unités textuelles d'un document; ii) et l'élimination de la redondance pour diversifier les informations à inclure dans le résumé finale. Pour cela, nous proposons une nouvelle méthode qui utilise un graphe bidimensionnel (avec deux arcs) pour représenter à la fois les relations statistiques et sémantiques existantes dans les textes à résumer. Comme illustré dans la figure A.1, notre méthode est constituée de plusieurs phases :

Phase de prétraitement: Le but de cette phase est de préparer le texte original pour les étapes ultérieures. Cela consiste à segmenter le document en phrases et les phrases en mots, supprimer les mots vides afin de réduire la taille du document et enfin extraire les racines des mots.

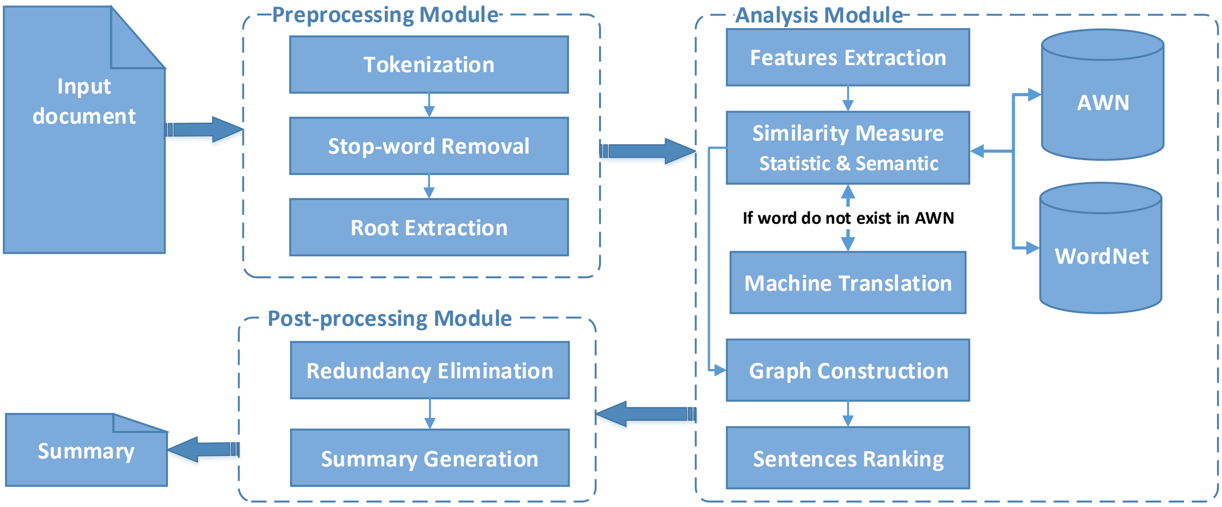
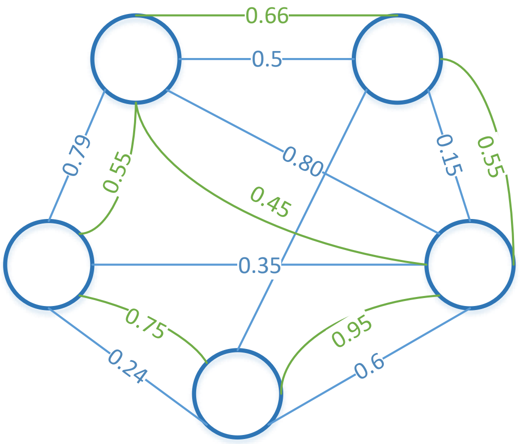


Figure A.1 Architecture de la méthode proposée basée sur l'analyse statistique et sémantique.



Chaque document Arabe est représenté sous forme de graphe à deux dimensions :

- Chaque phrase est représentée par un nœud
- Les relations entre deux phrases sont représentées par deux arcs reliant ces phrases
- L'arc bleu représente la similarité statistique entre deux phrases
- L'arc vert représente la similarité sémantique entre les deux phrases

Figure A.2 Graphe à bidimensionnel pour la représentation des documents Arabe

Phase d'analyse: Dans cette phase, nous extrayons d'abord les caractéristiques TF.IDF des phrases, après nous calculons la similarité statistique entre chaque paire de phrases en utilisant la mesure cosinus. Ensuite, nous calculons la similarité sémantique à l'aide des informations sémantiques extraites de la base de connaissances lexicales WordNet Arabe (ou AWN) dont l'utilisation permet à notre système d'appliquer un raisonnement en mesurant la distance sémantique entre des concepts réels développés manuellement. La section 2.3.3.3 du chapitre 2 présente en détail comment calculer cette mesure. Pour ce faire, nous procédons en quatre étapes :

a. Projection de chaque mot dans AWN pour extraire les concepts correspondants. AWN fournit pour chaque mot une liste de concepts classés du plus approprié au concept le moins approprié.

b. Application d'une stratégie de désambiguïsation pour assigner à chaque mot un seul et unique concept. Nous choisissons simplement le premier concept extrait de AWN.

c. Calcul de la similarité sémantique entre deux concept X et Y en utilisant la mesure de Wu and Palmer (1994) avec l'équation (A.1)

$$sim(X,Y) = \frac{2*N}{N1+N2} \tag{A.1}$$

Avec : N1 est la distance entre le concept X et le nœud racine de AWN. N2 est la distance entre le concept Y et le nœud racine. N est la distance entre le premier concept en commun entre le concept X et Y et le nœud racine dans la hiérarchie AWN.

d. Calcul de la similarité sémantique entre chaque paire de phrases en utilisant la mesure proposée par Malik et al. (2007). Pour cela, chaque phrase est représentée par un vecteur de concepts représentant ses mots, puis la similarité sémantique entre chaque paire de concepts (donc de mots) associés aux deux phrases en question est calculée par l'équation (A.1). Ensuite la similarité entre deux phrases est calculée par l'équation suivante :

$$Sim(S_i, S_j) = \frac{\sum_{w\in S_i} \max Sim_w(w, S_j) + \sum_{w\in S_j} \max Sim_w(w, S_i)}{|S_i|+|S_j|} \tag{A.2}$$

Dans cette équation, S_i et S_j sont les phrases en question; $\max Sim_w(w, S_j)$ représente le score maximal de similarité entre le mot w et tous les mots de la phrase S_j; et $|S_i|$ est la taille de la phrase S_i.

Phase de modélisation du texte sous forme de graphe: Nous procédons à la représentation des textes à résumer sous forme de graphe bidimensionnel. Comme le montre la figure A.1, les nœuds du graphe représentent les phrases du texte et chaque arc reliant deux nœuds représente la relation de similarité entre les deux phrases correspondantes. Dans notre travail, nous avons adopté deux types de relations: statistique et sémantique. Donc deux types d'arcs sont utilisés. Le premier représente la similarité statistique entre deux phrases calculée à l'aide de la mesure cosinus et le deuxième représente la similarité sémantique calculée en utilisant l'équation A.2.

Phase de classement des phrases: L'algorithme de classement PageRank est exécuté sur le graphe pour générer deux types de scores, $PR_{static}(N_i)$ et $PR_{semantic}(N_i)$ significatifs pour chaque phrase du texte.

Le score final de chaque phrase est obtenu par l'addition des deux scores selon la formule (A.3)

$$\mathrm{PR}(\mathrm{N_i}) = \mathrm{PR_{static}}(\mathrm{N_i}) + \mathrm{PR_{semantic}}(\mathrm{N_i}) \tag{A.3}$$

Phase de reclassement et élimination de la redondance: Le résumé final est formé en appliquant une technique de reclassement dans le but d'éliminer la redondance des informations et améliorer la qualité du résumé. Notre technique de reclassement repose sur une version adaptée de l'algorithme Marginal Maximal Relevance (MMR). Le principe est de n'inclure que les phrases qui sont à la fois pertinentes et nouvelles. Ainsi, La phrase qui a le score le plus élevé et qui a une faible similarité avec les phrases déjà sélectionnées sera considérée.

Pour évaluer notre méthode, nous avons mené plusieurs expérimentations sur deux corpus différents : EASC et notre propre corpus. Comme mesures d'évaluation, nous avons utilisé le rappel, précision, F1-mesure et Rouge-1. Le tableau A.2 présente les résultats obtenus par la mesure Rouge-1 en comparaison avec les principales méthodes de l'état de l'art en utilisant un pourcentage de compression de 30%. Nous pouvons remarquer l'efficacité de notre méthode par rapport aux autres. Les détails des expérimentations ainsi que les résultats obtenus sont présentés à la section 2.4 du chapitre 2.

Table A.0.2 Comparaison des résultats expérimentaux du système proposé et des systèmes de l'état de l'art sur les deux corpus en utilisant Rouge-1

Méthodes	Rouge-1 avec EASC	Rouge-1 avec notre corpus
Méthode proposée avec utilisation du MMR	0.4875	0.5765
Méthode proposée sans utilisation du MMR	0.4011	0.4906
ATSG (Alami et al., 2015)	0.4753	0.5734
ATSG sans MMR (Alami et al., 2015)	0.3678	0.4841
TextRank (Mihalcea and Tarau, 2004)	0.3892	0.4518
LexRank (Erkan and Radev, 2004)	0.3093	0.3841
TF.ISF	0.3759	0.4032

A.3.2 Méthode proposée basée sur l'apprentissage profond pour le résumé automatique des textes Arabes

Dans la section 3.3 du chapitre 3, nous avons proposé une nouvelle méthode de RAT Arabes basée sur l'apprentissage profond. Au moment de l'écriture de cette thèse, nous avons remarqué que les algorithmes d'apprentissage profond n'ont jamais été utilisés dans le cadre du RAT Arabes. Ces algorithmes ont montré leur puissance et efficacité dans plusieurs domaines d'application à savoir, la vision par ordinateur (Wang et al., 2016; Donahue et al., 2017; Kahou et al., 2015; Li et al., 2017b), le traitement du son (Lin et al., 2016; Li, Wang and Kot, 2017; Sun et al., 2017; Spille et al, 2018), et récemment le traitement automatique du langage naturel (Er et al., 2016; Li et al., 2017a; Ayinde et al., 2017; Firat et al., 2017; Yousefi-Azar and Hamey, 2017).

Après investigation des travaux existants sur le RAT Arabes, nous avons constaté qu'ils s'appuient sur l'approche sac-de-mots (BOW). Ces types d'approches provoquent deux problèmes majeurs. Premièrement, les textes sont représentés dans un espace de vocabulaire très large. Ce qui signifie que la matrice représentant le texte est creuse et compote beaucoup de valeurs nulles. Deuxièmement, les relations sémantiques entre les unités textuelles sont ignorées, puisque cette représentation implique des données d'entrée éparses et creuses représentées dans un espace de grandes dimensions. Par conséquent, la réduction de la dimensionnalité est nécessaire pour accroître la puissance d'extraction des caractéristiques pertinentes.

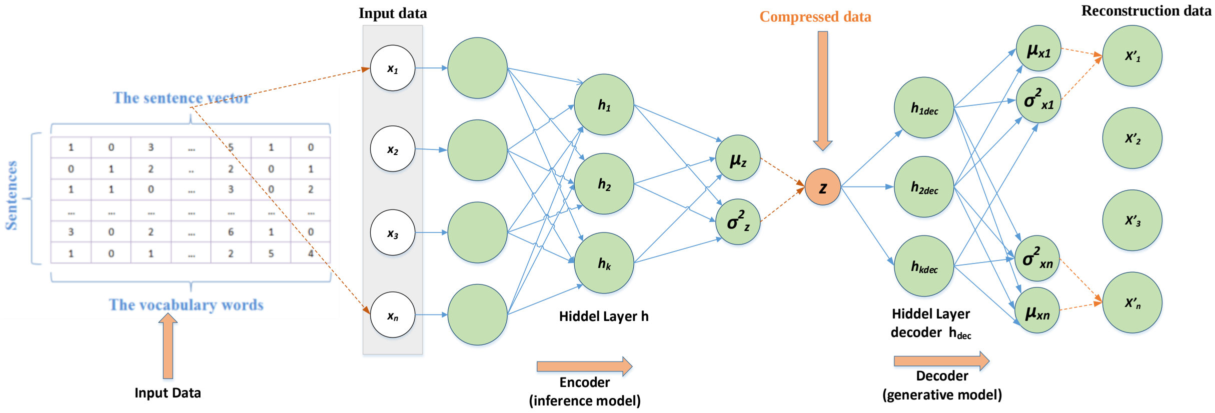


Figure A.3 Architecture de notre VAE pour la réduction de la dimensionnalité**.** A gauche, la représentation matricielle d'un document est générée et transmise dans l'entrée du VAE sous forme de couche visible pour être projetée dans un espace conceptuel z

Pour remédier à ces problèmes, nous avons adopté le variational auto-encoder (VAE) qui est un modèle d'apprentissage profond non supervisé permettant d'apprendre et de représenter les caractéristiques dans un espace de faible dimensions. Le VAE (Figure A.3) est utilisé pour projeter les phrases d'un texte dans un espace de faible dimension dans le but de construire une représentation abstraite pour chaque phrase du texte dans un espace sémantique latent appris par le VAE. Notre méthode de résumé automatique utilise cette nouvelle représentation pour calculer la similarité entre les phrases afin de les classer et d'en extraire les plus pertinentes en utilisant deux techniques de résumé automatique. Une première technique basée sur le modèle graphique dont les nœuds représentent les phrases du texte et les arcs représentent la similarité entre chaque paire de phrases (le modèle graphique est représenté en détail dans le chapitre 2). Une deuxième technique permettant de classer les phrases du texte selon leur degré de similarité avec une requête. La section 3.3 du chapitre 3 présente en détail la méthode proposée et les contributions apportées.

Pour évaluer notre méthode, nous avons mené plusieurs expérimentations sur le corpus EASC et notre propre corpus en calculant la mesure d'évaluation rappel Rouge-1. Le tableau A.3 présente les résultats obtenus sur le corpus EASC en comparaison aux différentes méthodes de l'état de l'art. Dans ce tableau, notre modèle proposé désigné par Graphe_VAE_TFIDF_V1000 indique que notre VAE a été entrainé sur une matrice de représentation TD.IDF avec un vocabulaire de 1000 mots et utilise la méthode des graphes pour le classement des phrases. Le modèle désigné par Requête_VAE_TFIDF_V1000 indique que notre VAE a été entrainé sur une matrice de représentation TD.IDF avec un vocabulaire de 1000 mots et utilise la méthode basée sur la similarité par rapport à la requête introduite par l'utilisateur pour le classement des phrases. Les résultats montrent clairement que la méthode proposée basée sur la requête introduite par l'utilisateur a abouti à des meilleures performances par rapport aux autres méthodes. Le deuxième meilleur résultat est obtenu par notre méthode basée sur le modèle graphique.

Le tableau A.4 présente les résultats obtenus sur notre propre corpus en comparaison aux différentes méthodes de l'état de l'art. Dans ce tableau, le modèle Graphe_VAE_TFIDF_V500 indique que la méthode proposée est basée sur le modèle graphique pour le classement des phrases et utilise la matrice de représentation TF.IDF avec un vocabulaire de taille 500 pour l'entrainement de notre VAE. Le modèle Requête_VAE_TFIDF_V500 indique que notre méthode proposée est basée sur le la similarité avec la requête introduite par l'utilisateur pour le classement des phrases et utilise la matrice de représentation TF.IDF avec un vocabulaire de taille 500 pour l'entrainement de notre VAE. Les résultats montrent clairement que la méthode proposée basée sur la requête introduite par l'utilisateur a donnée de meilleure performance par rapport aux autres méthodes. Le deuxième meilleur résultat est obtenu par notre méthode basée sur le modèle graphique.

Table A.3 Comparaison en termes de Rouge-1 des résultats expérimentaux du système proposé et des systèmes de l'état de l'art obtenus sur le corpus EASC avec des résumés de différentes tailles

Méthodes	10%	20%	30%	40%
Graphe_VAE_TFIDF_V1000	0.110	0.282	0.402	**0.529**
Requête_VAE_TFIDF_V1000	**0.115**	**0.286**	**0.403**	0.526
Graphe_AE	0.079	0.212	0.321	0.430
LSA	0.104	0.255	0.360	0.431
TextRank	0.112	0.278	0.382	0.501
LexRank	0.082	0.211	0.309	0.411
Baseline Tf.ISF	0.106	0.269	0.379	0.503

Table A.4 Comparaison en termes de Rouge-1 des résultats expérimentaux du système proposé et des systèmes de l'état de l'art obtenus sur notre propre corpus avec des résumés de différentes tailles

Méthodes	10%	20%	30%	40%
Graphe_VAE_TFIDF_V500	**0.158**	0.335	0.459	0.559
Requête_VAE_TFIDF_V500	0.150	**0.364**	**0.483**	**0.608**
Graphe_AE	0.096	0.233	0.370	0.483
LSA	0.134	0.293	0.416	0.540
TextRank	0.156	0.331	0.452	0.566
LexRank	0.111	0.259	0.378	0.484
Baseline Tf.ISF	0.116	0.261	0.389	0.518

Table A.5 Comparaison entre le résumé de référence, le résumé généré par la méthode proposée et celui généré par les systèmes concurrents

Résumé généré manuellement (Résumé de référence)		
- كشفت دراسة حديثة أجريت في تايوان أن أنواعا معينة من جراحات إنقاص الوزن قد تضعف العظام وتسبب هشاشتها وتزيد مخاطر الإصابة بالكسور. - وتساعد جراحات إنقاص الوزن، مثل تحويل مسار المعدة، مرضى السمنة في فقدان الوزن من خلال تقليل كمية الطعام التي يمتصها الجسم. - وقال كبير الباحثين في الدراسة الدكتور كو تشين هوانغ -وهو من كلية الطب في جامعة تايوان الوطنية في تايبيه - إن الجسم يُحرم من الكثير من العناصر الغذائية عندما يخضع المرضى لمثل هذه الجراحات. - وذكروا أن بحثا سابقا أشار بالفعل إلى أن هذه الجراحات قد تزيد مخاطر كسور العظام. - وأضاف "ثانيا يمكن أن يساعد التعرض للشمس وممارسة التمارين الرياضية في حمايتهم من هشاشة العظام، وأخيرا يجب عليهم مزاولة بعض تمارين التوازن لتجنب السقوط".		
Résumé automatique généré avec la méthode propose et les méthodes concurrents		
Modèle	**Nombre de phrases en commun**	**Résumé généré par le système**
VAE	3	- كشفت دراسة حديثة أجريت في تايوان أن أنواعا معينة من جراحات إنقاص الوزن قد تضعف العظام وتسبب هشاشتها وتزيد مخاطر الإصابة بالكسور. - وقال كبير الباحثين في الدراسة الدكتور كو تشين هوانغ -وهو من كلية الطب في جامعة تايوان الوطنية في تايبيه- إن الجسم يُحرم من الكثير من العناصر الغذائية عندما يخضع المرضى لمثل هذه الجراحات. - وبشكل عام، وجد الباحثون أن المرضى الذين خضعوا للجراحة زادت مخاطر إصابتهم بالكسور بنسبة 21% خلال السنوات الخمس التالية للجراحة. - وأضاف "ثانيا يمكن أن يساعد التعرض للشمس وممارسة التمارين الرياضية في حمايتهم من هشاشة العظام، وأخيرا يجب عليهم مزاولة بعض تمارين التوازن لتجنب السقوط".
AE	2	- كشفت دراسة حديثة أجريت في تايوان أن أنواعا معينة من جراحات إنقاص الوزن قد تضعف العظام وتسبب هشاشتها وتزيد مخاطر الإصابة بالكسور. - وذكروا أن بحثا سابقا أشار بالفعل إلى أن هذه الجراحات قد تزيد مخاطر كسور العظام. - وبشكل عام، وجد الباحثون أن المرضى الذين خضعوا للجراحة زادت مخاطر إصابتهم بالكسور بنسبة 21% خلال السنوات الخمس التالية للجراحة. - وقال هوانغ إن جراحات علاج السمنة يمكن أن تقلل احتمالات الإصابة بأمراض مثل السكري من النوع الثاني وارتفاع ضغط الدم.
LSA	2	- كشفت دراسة حديثة أجريت في تايوان أن أنواعا معينة من جراحات إنقاص الوزن قد تضعف العظام وتسبب هشاشتها وتزيد مخاطر الإصابة بالكسور. - وقال كبير الباحثين في الدراسة الدكتور كو تشين هوانغ -وهو من كلية الطب في جامعة تايوان الوطنية في تايبيه- إن الجسم يُحرم من الكثير من العناصر الغذائية عندما يخضع المرضى لمثل هذه الجراحات. - وقال هوانغ إن "العناصر الغذائية التي يُحرم منها الجسم في الغالب هي فيتامين "د" والكالسيوم والتي ترتبط بالإصابة بـهشاشة العظام، وربما توجد آليات أخرى ترتبط بالإصابة بالكسور". - ومن خلال قاعدة بيانات التأمين الصحي الوطنية، تتبع الباحثون 2064 مريضا خضعوا لجراحات علاج السمنة في الفترة من 2001 إلى 2009، و5027 مريضا بالسمنة لم يخضعوا لهذه الجراحات.
TextRank	2	- كشفت دراسة حديثة أجريت في تايوان أن أنواعا معينة من جراحات إنقاص الوزن قد تضعف العظام وتسبب هشاشتها وتزيد مخاطر الإصابة بالكسور. - وقال كبير الباحثين في الدراسة الدكتور كو تشين هوانغ -وهو من كلية الطب في جامعة تايوان الوطنية في تايبيه- إن الجسم يُحرم من الكثير من العناصر الغذائية عندما يخضع المرضى لمثل هذه الجراحات. - وقال هوانغ إن "العناصر الغذائية التي يُحرم منها الجسم في الغالب هي فيتامين "د" والكالسيوم والتي ترتبط بالإصابة بـهشاشة العظام، وربما توجد آليات أخرى ترتبط بالإصابة بالكسور". - وكتب هوانغ وزملاؤه في دورية الطب، أنه خلال العقد الماضي زادت جراحات علاج السمنة -وهي تقنية تستخدم إما في تصغير حجم المعدة وإما تحويل مسار أجزاء من القناة الهضمية - سبعة أمثال.
LexRank	1	- وقال هوانغ إن "العناصر الغذائية التي يُحرم منها الجسم في الغالب هي فيتامين "د" والكالسيوم والتي ترتبط بالإصابة بـهشاشة العظام، وربما توجد آليات أخرى ترتبط بالإصابة بالكسور". - وبشكل عام، وجد الباحثون أن المرضى الذين خضعوا للجراحة زادت مخاطر إصابتهم بالكسور بنسبة 21% خلال السنوات الخمس التالية للجراحة. - وقال هوانغ إن جراحات علاج السمنة يمكن أن تقلل احتمالات الإصابة بأمراض مثل السكري من النوع الثاني وارتفاع ضغط الدم. - وأضاف "ثانيا يمكن أن يساعد التعرض للشمس وممارسة التمارين الرياضية في حمايتهم من هشاشة العظام، وأخيرا يجب عليهم مزاولة بعض تمارين التوازن لتجنب السقوط".
Tf.ISF	1	- وقال هوانغ إن "العناصر الغذائية التي يُحرم منها الجسم في الغالب هي فيتامين "د" والكالسيوم والتي ترتبط بالإصابة بـهشاشة العظام، وربما توجد آليات أخرى ترتبط بالإصابة بالكسور". - وكتب هوانغ وزملاؤه في دورية الطب، أنه خلال العقد الماضي زادت جراحات علاج السمنة -وهي تقنية تستخدم إما في تصغير حجم المعدة وإما تحويل مسار أجزاء من القناة الهضمية - سبعة أمثال. - ومن خلال قاعدة بيانات التأمين الصحي الوطنية، تتبع الباحثون 2064 مريضا خضعوا لجراحات علاج السمنة في الفترة من 2001 إلى 2009، و5027 مريضا بالسمنة لم يخضعوا لهذه الجراحات. - وأضاف "ثانيا يمكن أن يساعد التعرض للشمس وممارسة التمارين الرياضية في حمايتهم من هشاشة العظام، وأخيرا يجب عليهم مزاولة بعض تمارين التوازن لتجنب السقوط".

Nous montrons dans le tableau A.5 un exemple des résumés extraits par notre méthode proposée et les systèmes concurrents. Le texte d'origine est donné dans la Figure 1.**3**. Il ressort clairement de cette comparaison que le résumé généré par notre VAE est plus proche du résumé généré par l'homme par rapport aux autres modèles. Notre méthode a 3 phrases en communes avec le résumé généré par l'homme, tandis que le meilleur résultat pour les concurrents est obtenu par les modèles AE, LSA et TextRank, qui ont deux phrases en commun avec le résumé généré par l'homme. Le mauvais résultat est obtenu par LexRank et Tf.ISF. La section 3.4 présente plus de détails sur les expériences menées et les résultats obtenus.

A.3.3 Méthodes proposées basées sur l'apprentissage ensembliste des réseaux de neurones et le plongement de mots

Dans la section précédente, nous avons présenté une nouvelle méthode de résumé automatique basée sur l'apprentissage profond en se basant sur le VAE pour apprendre les caractéristiques non supervisées des textes Arabes. Le modèle proposé a été entrainé en utilisant l'approche BOW générée à partir de la représentation TF.IDF du corpus d'entrainement. Dans cette section, nous proposons d'autres modèles pour améliorer la qualité du RAT Arabes en adoptant la représentation distribuée des mots pour l'entrainement de plusieurs modèles proposés basés sur les réseaux de neurones. En effet, il a été démontré que la représentation distribuée basée sur le plongement des mots (word embedding) surpasse celle par sac-de-mots en capturant les relations sémantiques dans le texte. Ainsi, pour améliorer le travail précédent, nous avons apporté les contributions suivantes :

- Nous avons exploré plusieurs modèles de réseaux de neurones. En plus du VAE étudié dans la section précédente, nous avons adopté les modèles suivants : i) la représentation distribuée des mots fournie par le plongement des mots ou le word embedding (WE); ii) l'Auto-Encoder (AE); iii) et la Machine d'apprentissage extrême Auto-Encoder (ELM-AE). Le détail de ces modèles est présenté dans la section 4.2 du chapitre 4.
- Nous avons construit notre propre modèle de plongement des mots Arabes (WE) par l'entrainement de l'algorithme word2vec sur un grand corpus de documents textuels.
- Nous avons entrainé les modèles des réseaux de neurones adoptés en utilisant la représentation distribuée WE des phrases (Phrase2Vec ou Sentence2vec) des textes au lieu du modèle classique BOW utilisé dans la contribution précédente. Elle est calculée à la base de la moyenne des vecteurs word2vec des mots constituants la phrase en question.
- Nous avons proposé de nouvelles méthodes basées sur l'apprentissage ensembliste (Ensemble learning) qui permet d'agréger les informations provenant de plusieurs modèles. Les figures A.4, A.5 et A.6 illustrent les grandes étapes de ces modèles.
- Nous avons évalué les approches proposées sur un corpus Anglais en plus de l'Arabe pour confirmer que ces approches améliorent aussi le résumé automatique des textes Anglais, en particulier, les approches basées sur l'apprentissage ensembliste.

Pour évaluer nos méthodes proposées, nous avons mené plusieurs expérimentations sur deux corpus. Le corpus Arabe EASC en calculant la mesure d'évaluation rappel Rouge-1, et le corpus Anglais SKE en calculant la mesure d'évaluation rappel Rouge-2. La raison pour laquelle nous avons utilisé Rouge-2 avec le corpus Anglais est de permettre la comparaison des résultats de nos modèles proposés avec ceux publiés par Youssef et al. (2017) qui utilise le même corpus d'évaluation SKE et la mesure Rouge-2. La méthode proposée par Youssef et al. (2017) a été comparé avec d'autres modèles basés sur l'apprentissage supervisés et non supervisés. Les auteurs ont constaté que leur approche dépassait les meilleurs systèmes non supervisés existants dans l'état de l'art, à savoir, le modèle basé sur les graphes (Hatori et al., 2011), MEAD (Radev et al., 2004) et ClueWordSummerizer (Ulrich et al., 2009). Pour les modèles supervisés, le modèle Ltf-ENAE (gaussien) de Youssef et al. (2017) surpasse les méthodes supervisées basées sur les techniques SVM, ME (lex) et BAG (lex-lc).

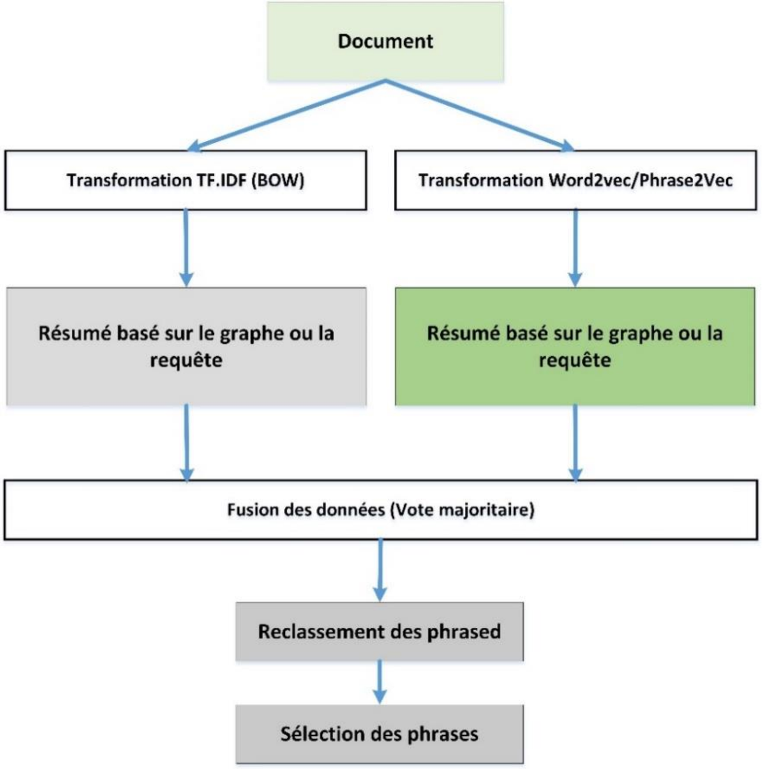

Document
Transformation TF.IDF (BOW)
Transformation Word2vec/Phrase2Vec
Résumé basé sur le graphe ou la requête
Résumé basé sur le graphe ou la requête
Fusion des données (Vote majoritaire)
Reclassement des phrased
Sélection des phrases

Figure A.4 Approche basée sur l'apprentissage ensembliste avec deux modèles de représentation des documents, BOW et word2vec. Le résultat final est obtenu en fusionnant les données provenant des deux modèles en utilisant la technique de vote majoritaire.

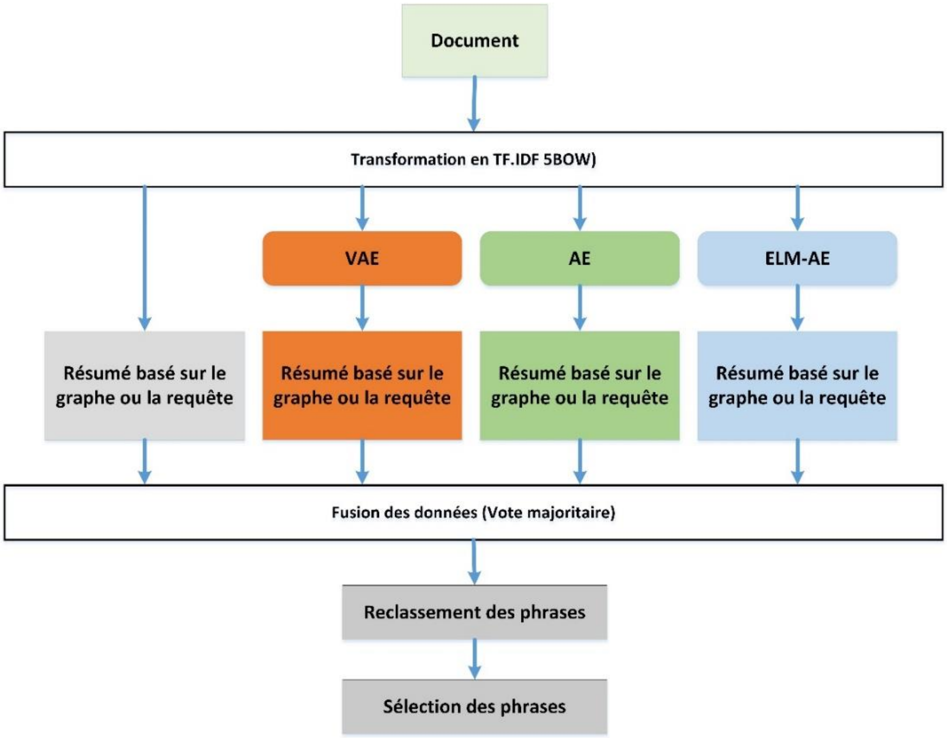

Document
Transformation en TF.IDF 5BOW)
VAE
AE
ELM-AE
Résumé basé sur le graphe ou la requête
Résumé basé sur le graphe ou la requête
Résumé basé sur le graphe ou la requête
Résumé basé sur le graphe ou la requête
Fusion des données (Vote majoritaire)
Reclassement des phrases
Sélection des phrases

Figure A.5 Approche basée sur l'apprentissage ensembliste avec quatre modèles utilisant la représentation BOW pour l'apprentissage des caractéristiques. Ici la matrice TF.IDF est utilisée comme entré pour l'entrainement des modèles. Le résultat final est obtenu en fusionnant les données provenant des quatre modèles en utilisant la technique de vote majoritaire

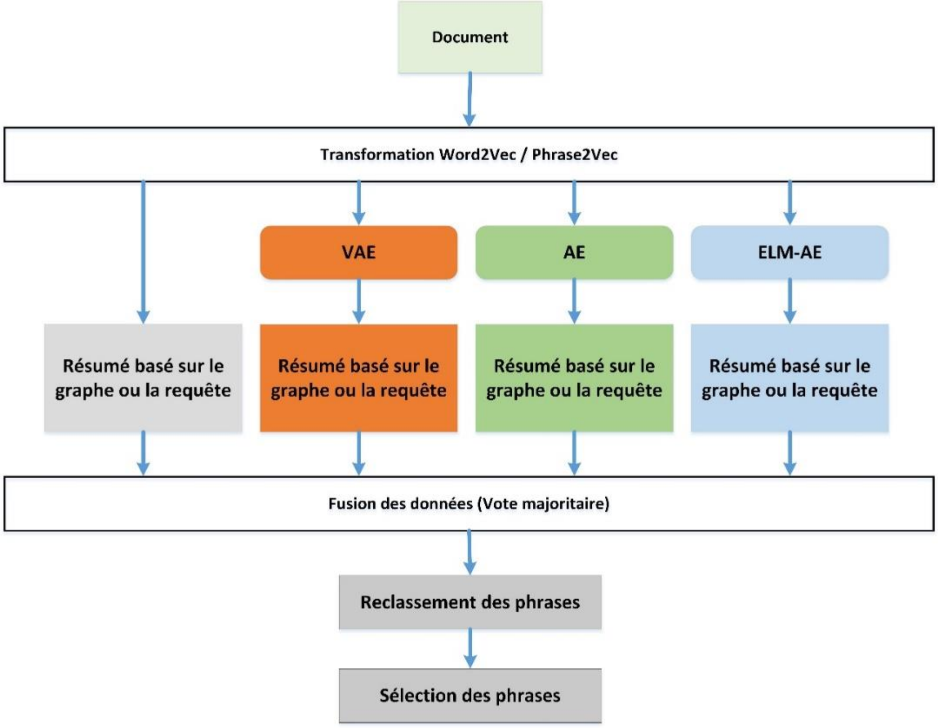

Document
Transformation Word2Vec / Phrase2Vec
VAE
AE
ELM-AE
Résumé basé sur le graphe ou la requête
Résumé basé sur le graphe ou la requête
Résumé basé sur le graphe ou la requête
Résumé basé sur le graphe ou la requête
Fusion des données (Vote majoritaire)
Reclassement des phrases
Sélection des phrases

Figure A.6 Approche basée sur l'apprentissage ensembliste avec quatre modèles utilisant la représentation distribuée word2vec comme entré pour l'entrainement des modèles. Le résultat final est obtenu en fusionnant les données provenant des quatre modèles en utilisant la technique de vote majoritaire

Le tableau A.6 présente les résultats obtenus sur le corpus EASC en termes de rappel Rouge-1 en comparaison aux principaux systèmes de l'état de l'art. Ces résultats montrent que notre modèle proposé Phrase2vec_AE_VAE_ELM-AE a abouti à des meilleures performances par rapport aux autres méthodes. Ce modèle, illustré dans la figure A.6, est basé sur l'approche ensembliste qui agrège les résultats de quatre modèles de réseaux de neurones qui utilisent la représentation distribuée des phrases Phrase2vec (ou sentence2vec) pour l'apprentissage des caractéristiques. La représentation Phrase2vec de chaque phrase a été générée en calculant la moyenne des vecteurs word2vec de tous les mots constituant la phrase en question.

Le meilleur résultat des systèmes concurrents est obtenu par notre méthode basée sur le VAE que nous avons proposé dans le chapitre 4. La mesure Rouge-1 obtenue par cette méthode est égale à 0,402 lorsque la taille du résumé est égale à 30%, alors que, selon nos expériences, le meilleur résultat Rouge-1 est égal à 0,5147 obtenu par notre modèle proposé Phrase2vec_AE_VAE_ELM-AE. Nous avons aussi remarqué que le résultat obtenu par tous les modèles proposés basés sur l'apprentissage ensembliste sont meilleurs que les autres systèmes de l'état de l'art. Ces résultats indiquent clairement que l'agrégation des informations provenant de plusieurs sources, permet au système de générer un résumé efficace et significatif.

Pour montrer l'efficacité de nos approches, nous avons également comparé notre meilleur modèle avec la méthode proposé par Youssef et al. (2017) (Ltf-ENAE (Gaussian)) qui a montré que son méthode a surpassé les méthodes de l'état de l'art. Le tableau A.7 montre que tous les modèles proposés basés sur la représentation Phrase2vec surpassent les méthodes de l'état de l'art. Dans ce tableau, *n* désigne le nombre de phrases à extraire. Les meilleures performances sont obtenues par notre modèle d'apprentissage ensembliste Phrase2vec_AE_VAE_ELM-AE. Par contre, le mauvais résultat est obtenu par l'apprentissage ensembliste basé sur la représentation BOW. Cela montre que l'approche BOW diminue les performances du RAT Anglais. Cependant, l'approche Word2Vec augmente les performances du système de résumé automatique, en particulier lorsqu'il est utilisé comme donnés d'entrés pour l'entrainement des modèles ensemblistes basés sur les réseaux de neurones. Le détail des expérimentations et les résultats obtenus sont présentés en section 4.3 du chapitre 4.

Table A.6 Comparaison des modèles proposés avec les systèmes existants en termes de rappel Rouge-1 sur le corpus EASC

Méthodes	Taille du résumé			
	10%	20%	30%	40%
BOW (TF-IDF)	0.0986	0.2537	0.3693	0.4885
LSA	0.1045	0.2559	0.3608	0.4312
TextRank	0.1197	0.2819	0.3892	0.5014
Modèle graphique_VAE (Alami et al., 2018)	0.1101	0.2825	0.4021	0.5298
Phrase2vec	0.1299	0.2924	0.3953	0.4969
Phrase2vec_AE	0.1024	0.2484	0.3515	0.4652
Phrase2vec_VAE	0.1117	0.2635	0.3705	0.4746
Phrase2vec_ELM-AE	0.1120	0.2662	0.3658	0.4746
Ensemble: BOW_S2V	0.3185	0.4305	0.5064	0.5798
Ensemble: BOW_AE_VAE_ELM-AE	0.2812	0.3752	0.4672	0.5579
Ensemble : Phrase2vec_AE_VAE_ELM-AE	0.3265	0.4444	0.5147	0.5910

Table A.7 Comparaison des modèles proposés avec les méthodes évaluées dans Youssef et al. (2017) en termes de rappel Rouge-2 sur le corpus SKE

Méthodes	Taille du résumé en nombre de phrases				
	n=1	n=2	n=3	n=4	n=5
Ltf-ENAE (Youssef et al., 2017)	0.1370	0.2471	0.3510	0.4325	0.5031
Phrase2Vec	0.1646	0.3012	0.4214	0.5136	0.5902
Ensemble: BOW_S2V	0.1556	0.2861	0.4083	0.4969	0.5771
Phrase2vec _VAE	0.1334	0.2591	0.3623	0.4575	0.5387
Phrase2vec_AE	0.1492	0.2812	0.4045	0.5122	0.5957
Phrase2vec_ELM-AE	0.1441	0.2671	0.3725	0.4677	0.5453
Ensemble BOW_AE_VAE_ELM-AE	0.1061	0.2064	0.3133	0.3989	0.4739
Ensemble Phrase2vec_AE_VAE_ELM-AE	0.1702	0.3074	0.4269	0.5235	0.6040

A.3.4 Amélioration des résumés automatiques des textes Arabes par la technique du clustering et la modélisation thématique

Dans la section 5.2 du chapitre 5, nous avons proposé d'améliorer les performances des modèles étudiés dans le chapitre 4 par l'utilisation des techniques de partitionnement des données (clustering), la modélisation thématique et les réseaux de neurones non supervisés.

La représentation des documents est une étape très importante pour toute application de TALN. Afin d'appliquer un traitement automatique via des algorithmes d'apprentissage automatique ou des techniques statistiques, les documents doivent être convertis en valeurs numériques ou en représentations vectorielles. Cette représentation numérique doit comporter les caractéristiques les plus significatives du texte. Les techniques les plus connues sont le calcul de la matrice de fréquence de mots (TF) et la méthode TF.IDF. Ces dernières années, des nouvelles techniques ont été découvertes pour réaliser cette transformation en utilisant l'approche de plongement de mots (word embedding) que nous avons adopté dans le chapitre 4.

Dans cet section, nous proposons de nouvelles approches utilisant les techniques de partitionnement des données (clustering), la modélisation thématique et les réseaux de neurones non supervisés pour construire un modèle efficace de représentation des textes Arabes pour le résumé automatique. L'architecture générale de ces approches est illustrée dans la figure A.7 et montre que le RAT est effectué en plusieurs étapes :

Tout d'abord, après la phase de prétraitement qui consiste à normaliser, segmenter et supprimer les mots vides, nous avons procédé au regroupement des textes en plusieurs clusters. Pour cela, nous avons proposé une nouvelle méthode de clustering des textes Arabes en utilisant la machine d'apprentissage extrême avec l'algorithme k-mean. Nous avons appliqué cette méthode sur une grande collection de documents textuels Arabes pour apprendre la tâche de clustering et regrouper chaque document dans le cluster adéquat. La figure A.8 illustre cette nouvelle méthode de clustering. Puis, nous avons appliqué la modélisation thématique en utilisant la méthode d'Allocation de Dirichlet Latente (ou LDA) (Blei et al., 2003). Cette méthode a été appliquée sur chaque groupe de documents pour identifier les sujets associés à chaque cluster. Ensuite, nous avons représenté chaque document dans l'espace des sujets (ou thèmes) déterminé dans l'étape précédente selon le cluster auquel le document appartient, puisque pour chaque document nous avons identifié à quel cluster il appartient, donc l'ensemble des sujets qui le représentent. Cette représentation est modélisée sous forme d'une matrice où les lignes représentent les phrases du document et les colonnes représentent les sujets du cluster. Dans cette contribution, nous avons adopté deux types de représentations basées sur l'espace des sujets :

- **Sent2Topic_prob:** Un texte est représenté par une matrice dans laquelle les lignes représentent les phrases et les colonnes représentent les thèmes (ou sujets) du cluster auquel appartient le texte. Chaque ligne est exprimée sous forme de vecteur dont la valeur de chaque colonne représente la probabilité du sujet étant donné l'ensemble des mots constituant la phrase en question.

  Soit une phrase $S = \{w_1, w_2, \dots, w_n\}$, chaque mot w_i peut être exprimé par un vecteur dans l'espace des sujets noté $L(w_i) = (P(z_1|w_i), P(z_2|w_i), \dots, P(z_k|w_i))$ avec k est le nombre de sujets, et $P(z_i|w_j)$ est la probabilité d'assigner le sujet z_i au mot w_j. Le vecteur représentant la phrase S dans l'espace des sujets est exprimé avec la formule suivante:

  $$L(S) = L(w_1, w_2, \dots, w_n) = \left(\frac{\sum_{i=1}^{n} P(z_1|w_i)}{n}, \frac{\sum_{i=1}^{n} P(z_2|w_i)}{n}, \dots, \frac{\sum_{i=1}^{n} P(z_k|w_i)}{n}\right) \quad \text{(A.4)}$$

  La section 5.2.4 du chapitre 5 donne plus de détail sur le calcul de cette probabilité.

- **Sent2Topic_w2v:** C'est le deuxième modèle de représentation que nous avons construit à partir de l'espace de sujet identifié. La différence entre ce modèle et le précédent (Sent2Topic_prob) réside dans le fait que les valeurs dans la matrice sont calculées en fonction de la similarité sémantique entre la phrase donnée et chaque sujet. La similarité sémantique est calculée en utilisant le modèle word2vec que nous avons construit à partir d'une grande collection de documents Arabe (voir section 4.3.1 du chapitre 4).

  Soit une phrase $S = \{w_1, w_2, \dots, w_n\}$, et $T = \{topic_1, \dots, topic_k\}$ l'ensemble des sujets identifiés pour le document en question. Chaque sujet $topic_i$ est composé d'un ensemble de mots $topic_i = \{terme_{i,1}, \dots, terme_{i,m}\}$.
  Pour calcule la similarité entre la phrase S et le sujet $topic_i$, nous avons calculé d'abord le vecteur moyen pour tous les mots de la phrase S et le vecteur moyen de tous les mots du sujet $topic_i$. Ensuite, nous avons utilisé la similarité cosinus entre les deux vecteurs.

Après, nous avons utilisé cette représentation comme données d'entrainement de différents modèles de réseaux de neurones non supervisés et d'apprentissage ensembliste (figures A.9 et A.10) afin de générer une représentation abstraite de chaque document textuel dans l'espace latente formé par ces modèles. Enfin, nous avons utilisé cette représentation abstraite pour calculer la similarité entre chaque paire de phrases afin de modéliser le texte à résumer sous forme graphique. Nous avons appliqué l'algorithme de classement PageRank sur le graphe généré pour calculer le score de chaque phrase du texte. Le résumé final est construit à partir des phrases les mieux classées tout en respectant la taille du résumé souhaitée et en supprimant les phrases redondantes et similaires. En effet, la redondance et la diversité des informations constituent un problème typique dans les systèmes de résumé automatique et en particulier ceux désignés aux textes Arabes. Pour cela, nous avons utilisé l'algorithme Maximal Marginal Relevance (MMR) pour éliminer les informations inutiles et améliorer la qualité du résumé final.

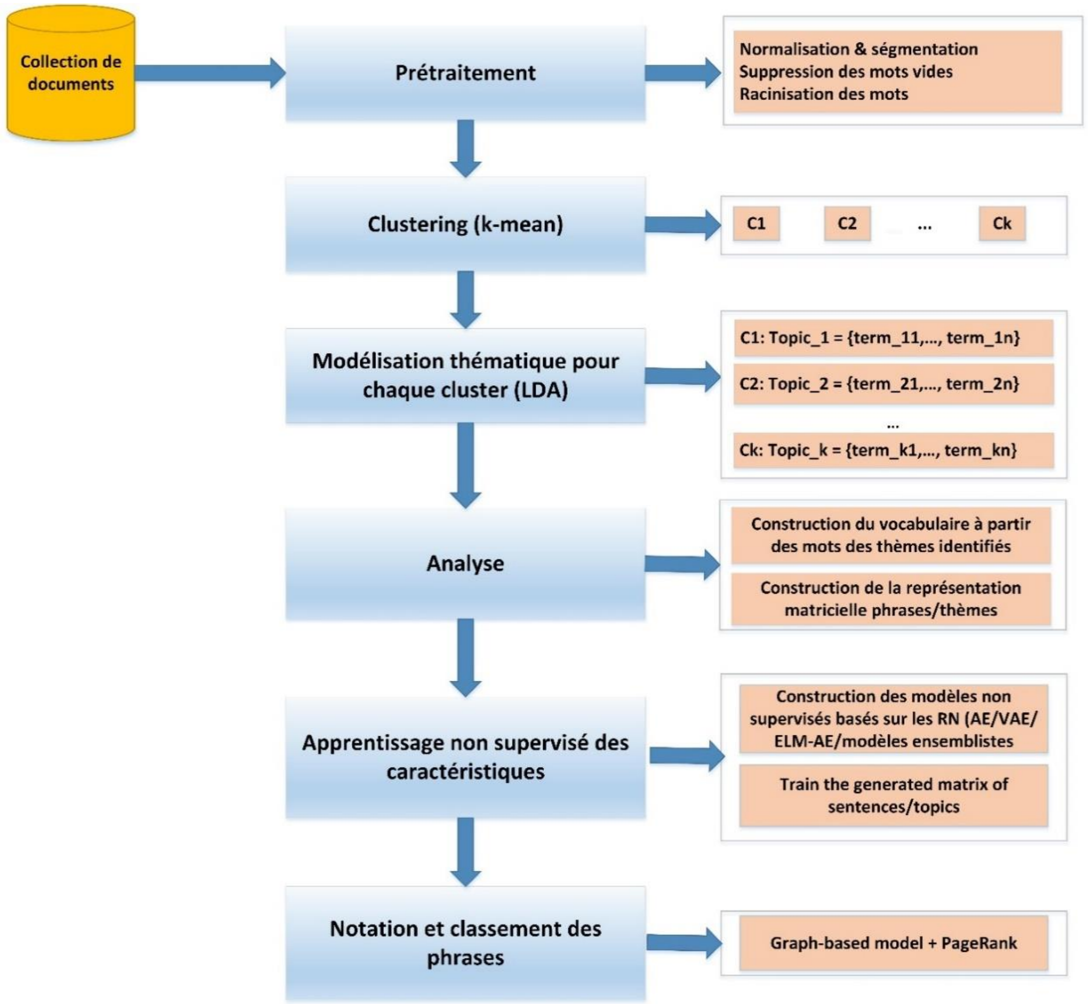

Collection de documents
Prétraitement
Normalisation & ségmentation
Suppression des mots vides
Racinisation des mots
Clustering (k-mean)
C1
C2
...
Ck
Modélisation thématique pour chaque cluster (LDA)
C1: Topic_1 = {term_11,..., term_1n}
C2: Topic_2 = {term_21,..., term_2n}
...
Ck: Topic_k = {term_k1,..., term_kn}
Analyse
Construction du vocabulaire à partir des mots des thèmes identifiés
Construction de la représentation matricielle phrases/thèmes
Apprentissage non supervisé des caractéristiques
Construction des modèles non supervisés basés sur les RN (AE/VAE/ELM-AE/modèles ensemblistes
Train the generated matrix of sentences/topics
Notation et classement des phrases
Graph-based model + PageRank

Figure A.7 Les différentes étapes de l'approche proposée pour le RAT Arabes

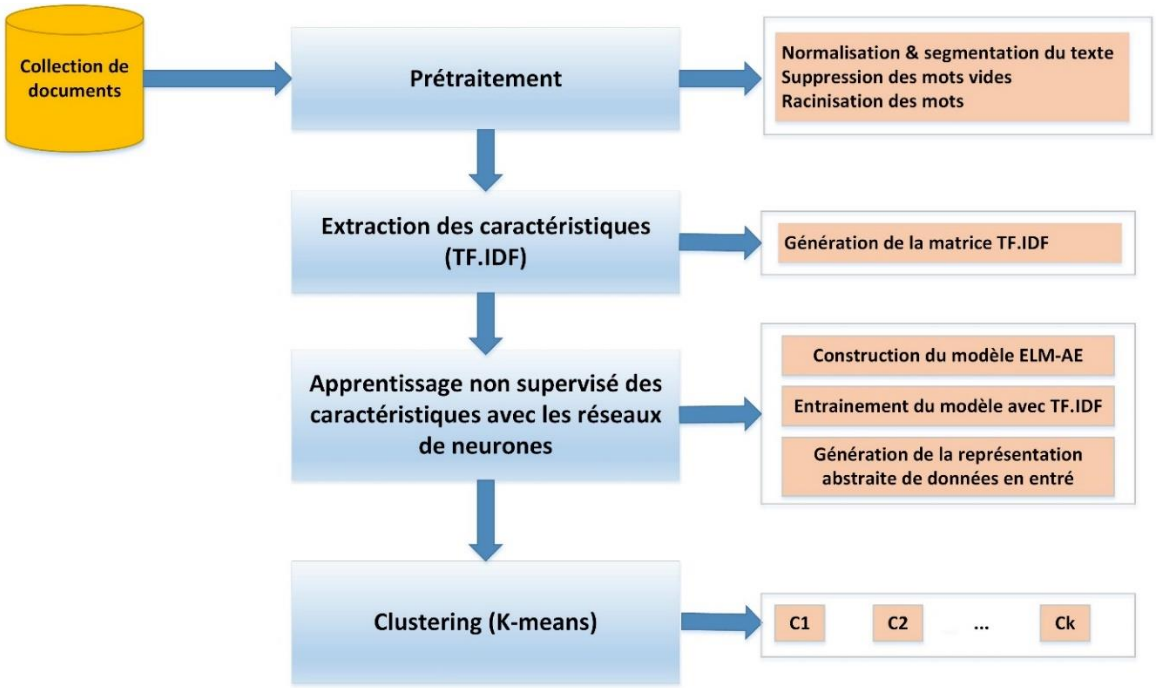

Collection de documents
Prétraitement
Normalisation & segmentation du texte
Suppression des mots vides
Racinisation des mots
Extraction des caractéristiques (TF.IDF)
Génération de la matrice TF.IDF
Apprentissage non supervisé des caractéristiques avec les réseaux de neurones
Construction du modèle ELM-AE
Entrainement du modèle avec TF.IDF
Génération de la représentation abstraite de données en entré
Clustering (K-means)
C1
C2
...
Ck

Figure A.8 Architecture de notre méthode de clustering des documents Arabes pour le résumé automatique

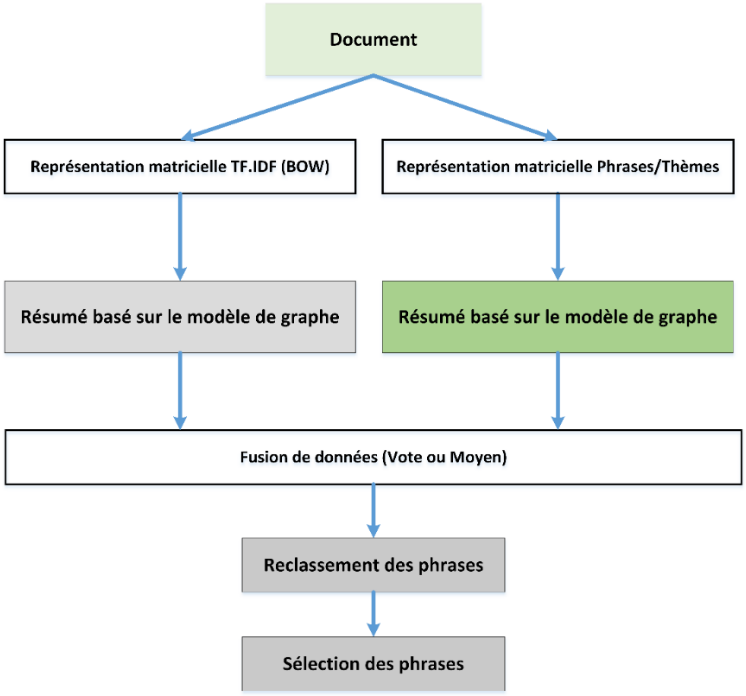

Document
Représentation matricielle TF.IDF (BOW)
Représentation matricielle Phrases/Thèmes
Résumé basé sur le modèle de graphe
Résumé basé sur le modèle de graphe
Fusion de données (Vote ou Moyen)
Reclassement des phrases
Sélection des phrases

Figure A.9 Approche basée sur l'apprentissage ensembliste de deux modèles. Le premier modèle est basé sur la représentation classique BOW. Le deuxième modèle est basé sur la représentation obtenue à partir de l'espace des sujets (Sent2Topic_prob ou Sent2Topic_w2v)

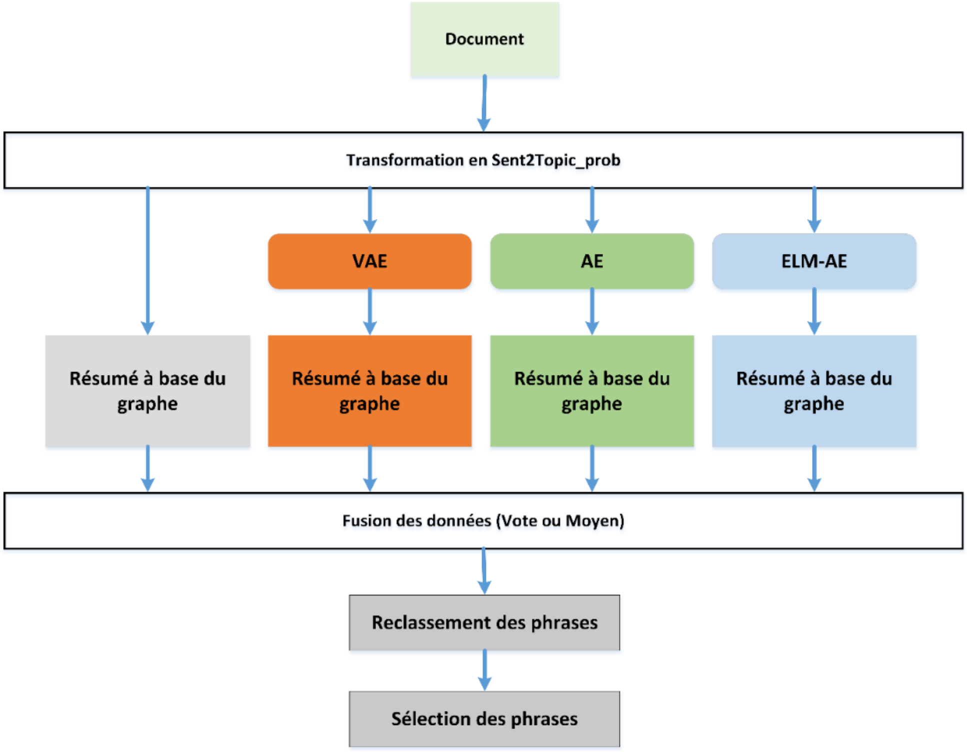


Figure A.10 Approche basée sur l'apprentissage ensembliste de quatre modèles. Le premier modèle est basé sur la représentation obtenue à partir de l'espace des sujets (Sent2Topic_prob). Le deuxième, troisième et quatrième modèle sont basés respectivement sur le VAE, AE et l'ELM-AE. Ici, l'apprentissage des réseaux de neurones est effectué par les données de représentation Sent2Topic_prob

Pour évaluer nos approches proposées, nous avons mené plusieurs expérimentations sur le corpus EASC. Nous avons utilisé la mesure Roue-1 et Rouge-2 pour évaluer les différentes approches proposées et les comparer avec les systèmes de l'état de l'art.

Table A.8 Comparaison en termes de rappel Rouge-1 entre l'approche propose avec les méthodes de l'état de l'art sur le corpus EASC corpus avec des résumés de différentes tailles

	Taux de compression			
Méthodes	10%	20%	30%	40%
Approche proposée avec MMR	0.4688	0.5702	0.6543	0.7121
Approche proposée sans MMR	0.5487	0.6404	0.6966	0.7631
VAE_Graphe (Alami et al., 2018)	0.1101	0.2825	0.4021	0.5298
VAE _Requête (Alami et al., 2018)	0.115	0.286	0.403	0.526
LSA	0.1045	0.2559	0.3608	0.4312
TextRank	0.1197	0.2819	0.3892	0.5014
Baseline Tf.ISF	0.106	0.269	0.379	0.503

Le tableau A.8 établit une comparaison entre notre méthode proposée (Ensemble NN_Sent2Topic_prob) illustrée dans la figure A.10 et ses concurrents en termes de rappel Rouge-1. La méthode proposée NN_Sent2Topic_prob repose sur les trois modèles de réseaux de neurones (AE, VAE et ELM-AE) entrainés sur la représentation Sent2Topic_prob. Nous pouvons facilement constater que notre méthode a la valeur la plus élevée du score Rouge-1 et surpasse tous les autres systèmes dans les deux cas : sans et avec élimination de la redondance. Par exemple, Avec un taux de compression de 30%, le meilleur résultat de Rouge-1 obtenu par les autres systèmes a été rapporté par l'approche VAE basée sur la requête (Alami et al., 2018) avec 0,403. Le deuxième bon résultat a été rapporté par le VAE basée sur le modèle des graphes proposé par Alami et al. (2018) avec 0,4021 de rappel Rouge-1 lorsque a taille du résumé est 30%. Alors que dans notre expérience, le score de rappel Rouge-1 obtenu par la méthode proposée est de 0,6966 avec l'utilisation de la technique MMR et de 0,6543 sans utilisation de MMR. En termes de rappel Rouge-1, la performance de l'approche proposée est augmentée de 0,2936 avec l'utilisation du MMR et de 0,2513 sans l'utilisation du MMR, ce qui revient à dire que notre algorithme donne les meilleurs résultats par rapport à l'état de the-art et améliore la performance des systèmes de résumé Arabe.

D'après les résultats présentés dans le tableau A.8, il est évident que notre méthode surpasse toutes les autres méthodes, car notre approche est capable de repérer les relations entre les phrases à l'aide de leur projection dans l'espace des sujets appris par l'algorithme LDA. Ces relations ne peuvent pas être identifiées par les systèmes de référence utilisés dans l'expérimentation. En plus, ces résultats indiquent clairement que lorsque les informations proviennent de plusieurs sources (modèles différents) et agrégées via une technique d'apprentissage ensembliste, le système génère un résumé efficace et significatif.

Les deux tableaux A.9 et A.10 présentent une comparaison entre nos approches et celle proposées par Al-Radaideh and Bataineh (2018), Al-Khawaldeh and Samawi (2015) (LCEAS), Oufaida et al. (2014) et Al-Omour (2012). Une première comparaison de ces méthodes a été présentée par Al-Radaideh and Bataineh (2018). L'évaluation a été faite sur la base des métriques Rouge-1 et Rouge-2 en utilisant le corpus EASC et en générant des résumés de tailles 25% et 40%. Après l'analyse des résultats obtenus, nous avons conclus que les approches proposées dépassent les méthodes comparées en termes de rappel Rouge-1 et rappel Rouge-2 pour les deux cas 25% et 40% de la taille du résumé.

Table A.9 Comparaison en termes de Rouge-1 des approches proposées avec celles de l'état de l'art sur le corpus EASC avec un taux de compression de 25% et 40%

	CR=25%		CR=40%	
Méthodes	Rappel	F1-mesure	Rappel	F1-mesure
Modèle proposé VAE_Sent2Topic_w2v	0.6170	**0.5433**	0.7388	**0.5452**
Modèles proposés basés sur l'apprentissage ensembliste avec la technique du vote majoritaire				
Ensemble BOW_Sent2Topic_prob	0.6719	0.2115	0.7596	0.1674
Ensemble BOW_Sent2Topic_w2v	0.6658	0.2113	**0.7636**	0.1679
Ensemble NN_Sent2Topic_prob	**0.6758**	0.2104	**0.7631**	0.1669
Ensemble NN_Sent2Topic_w2v	0.6634	0.2104	0.7705	0.1698
Modèles proposés basés sur l'apprentissage ensembliste avec la technique du moyen				
Ensemble BOW_Sent2Topic_prob	0.6081	0.5037	0.7348	0.5124
Ensemble BOW_Sent2Topic_w2v	0.5756	0.4986	0.7138	0.5141
Ensemble NN_Sent2Topic_prob	0.5890	0.4938	0.7140	0.4986
Ensemble NN_Sent2Topic_w2v	0.5852	0.4949	0.7178	0.5032
Compétiteurs				
Al-Radaideh and Bataineh (2018)	0.395	0.476	0.588	0.542
Oufaida et al. (2014)	0.420	0.370	-	-
Al-Omour (2012)	0.324	0.411	0.449	0.485

Table A.10 Comparaison en termes de Rouge-2 des approches proposées avec celles de l'état de l'art sur le corpus EASC avec un taux de compression de 25% et 40%

	CR=25%		CR=40%	
Méthodes	Rappel	F1-mesure	Rappel	F1-mesure
Modèle proposé VAE_Sentence2Topic_w2v	0.5026	**0.4435**	0.6310	**0.4617**
Modèles proposés basés sur l'apprentissage ensembliste avec la technique du moyen				
Ensemble BOW_Sent2Topic_prob	0.4589	0.3844	0.5992	0.4207
Ensemble BOW_Sent2Topic_w2v	0.4229	0.3702	0.5738	0.4150
Ensemble NN_Sent2Topic_prob	0.4403	0.3740	0.5718	0.4028
Ensemble NN_Sent2Topic_w2v	0.4361	0.3744	0.5767	0.4067
Compétiteurs				
Al-Radaideh and Bataineh (2018)	0.334	0.372	0.465	0.422
Oufaida et al. (2014)	-	-	0.290	0.260
Al-Khawaldeh and Samawi (2015)	-	-	0.270	0.28

Comme le montre le tableau A.9, en termes de métrique F1-mesure Rouge-1, le meilleur résultat a été obtenu par notre modèle VAE_Sent2Topic_w2v qui a dépassé toutes les systèmes de l'état de l’art. Les modèles basés sur l’approche ensembliste avec la technique de vote majoritaire ont données le mauvais score. Par contre les modèles basés sur l’approche ensembliste avec la technique du moyen ont surpassé tous les systèmes compétiteurs lorsque la taille du résumé est de 25%, mais lorsque la taille est de 40%, ces modèles ont donnés un F1-mesure compétitif.

Le tableau A.10 montre qu’en termes de métrique F1-mesure Rouge-2, les modèles proposés basés sur l’approche ensembliste avec la technique du moyen ont surpassé tous les systèmes compétiteurs lorsque la taille du résumé est de 25%. Ces modèles ont donnés un F1-mesure compétitif lorsque la taille du résumé est de 40%. Le meilleur résultat en termes de F1-mesure de Rouge-2 est obtenu par notre modèle VAE_Sent2Topic_w2v pour les deux tailles de résumé 25% et 40%. Les détails concernant les expérimentations menées ainsi que les résultats obtenus sont présentés à la section 5.3 du chapitre 5.

A.4 Conclusion générale

Dans ce travail de thèse, nous nous sommes intéressés à l’étude et l’amélioration des méthodes de résumé automatique des textes Arabes. Ces améliorations touchent plusieurs aspects, en particulier, i) la représentation des textes sous forme numérique ou vectorielle pour perfectionner la tâche d’apprentissage, ii) le classement et la notation des unités textuelles et iii) la construction du résumé finale. En effet, nous avons proposé un ensemble de méthodes extractives pour le résumé automatique des textes Arabes permettant aux grand nombre d’utilisateurs de cette langue d’accéder facilement et rapidement aux informations les plus pertinentes d’un document textuel de grande taille. Nos méthodes proposées peuvent être aussi utilisées pour améliorer d'autres tâches de traitement automatique du langage naturel telles que, le partitionnement (clustering), la classification, l'indexation et l'extraction des mots-clés, etc.

En se basant sur l’état de l’art établie dans le chapitre 1, plusieurs problèmes ont été identifiés et résolus. Tout d’abord, nous avons conclu que les travaux de recherche menés dans ce domaine ont connu une forte progression, en particulier pour l’anglais. Cependant, les recherches sur le résumé automatique des textes Arabe sont rares et ne sont pas du même niveau de maturité à ceux établies pour les autres langues occidentales et en particulier l’anglais. Deuxièmement, nous avons conclu que la plupart des méthodes existantes pour le résumé Arabe ne tiennent pas en considération les relations sémantiques entre les unités textuelles (mots, phrases, paragraphes, etc.). De ce fait, le résumé final comportera des informations redondantes et d’autres idées pertinentes ne seront pas incluses vu la taille limite du résumé à produire. Troisièmement, nous avons constaté que la plupart des travaux existants sont basés sur une approche classique de représentation des textes en sac de mots (Bag-of-Words ou BOW). Cette approche implique des données d'entrée éparses, creux et de grande dimension, en plus, elle ignore les relations sémantiques entre les unités textuelles du document. Par conséquent, l’identification des informations pertinentes dans le texte à résumer devient une tâche complexe. Quatrièmement, les méthodes traditionnelles de résumé automatique des textes Arabes ne prennent pas en considération le contexte abordé par les textes traités (sujets ou thèmes du document). Ces sujets, s’ils sont identifiés et pris en compte par le système, peuvent être utiles pour extraire les informations pertinentes du texte original, et par conséquent, améliorer la qualité du résumé produit.

Compte tenu de ces limitations et lacunes, nous avons proposé plusieurs contributions dans ce travail de thèse afin d'améliorer la tâche de résumé automatique des textes en langue Arabe.

Notre première contribution est présentée dans le chapitre 2, dans lequel nous avons proposé une nouvelle méthode de résumé automatique des textes Arabes basée sur les graphes et utilise une structure hiérarchique d'ontologie pour fournir une mesure de similarité plus précise entre les unités textuelles. La méthode proposée est basée sur un modèle graphique bidimensionnel qui combine l'analyse statistique et l'analyse sémantique. Ainsi, le document est représenté par un graphe à deux dimensions dans lequel chaque nœud représente une phrase du texte, et les nœuds sont reliés par deux arcs. Le premier arc représente la similarité statistique entre les deux phrases et le deuxième arc représente la similarité sémantique. La similarité statistique est basée sur le chevauchement du contenu entre deux phrases, tandis que la similarité sémantique est calculée à l'aide des informations sémantiques extraites de la base de connaissances lexicale AWN dont l'utilisation permet à notre système d'appliquer un raisonnement en mesurant la distance sémantique entre des concepts réels développés par l'humain. Nous avons exécuté l'algorithme de classement PageRank sur le graphe pour générer deux types de scores significatifs pour chaque phrase du texte à résumer. De plus, nous avons abordé le problème de la redondance et la diversité des informations dans le résumé final en utilisant une version adaptée de l'algorithme Maximal Marginal Relevance (MMR). Les résultats expérimentaux sur EASC et sur notre propre jeu de données ont montré l'efficacité de l'approche proposée par rapport aux méthodes de l'état de l'art. En plus, l'utilisation de la méthode MMR pour la suppression des phrases redondantes a amélioré significativement la qualité du résumé final. Nous avons également étudié l'effet du processus de prétraitement (stemming) sur la tâche de résumé automatique des textes Arabes. Ce processus est utilisé dans de nombreuses applications de feuille de textes en tant que technique d'extraction des caractéristiques. Pour cela, nous avons évalué l'impact de trois stemmers distingues utilisés spécifiquement pour la langue Arabe (Khoja, Larekey et Alkhalil) sur la qualité du résumé généré. L'évaluation du système proposé sur l'ensemble des données de teste, avec les trois stemmers et sans utilisation du stemming, montre que les meilleures performances ont été obtenues par le stemmer Khoja en termes de rappel, précision et F1-mesure. L'évaluation montre également que la performance de la méthode proposée est considérablement améliorée par l'application du processus du stemming dans la phase du prétraitement.

Dans le chapitre 3, nous avons détaillé notre deuxième contribution, qui consiste à adopter un algorithme d'apprentissage profond non supervisé pour le résumé automatique des textes Arabes. Nous avons proposé d'utiliser le modèle Variationnal Auto-Encoder (VAE) pour apprendre un nouvel espace de concept à partir de données d'entrée de grande dimension. Nous avons exploré plusieurs représentations comme données d'entrée pour l'entrainement de notre VAE, telles que la fréquence de terme (TF), TF.IDF en utilisant un vocabulaire local (mots du document) et global (mots du corpus entier). Toutes les phrases sont classées en fonction de la représentation latente produite par le VAE. Nous avons étudié les résultats obtenus par le VAE sur deux approches de résumé automatique: une approche basée sur un modèle de graphe et une approche basée sur la similarité par rapport à une requête. Les expériences menées sur deux jeux de données de référence spécifiquement conçus pour cette tâche ont montré clairement que le VAE utilisant la représentation TF.IDF avec un vocabulaire global a fourni un espace de caractéristiques plus discriminant et améliore le score des autres modèles utilisés. Les résultats des expériences confirment aussi que la méthode proposée, que ce soit en utilisant la technique des graphes ou celle de requête, conduit à de meilleures performances en comparaison avec les méthodes de résumé par extraction de l’état de l’art.

Le chapitre 4 a été consacré à notre troisième contribution qui consiste à adopter plusieurs modèles de réseaux de neurones pour améliorer la tâche de résumé automatique des textes Arabes. Les réseaux de neurones ont prouvé leur capacité à obtenir d'excellents résultats dans de nombreuses applications de traitement du langage naturel et de vision par ordinateur. Cependant, ces techniques non pas été étudiées dans le cadre du résumé automatique des textes Arabes. Le but de ces approches est de découvrir et apprendre un espace de faible dimension représentant la structure latente (ou abstraite) d’une grande collection de données textuelles représentées dans un espace à grande dimension. Dans cette contribution, nous avons proposé des approches de résumé automatique des textes Arabes en se basant sur trois modèles non supervisés : l’auto-encoder (AE), le variationnal auto-encoder (VAE), et la machine d’apprentissage extrême (ELM). Nous avons proposé ici d’autres améliorations du modèle basé sur le VAE qui a été présenté dans le chapitre 3. En revanche, le plongement de mots (ou le Word Embedding) est une autre technique de réseau de neurones qui génère une représentation distribuée des mots plus compacte qu’une approche classique en sac de mots (BOW). L'objectif de ce chapitre est d'améliorer la qualité du résumé automatique en combinant cette représentation distribuée avec les modèles de réseaux de neurones non supervisées et les modèles d’apprentissage ensembliste. Premièrement, nous avons construit notre modèle de plongement de mots arabes par l’entrainement d’une grande collection de documents textuel avec la méthode word2vec. Deuxièmement, nous avons montré que le résumé de texte basé sur le modèle Word2vec donne de meilleurs résultats par rapport au modèle de représentation classique BOW. Troisièmement, nous avons proposé d'autres modèles en combinant le modèle Word2vec avec les méthodes d'apprentissage profond non supervisées. Pour cela, nous avons utilisé la représentation distribuée Word2vec comme données d’entrée pour l’entrainement de nos modèles AE, VAE et ELM. Les résultats ont montré que la qualité du résumé automatique est améliorée lorsque ces modèles utilisent la représentation Word2vec dans la phase d’apprentissage en comparaison avec les résultats obtenus avec les mêmes modèles utilisant la représentation BOW (TF.IDF) dans la phase d’apprentissage. Quatrièmement, nous avons également proposé trois techniques basées sur l’apprentissage ensembliste. Le premier modèle ensembliste combine BOW et word2vec en utilisant la technique du vote majoritaire. Le deuxième modèle agrège les informations fournies par l’approche BOW et les réseaux de neurones non supervisés (ici l’AE, VAE et ELM). Le troisième modèle regroupe les informations fournies par la représentation Word2vec et les réseaux de neurones non supervisés. Nous avons montré que les méthodes d'apprentissage ensembliste améliorent significativement la qualité des résumés Arabes. En particulier, le modèle ensembliste basé sur les réseaux de neurones et l'approche Word2vec pour l’apprentissage des caractéristiques a donné les meilleurs résultats. Enfin, nous avons effectué différentes expériences pour évaluer les performances des méthodes proposées. Nous avons utilisé deux types de jeux de données disponibles publiquement pour évaluer la tâche de résumé automatique des documents en anglais et en arabe. Les résultats obtenus par les études statistiques confirment que les modèles basés sur le plongement de mots (word embedding) sont plus efficaces que les modèles basés sur l'approche BOW. En particulier, la représentation Word2vec utilisée avec la technique d'apprentissage ensembliste surpasse tous les modèles étudiés.

Dans le chapitre 5, nous avons détaillé notre quatrième contribution, qui consiste à améliorer les résultats précédents en cherchant la meilleure manière pour représenter les textes Arabes sous forme numérique (ou matricielle) pour un apprentissage plus efficace des caractéristiques. À cet égard, nous avons proposé de nouvelles approches pour résumer les textes Arabes en utilisant les techniques de clustering, la modélisation thématique et les réseaux de neurones. La tâche de résumé est effectuée en quatre étapes principales. Tout d'abord, une nouvelle méthode pour le clustering des documents Arabes a été proposée en utilisant la machine d'apprentissage extrême. Cette nouvelle méthode est appliquée sur une grande collection de documents afin d'apprendre la tache de clustering et de regrouper chaque document dans un cluster spécifique. Deuxièmement, nous avons utilisé la modélisation thématique avec l'algorithme d'Allocation de Dirichlet Latente (LDA). Nous avons appliqué LDA sur chaque groupe de documents appartenant à clusters identifiés dans l'étape précédente. Le but est d'identifié l'espace des thèmes (ou des sujets) associé à chaque cluster. Ainsi, pour chaque document dans notre corpus d'apprentissage, nous avons identifié son cluster ou il appartient et son espace de sujets associé à son cluster. Troisièmement, nous avons construit une représentation de chaque document dans son espace de sujets par une matrice où les lignes représentent les phrases du texte et les colonnes représentent les sujets du cluster. La matrice générée est ensuite utilisée pour entrainer les différents modèles de réseaux de neurones non supervisés et les modèles d'apprentissage ensembliste afin de générer une représentation abstraite des données d'entrées dans l'espace de concept. La matrice résultante dans l'espace du concept consiste en une nouvelle représentation latente du texte original. Nous avons utilisé cette nouvelle représentation pour modéliser le texte à résumer sous forme de graphe. L'algorithme de classement PageRank est ensuite exécuté sur le graphe pour générer un score significatif pour toutes les phrases du texte. Nous avons amélioré les approches proposées par l'utilisation de l'algorithme MMR pour la suppression de la redondance et la diversification des informations à inclure dans le résumé final. Les résultats expérimentaux sur EASC ont montré l'efficacité des approches proposées par rapport aux méthodes de l'état de l'art.

Les contributions apportées et les résultats prometteurs découverts dans ce travail de thèse ouvrent plusieurs directions pour des futures opportunités de recherche dans le domaine de résumé automatique des textes Arabes :

- La première direction consiste à élargir le corpus utilisé dans le processus d'évaluation en développant un corpus de documents plus large avec leurs résumés manuels. Cela donnera plus de valeur lors de l'évaluation des approches proposées.

- Une deuxième direction consiste à examiner les caractéristiques linguistiques spécifiques à la langue Arabe, telles que la catégorie grammaticale du discoure, la coréférence et la résolution de l'anaphore, qui sont des champs de recherche ouverts en traitement automatique de la langue Arabe. Les caractéristiques sémantiques peuvent également être incorporées à l'aide d'utilisation d'autres ressources de connaissance, telles que Wikipedia Arabe en plus de WordNet Arabe qui ne constitue pas une solution complète pour calculer la similarité sémantique entre les unités textuelles en raison de la limite des concepts couvertes dans cette base de connaissance.

- Nous avons aussi l'intention d'évaluer les performances des approches proposées dans un domaine spécifique, telles que le résumé des commentaires en ligne, les textes biomédicaux et les tweets en ligne. Alors que les données en ligne (tweets en arabe, critiques et commentaires sur les hôtels, etc.) sont largement disponibles sur Internet, l'entrainement et le test des approches proposées sur ces données peuvent améliorer la performance du résumé automatique appliqué à un domaine spécifique.

- Nous avons l'intention d'intégrer d'autres modèles d'apprentissage profond et des réseaux de neurones non supervisés, tels que les auto-encodeurs empilés, l'auto-encodeur d'attention, la machine de Boltzmann restreinte et la version non supervisée du réseau de neurones à convolution. De plus, les approches supervisées peuvent aider à améliorer la tâche de résumé des textes Arabe. Le problème avec les approches supervisées c'est qu'elles ont besoin d'un grand corpus annoté pour l'apprentissage automatique. Nous pouvons résoudre ce problème en utilisant des documents extraits depuis les sites Web les plus connus et en générant des résumés humains pour chaque document.

- Au moment de la rédaction de ce rapport de thèse, les approches abstractives pour le résumé automatique des textes Arabes n'ont pas été abordées. Le résumé abstrait consiste à comprendre les concepts principaux du document original et les présenter dans un document plus court avec une forme différente. Cela nécessite beaucoup de connaissances développé par l'humain, des méthodes statistiques et des traitements linguistiques. La catégorie de résumé abordée dans ce travail de thèse est extractive, ce qui implique l'utilisation des outils basiques de traitement automatique de la langue pour générer le résumé final. Le traitement automatique de l'Arabe souffre d'un manque de ressources linguistique et ontologiques matures et des outils puissants pour la génération du langage naturel. Il est donc difficile pour les chercheurs en langue Arabe d'aborder profondément ce type d'approche. Par conséquent, nous pouvons commencer à nous attaquer au domaine du résumé abstractif en développant dans un premier temps des outils et des ressources capables de générer une séquence correcte de phrases. Parmi ces outils, nous illustrons des lexiques arabes, des ontologies, des connaissances linguistiques développées manuellement et des modèles de langage.

Publications of the Author

The research works carried out for the completion of this thesis work have resulted in several publications in national conferences, international conferences, book chapter, and specialized journals. We present these publications next, sorting them by publication type.

International journals (#4)

1. Alami, N., Meknassi, M., & Rais, N. (2015). Automatic Texts Summarization: Current State of the Art. Journal of Asian Scientific Research, 5(1), 1–15. https://doi.org/10.18488/journal.2/2015.5.1/2.1.1.15
2. Alami, N., En-nahnahi, N., Ouatik, S. A., & Meknassi, M. (2018). Using Unsupervised Deep Learning for Automatic Summarization of Arabic Documents. Arabian Journal for Science and Engineering, 43(12), 7803–7815. doi:10.1007/s13369-018-3198-y
3. Alami, N., Meknassi, M., & En-nahnahi, N. (2019). Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning. Expert Systems with Applications, 123, 195–211. doi:10.1016/j.eswa.2019.01.037
4. Alami, N., Meknassi, M., En-nahnahi, N., Ouatik, S. A., Ammor, O., El Adlouni, Y. Enhancing Neural Networks-Based Arabic Text Summarization by Document Clustering and Topic modeling. Cognitive Computation. Submitted to the journal on 28 October 2018 (Under review)

Book chapter (#1)

1. Alami, N., El Adlouni, Y., En-nahnahi, N., & Meknassi, M. (2018). Using Statistical and Semantic Analysis for Arabic Text Summarization. In: Noreddine G., Kacprzyk J. (eds) International Conference on Information Technology and Communication Systems. ITCS 2017. Advances in Intelligent Systems and Computing, vol. 640, pp. 35–50, https://doi.org/10.1007/978-3-319-64719-7_4, Springer, Cham.

International conferences and workshops (#3)

1. Alami, N., Meknassi, M., Alaoui Ouatik, S., & Ennahnahi, N. (2015). Arabic text summarization based on graph theory. 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA). doi:10.1109/aiccsa.2015.7507254
2. Alami, N., Meknassi, M., Ouatik, S. A., & Ennahnahi, N. (2016). Impact of stemming on Arabic text summarization. 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt). doi:10.1109/cist.2016.7805067
3. Alami, N., El Adlouni, Y., En-nahnahi, N., & Meknassi, M. (2017) Using Statistical and Semantic Analysis for Arabic Text Summarization. In: Proceeding of the International Conference on Information Technology and Communication Systems. ITCS 2017. ENSA Khouribga.

National conferences (#1)

1. Alami, N., En-nahnahi, N., & Meknassi, M. (2016). Apprentissage profond pour le résumé automatique des textes Arabe. In: Journées scientifiques nationales en Signal, Image, Multimédia et Applications (SIGMA). December 21-22, 2016, Fez, Morocco.

Bibliography

Abuobieda, A., Salim, N., Kumar, YJ., & Osman, AH. (2013). An improved evolutionary algorithm for extractive text summarization. In: Intelligent information and database systems, Springer, pp 78–89

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for boltzmann machines. Cognitive Science, 9(1), 147–169. doi:10.1016/s0364-0213(85)80012-4

Ahmadi, N., & Akbarizadeh, G. (2017). Hybrid robust iris recognition approach using iris image pre-processing, two-dimensional gabor features and multi-layer perceptron neural network/PSO. IET Biometrics. doi:10.1049/iet-bmt.2017.0041

Akbarizadeh, G., & Moghaddam, A.E. (2016). Detection of Lung Nodules in CT Scans Based on Unsupervised Feature Learning and Fuzzy Inference. Journal of Medical Imaging and Health Informatics 6(2):477–483. doi:10.1166/jmihi.2016.1720

Akbarizadeh, G., Tirandaz, Z., & Kooshesh, M. (2014). A New Curvelet Based Texture Classification Approach For Land Cover Recognition of SAR Satellite Images. Malaysian Journal of Computer Science 27(3):218-239.

Akbarizadeh, G. (2013). Segmentation of SAR Satellite Images Using Cellular Learning Automata and Adaptive Chains. Journal of Remote Sensing Technology 44–51. doi:10.18005/jrst0102003

Alami, N., En-nahnahi, N., Ouatik, S. A., & Meknassi, M. (2018). Using Unsupervised Deep Learning for Automatic Summarization of Arabic Documents. Arabian Journal for Science and Engineering, 43(12), 7803–7815. doi:10.1007/s13369-018-3198-y

Alami, N., Meknassi, M., Alaoui Ouatik, S., & Ennahnahi, N. (2015). Arabic text summarization based on graph theory. 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA). doi:10.1109/aiccsa.2015.7507254

Alami, N., Meknassi, M., Ouatik, S. A., & Ennahnahi, N. (2016). Impact of stemming on Arabic text summarization. 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt). doi:10.1109/cist.2016.7805067

Al-Anzi, F. S., & AbuZeina, D. (2015). Stemming impact on Arabic text categorization performance: A survey. 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA), Marrakech, 2015, pp. 1-7.

Alguliyev, R. M., Aliguliyev, R. M., & Isazade, N. R. (2015). An unsupervised approach to generating generic summaries of documents. Applied Soft Computing, 34, 236–250. doi:10.1016/j.asoc.2015.04.050

Aljlayl, M., & Frieder, O. (2002). On Arabic search: improving the retrieval effectiveness via a light stemming approach. Proceedings of the Eleventh International Conference on Information and Knowledge Management - CIKM ’02, McLean, VA, USA. PP 340-347. doi:10.1145/584845.584848

Al-Khawaldeh, F., & Samawi, V. (2015). Lexical cohesion and entailment based segmentation for Arabic text summarization (LCEAS). World of Computer Science & Information Technology Journal, 5(03), 51–60.

Al-Omour, M. (2012). Extractive-based Arabic text summarization approach. M.Sc Thesis: Department of Computer Science, Yarmouk University, Irbid, Jordan.

Alqudsi, A., Omar, N., & Shaker, K. (2012). Arabic machine translation: a survey. Artificial Intelligence Review, 42(4), 549–572. doi:10.1007/s10462-012-9351-1

Al-Radaideh, Q.A., & Bataineh, D.Q. (2018). A Hybrid Approach for Arabic Text Summarization Using Domain Knowledge and Genetic Algorithms. Cognitive Computation, 10(4), pp.651–669. Available at: http://dx.doi.org/10.1007/s12559-018-9547-z.

Al-Saleh, A. B., & Menai, M. E. B. (2015). Automatic Arabic text summarization: a survey. Artificial Intelligence Review, 45(2), 203–234. doi:10.1007/s10462-015-9442-x

Al-Sanie, W. (2005). Towards an infrastructure for Arabic text summarization using rhetorical structure theory. Master's thesis, King Saud University, Riyadh

Andrushia, A. D., & Thangarajan, R. (2015). Visual attention-based leukocyte image segmentation using extreme learning machine. International Journal of Advanced Intelligence Paradigms, 7(2), 172. doi:10.1504/ijaip.2015.070771

Aone, C., Okurowski, M. E., & Gorlinsky, J. (1998). Trainable, scalable summarization using robust NLP and machine learning. Proceedings of the 36th Annual Meeting on Association for Computational Linguistics -. doi:10.3115/980845.980856

Atkinson, J., & Munoz, R. (2013). Rhetorics-based multi-document summarization. Expert Systems with Applications, 40 (11), 4346–4352.

Atwan, J., Mohd, M., Kanaan, G., & Bsoul, Q. (2014). Impact of Stemmer on Arabic Text Retrieval. Lecture Notes in Computer Science, 314–326. doi:10.1007/978-3-319-12844-3_27

Ayinde, B. O., & Zurada, J. M. (2017). Deep Learning of Constrained Autoencoders for Enhanced Understanding of Data. IEEE Transactions on Neural Networks and Learning Systems 99:1–11. doi:10.1109/tnnls.2017.2747861

Azmi, A., & Al-thanyyan Suha. (2009). Ikhtasir — A user selected compression ratio Arabic text summarization system. 2009 International Conference on Natural Language Processing and Knowledge Engineering, pp. 1-7. doi:10.1109/nlpke.2009.5313732

Azmi, A.M., & Al-Thanyyan, S. (2012). A text summarizer for Arabic. Computer Speech and Language 26 (4), 260–273.

Baccour, L. (2004). Conception et réalisation d'un système de segmentation de textes arabes non voyellés. PhD thesis, Faculté des Sciences Économiques et de Gestion de Sfax.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). Modern Information Retrieval, vol. 463. ACM, New York.

Baralis, E., Cagliero, L., Mahoto, N., & Fiori, A. (2013). GraphSum: Discovering correlations among multiple terms for graph-based summarization. Information Sciences, 249, 96–109. doi:10.1016/j.ins.2013.06.046

Barzilay, R., & Elhadad, M. (1997). Using Lexical Chains for Text Summarization. In: Proceedings of the ACL/EACL'97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain, pp. 10–17.

Barzilay, R., & Lee, L. (2004). Catching the drift: Probabilistic content models, with applications to generation and summarization. In Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, pp. 113-120

Barzilay, R., & McKeown, K. R. (2005). Sentence Fusion for Multidocument News Summarization. Computational Linguistics, 31(3), 297–328. doi:10.1162/089120105774321091

Batcha, N. K., Aziz, N. A., & Shafie, S. I. (2013). CRF based Feature Extraction Applied for Supervised Automatic Text Summarization. Procedia Technology, 11, 426–436. doi:10.1016/j.protcy.2013.12.212

Boudlal, A., Lakhouaja, A., Mazroui, A., Meziane, A., Ould Abdallahi Ould Bebah, M., & Shoul, M., (2010). Alkhalil Morpho SYS1: a morphosyntactic analysis system for arabic texts. In: International Arab Conference on Information Technology. Benghazi, Libya, pp. 1–6.

Belguith, L., Aloulou, C. & Ben Hamadou, A. (2008). MASPAR : De la segmentation à l'analyse syntaxique de textes arabes. In CÉPADUÈS-Editions, editeur, Revue Information Interaction Intelligence I3, volume 7, pages 9 – 36, http ://www.revue-i3.org/, mai 2008. 2008. ISSN : 1630-649x.

Belguith, L., Baccour, L., & Mourad G. (2005). Segmentation de textes arabes basées sur l'analyse contextuelle des signes de ponctuations et de certaines particules. In Actes de la 12ème Conférence annuelle sur le Traitement Automatique des Langues Naturelles, Dourdan France, vol. 1, pp. 451–456.

Belkebir, R., & Guessoum, A. (2015). A supervised approach to arabic text summarization using adaboost. In: Rocha A, Correia AM, Costanzo S, Reis LP (eds) New contributions in information systems and technologies, advances in intelligent systems and computing, vol 353, pp 227–236.

Bellare, K., Sarma, A.D., & Loiwal, N. (2004). Generic Text Summarization using Word Net, International Conference on Language Resources and Evaluation, 2004.

Bengio, Y., Lamblin, P., Popovici, V., Larochelle, H. (2007). Greedy layer-wise training of deep networks. In: B. Schölkopf, J. Platt, and T. Hoffman, editors, Advances in Neural Information Processing Systems 19, MIT Press, Cambridge, MA, pp 153–160.

Bengio, Y., LeCun, Y.: Scaling learning algorithms towards ai. (2007). In: Large-Scale Kernel Machines. MIT Press.

Bengio, Y., Schwenk, H., Senécal, J.-S., Morin, F., & Gauvain, J.-L. (2006). Neural Probabilistic Language Models. Studies in Fuzziness and Soft Computing, 137–186. doi:10.1007/10985687_6

Bengio, Y.: Learning deep architectures for AI. (2009). Foundations and Trends in Machine Learning 2(1):1–127.

Berger, A. L., & Mittal, V. O. (2000). OCELOT: a system for summarizing web pages. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR, pp 144–151.

Blei, D. M., Ng, A.Y., & Jordan, M.I. (2003). Latent Dirichlet allocation. Journal of Machine Learning Research, vol. 3, no. 5, pp. 993-1022.

Boudabous, M., Maaloul, M., & Belguith, L. (2010). Digital learning for summarizing arabic documents. In: Loftsson H, Rgnvaldsson E, Helgadttir S (eds) Advances in natural language processing, lecture notes in computer science, vol 6233. Springer, Berlin, pp 79–84.

Boudchiche, M., Mazroui, A., Ould Abdallahi Ould Bebah, M., Lakhouaja, A., & Boudlal, A. (2017). AlKhalil Morpho Sys 2: A robust Arabic morpho-syntactic analyzer. Journal of King Saud University - Computer and Information Sciences, 29(2), 141–146. doi:10.1016/j.jksuci.2016.05.002

Bouras, C., & Tsogkas, V. (2012). A clustering technique for news articles using WordNet. Knowledge-Based Systems, 36, 115–128.

Bouzoubaa, K., Baidouri, H., Loukili, T., & El Yazidi T. (2009). Arabic Stop Words: Towards a Generalisation and Standardisation. In the 13th International Business Information Management Association Conference IBIMA.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30(1-7), 107–117. doi:10.1016/s0169-7552(98)00110-x

Bsoul, Q., Al-Shamari, E., Mohd, M., & Atwan, J. (2014). Distance Measures and Stemming Impact on -319-3-doi:10.1007/978 .339–Arabic Document Clustering. Lecture Notes in Computer Science, 327 28_3-12844

Buckwalter, T. (2002). Arabic Morphological Analyzer Version 1.0. Linguist. Data Consort. N° LDC2002L49.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. Proceedings of the 22nd International Conference on Machine Learning - ICML '05, pp 89-96. doi:10.1145/1102351.1102363

Cabral, L. de S., Lins, R. D., Mello, R. F., Freitas, F., Ávila, B., Simske, S., & Riss, M. (2014). A platform for language independent summarization. Proceedings of the 2014 ACM Symposium on Document Engineering - DocEng '14, ACM, pp 203–206. doi:10.1145/2644866.2644890

Cao, J., Zhang, K., Luo, M., Yin, C., & Lai, X. (2016). Extreme learning machine and adaptive sparse representation for image classification. Neural Networks, 81, 91–102. doi:10.1016/j.neunet.2016.06.001

Cao, Z., Wei, F., Dong, L., Li, S., Zhou, M. (2015). Ranking with Recursive Neural Networks and its Application to Multi-Document Summarization. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, Texas, pp 2153-2159

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. Proceedings of the 21st Annual International ACM SIGIR

Conference on Research and Development in Information Retrieval - SIGIR '98. doi:10.1145/290941.291025

Chali, Y., Hasan, S. A., & Joty, S. R. (2009). A SVM-Based Ensemble Approach to Multi-Document Summarization. Lecture Notes in Computer Science, 199–202. doi:10.1007/978-3-642-01818-3_23

Chen, A., & Gey, F. C. (2002). Building an Arabic Stemmer for Information Retrieval. In Proceedings of the 11th Text Retrieval Conference (TREC 2002), National Institute of Standards and Technology.

Chen, A., & Gey, F. C. (2001). Translation Term Weighting and Combining Translation Resources in Cross-Language Retrieval. In TREC, 2001.

Chen, C.-L., Tseng, F. S. C., & Liang, T. (2010). An integration of WordNet and fuzzy association rule mining for multi-label document clustering. Data & Knowledge Engineering, 69(11), 1208–1226.

Chen, J., & Zhuge, H. (2014). Summarization of scientific documents by detecting common facts in citations. Future Generation Computer Systems, 32, 246–252. doi:10.1016/j.future.2013.07.018

Chen, Y.-L., & Hung, L. T.-H. (2009). Using decision trees to summarize associative classification rules. Expert Systems with Applications, 36(2), 2338–2351. doi:10.1016/j.eswa.2007.12.031

Chennoufi, A., & Mazroui, A. (2017). Morphological, syntactic and diacritics rules for automatic diacritization of Arabic sentences. Journal of King Saud University - Computer and Information Sciences, 29(2), 156–163. doi:10.1016/j.jksuci.2016.06.004

Chien, J.T. & Chueh, C.H. (2008). Latent Dirichlet language model for speech recognition. In Proc. Of IEEE Workshop on Spoken Language Technology, pp. 201-204.

Chien, J.-T., & Wu, M.-S. (2008). Adaptive Bayesian Latent Semantic Analysis. IEEE Transactions on Audio, Speech, and Language Processing, 16(1), 198–207. doi:10.1109/tasl.2007.909452

Cohn, T. A., & Lapata, M. (2009). Sentence Compression as Tree Transduction. Journal of Artificial Intelligence Research, 34, 637–674. doi:10.1613/jair.2655

Conroy, J. M., & O'leary, D. P. (2001). Text summarization via hidden Markov models. Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '01. doi:10.1145/383952.384042

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3):273–297

Dang, Q., Zhang, J., Lu, Y., & Zhang, K. (2013). WordNet-based suffix tree clustering algorithm. In Paper presented at the 2013 international conference on information science and computer applications (ISCA 2013).

Darwish, K. & Oard, D. (2002). CLIR Experiments at Maryland for TREC 2002: Evidence Combination for Arabic-English Retrieval. In TREC 2002. Gaithersburg, MD.

Darwish, K., & Magdy, W. (2014). Arabic Information Retrieval. Foundations and Trends® in Information Retrieval, 7(4), 239–342. doi:10.1561/1500000031

Das, A., Ganguly, D., & Garain, U. (2017). Named Entity Recognition with Word Embeddings and Wikipedia Categories for a Low-Resource Language. ACM Transactions on Asian and Low-Resource Language Information Processing, 16(3), 1–19. doi:10.1145/3015467

Daume III, H., Echihabi, A., Marcu, D., StefanMunteanu, D., & Soricu, R. (2002). GLEANS: A Generator of Logical Extracts and Abstracts for Nice Summaries. In Proceedings of the Document Understanding Conference (DUC-2002), Philadelphia, PA, July 2002.

De Jong, G. (1982). An overview of the FRUMP system. In Strategies for natural language processing, W.G. Lehnert and M.H. Ringle (ed.), Hillsdale, Erlbaum, pp.149-176

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American Society for Information Science, vol. 41, no. 6, p. 391-407.

Deng, WY., Zheng, QH., Chen, L., & Xu, XB. (2010). Research on extreme learning of neural networks. Chinese Journal of Computers, 33(2), 279–287. doi:10.3724/sp.j.1016.2010.00279

Denil, M., Demiraj, A., de Freitas, N.: Extraction of salient sentences from labelled documents. arXiv preprint arXiv:1412.6815 (2014)

Dhungana, U.R., Shakya, S., Baral, K., & Sharma, B. (2015). Word Sense Disambiguation using WSD specific WordNet of polysemy words. In Semantic Computing (ICSC), 2015 IEEE International Conference on , vol., no., pp.148-152, 7-9.

Diab, M. T. (2009). Second Generation Tools (AMIRA 2.0) : Fast and Robust Tokenization, POS tagging, and Base Phrase Chunking. In Khalid Choukri et Bente Maegaard, editeurs, Proceedings of the Second International Conference on Arabic Language Resources and Tools, Cairo, Egypt, April 2009. The MEDAR Consortium.

Diab, M. T., Hacioglu, K., & Jurafsky, D. (2004). Automatic tagging of Arabic text : From raw text to base phrase chunks. In Proceedings of HLT-NAACL 2004: Short papers, pages 149–152. Association for Computational Linguistics.

Diab, M. T., Kadri, H., Daniel, J. (2007). Automated methods for processing Arabic text: from tokenization to base phrase chunking. Arabic Computational Morphology: Knowledge-Based and Empirical Methods.

Donahue, J., Hendricks, L. A., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K., & Darrell, T. (2017). Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), 677–691. doi:10.1109/tpami.2016.2599174

Douzidia, F.S., & Lapalme, G. (2004). Lakhas, an Arabic summarization system. In: Proceedings of 2004 Doc. Understanding Conf. (DUC2004), Boston, MA.

Edmundson, H. P. (1969). New Methods in Automatic Extracting. Journal of the ACM, 16(2), 264–285. doi:10.1145/321510.321519

Elberrichi, Z., & Abidi, K. (2012). Arabic text categorization: a comparative study of different representation modes. The International Arab Journal of Information Technology, 9, 465-470.

El-Fishawy, N., Hamouda, A., Attiya, G.M., & Atef, M. (2014). Arabic summarization in twitter social network. Ain Shams Engineering Journal 5(2):411–420.

El-Haj, M. O., & Hammo, B. H. (2008). Evaluation of Query-Based Arabic Text Summarization System. 2008 International Conference on Natural Language Processing and Knowledge Engineering. Beijing, pp. 1-7. doi:10.1109/nlpke.2008.4906790

El-Haj, M. O., Kruschwitz, U., & Fox, C. (2010). Using mechanical turk to create a corpus of arabic summaries. In: Proceedings of the international conference on language resources and evaluation (LREC), Valletta, Malta, pp 36–39, in the language resources (LRs) and Human Language Technologies (HLT) for Semitic Languages workshop held in conjunction with the 7th international language resources and evaluation conference (LREC 2010).

El-Haj, M. O., Kruschwitz, U., & Fox, C. (2011a). Experimenting with Automatic Text Summarisation for Arabic. Lecture Notes in Computer Science, 490–499. doi:10.1007/978-3-642-20095-3_45

El-Haj, M. O., Kruschwitz, U., & Fox, C. (2011b). Exploring Clustering for Multi-document Arabic Summarisation. Lecture Notes in Computer Science, 550–561. doi:10.1007/978-3-642-25631-8_50

Elman, J. L. (1991). Distributed representations, simple recurrent networks, and grammatical structure. Machine Learning, 7(2-3), 195–225. doi:10.1007/bf00114844

El-Shishtawy, T., & Al-sammak, A. (2009). Arabic Keyphrase Extraction using Linguistic knowledge and Machine Learning Techniques. In Proceedings of the Second International Conference on Arabic Language Resources and Tools, The MEDAR Consortium, Cairo, Egypt.

El-Shishtawy, T., & El-Ghannam, F. (2012). Keyphrase based Arabic summarizer (KPAS). Informatics and Systems (INFOS), 2012 8th International Conference on, vol., no., pp.NLP-7, NLP-14, 14-16 May.

Enríquez, F., Troyano, J. A., & López-Solaz, T. (2016). An approach to the use of word embeddings in an opinion classification task. Expert Systems with Applications, 66, 1–6. doi:10.1016/j.eswa.2016.09.005

Er, M. J., Zhang, Y., Wang, N., & Pratama, M. (2016). Attention pooling-based convolutional neural network for sentence modelling. Information Sciences, 373, 388–403. doi:10.1016/j.ins.2016.08.084

Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. Journal of Artificial Intelligence Research, 22, 457–479.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639):115-118 (2017). doi:10.1038/nature21056

Estiri, A., Kahani, M., Ghaemi, H., & Abasi, M. (2014). Improvement of an abstractive summarization evaluation tool using lexical-semantic relations and weighted syntax tags in Farsi language. In Intelligent Systems (ICIS), 2014 Iranian Conference on , vol., no., pp.1-6, 4-6 doi: 10.1109/IranianCIS.2014.6802594.

Fang, H., Lu, W., Wu, F., Zhang, Y., Shang, X., Shao, J., & Zhuang, Y. (2015). Topic aspect-oriented summarization via group selection. Neurocomputing, 149, 1613–1619. doi:10.1016/j.neucom.2014.08.031

Farghaly, A., & Shaalan, K. (2009). Arabic Natural Language Processing. ACM Transactions on Asian Language Information Processing, 8(4), 1–22. doi:10.1145/1644879.1644881

Fattah, M. A. (2014). A hybrid machine learning model for multi-document summarization. Applied Intelligence, 40(4), 592–600. doi:10.1007/s10489-013-0490-0

Fattah, M. A., & Ren, F. (2009). GA, MR, FFNN, PNN and GMM based models for automatic text summarization. Computer Speech & Language, 23(1), 126–144. doi:10.1016/j.csl.2008.04.002

Feldman, R., & Sanger, J. (2007). The Text Mining Handbook. Advanced Approaches In Analyzing Unstructured Data. Press, Cambridge University. ISBN 978-0-521-83657-9

Ferreira, R., De Souza Cabral, L., Freitas, F., Lins, R. D., de França Silva, G., Simske, S. J., & Favaro, L. (2014). A multi-document summarization system based on statistics and linguistic treatment. Expert Systems with Applications, 41(13), 5780–5787. doi:10.1016/j.eswa.2014.03.023

Ferreira, R., De Souza Cabral, L., Lins, R. D., Pereira e Silva, G., Freitas, F., Cavalcanti, G. D. C., … Favaro, L. (2013b). Assessing sentence scoring techniques for extractive text summarization. Expert Systems with Applications, 40(14), 5755–5764. doi:10.1016/j.eswa.2013.04.023

Ferreira, R., Freitas, F., de Souza Cabral, L., Dueire Lins R, Lima R, França G, Simskez SJ, Favaro L (2013a). A four dimension graph model for automatic text summarization. In: 2013 IEEE/WIC/ACM international joint conferences on web intelligence (WI) and intelligent agent technologies (IAT), vol. 1, IEEE, pp 389–396

Firat, O., Cho, K., Sankaran, B., Yarman Vural, F. T., & Bengio, Y. (2017). Multi-way, multilingual neural machine translation. Computer Speech & Language 45:236–252. doi:10.1016/j.csl.2016.10.006

Fodeh, S. J., Punch, W. F., & Tan, P. -N. (2009). Combining statistics and semantics via ensemble model for document clustering. In Paper presented at the proceedings of the 2009 ACM symposium on applied computing.

Froud, H., Benslimane, R., Lachkar, A., & Ouatik, S. A. (2010). Stemming and similarity measures for Arabic Documents Clustering. 2010 5th International Symposium On I/V Communications and Mobile Network, Rabat. doi:10.1109/isvc.2010.5656417

Froud, H., Lachkar, A., & Ouatik, S. A. (2012a). Stemming versus Light Stemming for measuring the simitilarity between Arabic Words with Latent Semantic Analysis model. 2012 Colloquium in Information Science and Technology. doi:10.1109/cist.2012.6388065

Froud, H., Lachkar, A., & Ouatik, S. A. (2012b). Stemming for Arabic words similarity measures based on Latent Semantic Analysis model. 2012 International Conference on Multimedia Computing and Systems, Tangier. doi:10.1109/icmcs.2012.6320289

Fung, P., & Ngai, G. (2006). One story, one flow: Hidden Markov Story Models for multilingual multidocument. ACM Transactions on Speech and Language Processing, 3(2), 1–16. doi:10.1145/1149290.1151099

Gagnon, M., & Da Sylva, L. (2006). Text Compression by Syntactic Pruning. Lecture Notes in Computer Science, 312–323. doi:10.1007/11766247_27

Gao, JB., Zhang, BW., & Chen, XH. (2015). A WordNet-based semantic similarity measurement combining edge-counting and information content theory. Engineering Applications of Artificial Intelligence, 39, 80–88.

García-Hernández, R. A., & Ledeneva, Y. (2013). Single Extractive Text Summarization Based on a Genetic Algorithm. Pattern Recognition, 374–383. doi:10.1007/978-3-642-38989-4_38

García-Hernández, R. A., Montiel, R., Ledeneva, Y., Rendón, E., Gelbukh, A., & Cruz, R. (2008). Text Summarization by Sentence Extraction Using Unsupervised Learning. In Proceedings of the 7th Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence (MICAI '08), Alexander Gelbukh and Eduardo F. Morales (Eds.). Springer-Verlag, Berlin, Heidelberg, 133-143.Lecture Notes in Computer Science, pp. 133–143. doi:10.1007/978-3-540-88636-5_12

Giatsoglou, M., Vozalis, M. G., Diamantaras, K., Vakali, A., Sarigiannidis, G., & Chatzisavvas, K. C. (2017). Sentiment analysis leveraging emotions and word embeddings. Expert Systems with Applications, 69, 214–224. doi:10.1016/j.eswa.2016.10.043

Goldstein J, Kantrowitz M, Mittal VO, Carbonell JG (1999) Summarizing text documents: sentence selection and evaluation metrics. In: SIGIR ’99: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, August 15–19, 1999, Berkeley, CA, USA. ACM, New York, pp 121–128

Gong, Y., & Liu, X. (2001). Generic text summarization using relevance measure and latent semantic analysis. Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR ’01. doi:10.1145/383952.383955

Graff, D., Maamouri, M., Bouziri, B., Krouna, S., Kulick, S., Buckwalter, T. (2010). Standard Arabic Morphological Analyzer (SAMA).

Gross, O., Doucet, A., Toivonen, H. (2014). Document summarization based on word associations. In: Proceedings of the 37th international ACM SIGIR conference on research & development in information retrieval, ACM, pp 1023–1026

Gulshan, V., Peng, L., Coram, M. et al. : Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. JAMA 316(22):2402 (2016). doi:10.1001/jama.2016.17216

Gupta, V. (2014). A language independent hybrid approach for text summarization. In: Emerging trends in computing and communication, Springer, pp 71–77

Ha, J.W., Kang, D., Pyo, H., Kim, J.: News2Images: Automatically Summarizing News Articles into Image-Based Contents via Deep Learning. In: 3rd International Workshop on News Recommendation and Analytics (INRA 2015) (with RECSYS 2015), Vienna, Austria (2015)

Habash, N., Rambow, O., Roth, R. (2009). MADA + TOKAN: a toolkit for Arabic tokenization, diacritization, morphological disambiguation, POS tagging, stemming and lemmatization, Proc. Second Int. Conf. Arab. Lang. Resour. Tools, pp. 102–109.

Habash, N., Roth, R., Rambow, O., Eskander, R., Tomeh, N. (2013). Morphological analysis and disambiguation for dialectal Arabic. Hlt-Naacl, 426–432.

Habash, N.Y. (2010). Introduction to Arabic natural language processing. Synthesis Lectures on Human Language Technologies, 3, 1–187.

Haboush, A., Momani, A., Al-Zoubi, M., & Tarazi, M. (2012). Arabic Text Summarization Model Using Clustering Techniques. World of Computer Science and Information Technology Journal (WCSIT) ISSN: 2221-0741 Vol. 2, No. 3,62 – 67.

Hannah, M. E., Geetha, T. V., & Mukherjee, S. (2011). Automatic Extractive Text Summarization Based on Fuzzy Logic: A Sentence Oriented Approach. Lecture Notes in Computer Science, 530–538. doi:10.1007/978-3-642-27172-4_63

Hatori, J., Murakami, A., & Tsujii, J. (2011). Multi-topical discussion summarization using structured lexical chains and cue words. In International conference on intelligent text processing and computational linguistics (pp. 313–327). Springer.

Heu, J.-U., Qasim, I., & Lee, D.-H. (2015). FoDoSu: Multi-document summarization exploiting semantic analysis based on social Folksonomy. Information Processing & Management, 51(1), 212–225. doi:10.1016/j.ipm.2014.06.003

Hinton, G. E., & Salakhutdinov, R.R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), 504–507. doi:10.1126/science.1127647

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A Fast Learning Algorithm for Deep Belief Nets. Neural Computation, 18(7), 1527–1554. doi:10.1162/neco.2006.18.7.1527

Hirao, T., Yoshida, Y., Nishino, M., Yasuda, N., Nagata, M. (2013). Single-document summarization as a tree knapsack problem. In: EMNLP, pp 1515–1520

Horowitz, E., Sahni, S., Rajasekaran, S. (1998). Computer Algorithms. Computer Science Press, New York.

Hovy, E.H. (2005). Automated text summarization. In: Mitkov, R. (Ed.), The Oxford Handbook of Computational Linguistics. Oxford Univ. Press, pp. 583–598.

Huang, G., Song, S., Gupta, J. N. D., Wu, C. (2014). Semi-Supervised and Unsupervised Extreme Learning Machines. IEEE Transactions on Cybernetics, 44(12), 2405–2417. doi:10.1109/tcyb.2014.2307349

Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme Learning Machine for Regression and Multiclass Classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 42(2), 513–529. doi:10.1109/tsmcb.2011.2168604

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. Neurocomputing, 70(1-3), 489–501. doi:10.1016/j.neucom.2005.12.126

Huang, H., Ma, H., JW van Triest, H., Wei, Y., & Qian, W. (2018b). Automatic detection of neovascularization in retinal images using extreme learning machine. Neurocomputing, 277, 218–227. doi:10.1016/j.neucom.2017.03.093

Huang, J., Yu, Z. L., & Gu, Z. (2018a). A clustering method based on extreme learning machine. Neurocomputing, 277, 108–119. doi:10.1016/j.neucom.2017.02.100

Ibrahim, A., Elghazaly, T. (2013). Rhetorical representation and vector representation in summarizing Arabic text. Natural language processing and information systems, lecture notes in computer science, vol 7934. Springer, Berlin, pp 421–424.

Ibrahim, A., & Elghazaly, T. (2012). Arabic text summarization using Rhetorical Structure Theory. In: 2012 8th International Conference on Informatics and Systems (INFOS), Cairo, 2012, pp. NLP-34-NLP-38.

Ijjina, E. P., & C, K. M. (2016). Classification of human actions using pose-based features and stacked auto encoder. Pattern Recognition Letters, 83, 268–277. doi:10.1016/j.patrec.2016.03.021

Iosifidis, A., Tefas, A., & Pitas, I. (2015). Human Action Recognition Based on Multi-View Regularized Extreme Learning Machine. International Journal on Artificial Intelligence Tools, 24(05), 1540020. doi:10.1142/s0218213015400205

Järvelin K, Kekäläinen J (2002) Cumulated gain-based evaluation of ir techniques. ACM Trans Inf Syst 20(4):422–446.

Jin, R., Abu-Ata, M., Xiang, Y., & Ruan, N. (2008). Effective and efficient itemset pattern summarization. Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 08. doi:10.1145/1401890.1401941

Kahou, S. E., Bouthillier, X., Lamblin, P., Gulcehre, C., Michalski, V., Konda, K., … Bengio, Y. (2015). EmoNets: Multimodal deep learning approaches for emotion recognition in video. Journal on Multimodal User Interfaces, 10(2), 99–111. doi:10.1007/s12193-015-0195-2

Kamkarhaghighi, M., & Makrehchi, M. (2017). Content Tree Word Embedding for document representation. Expert Systems with Applications, 90, 241–249. doi:10.1016/j.eswa.2017.08.021

Kammoun, N., Belguith, L., Hamadou, A. (2010). The MORPH2 new version: a robust morphological analyzer for Arabic texts. 10th International Conference Journées d'Analyse Statistique Des Données Textuelles. Sapienza University of Rome.

Kasun, L. L. C., Zhou, H., Huang, G. B., Vong, C. M. (2013). Representational learning with ELMs for big data, IEEE Intelligent Systems 28 (6) (2013) 31–34.

Khan, A., Salim, N., & Jaya Kumar, Y. (2015). A framework for multi-document abstractive summarization based on semantic role labelling. Applied Soft Computing, 30, 737–747.

Khan, A.U., Khan, S., & Mahmood, W. (2005). MRST: A New Technique For Information Summarization. In: The Second World Enformatika Conference, WEC'05, pp 249–252, February 25-27, 2005, Istanbul, Turkey.

Khoja, S. (2001). APT: Arabic Part-of-speech tagger. In: Proceedings of the Student Workshop at the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL2001), Carnegie Mellon University, Pittsburgh, Pennsylvania, pp. 20–25.

Khoja, S., & Garside, R. (1999). Stemming Arabic Text. In Computing Department. Lancaster University. http://zeus.cs.pacificu.edu/shereen/research.htm

Thakkar, K. S., Dharaskar, R. V., & Chandak, M. B. (2010). Graph-Based Algorithms for Text Summarization. 2010 3rd International Conference on Emerging Trends in Engineering and Technology. doi:10.1109/icetet.2010.104

Kikuchi, Y., Hirao, T., Takamura, H., Okumura, M., Nagata, M. (2014). Single document summarization based on nested tree structure. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, vol. 2, pp 315–320

Kim, E., Corte-Real, M., & Baloch, Z. (2016). A deep semantic mobile application for thyroid cytopathology. Medical Imaging 2016: PACS and Imaging Informatics: Next Generation and Innovations (2016). doi:10.1117/12.2216468

Kingma, D.P., Mohamed, S., Rezende, D.J., & Welling, M. (2014). Semi-Supervised Learning with Deep Generative Models. In: Proceedings of Neural Information Processing Systems (NIPS'14), pp 3581–3589.

Kingma, DP., & Welling, M. (2014). Auto-encoding variational bayes. In: Proceedings of the International Conference on Learning Representations, Banff, Canada

Knight, K., & Marcu, D. (2002). Summarization beyond sentence extraction: A probabilistic approach to sentence compression. Artificial Intelligence, 139(1), 91–107. doi:10.1016/s0004-3702(02)00222-9

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS'12), Lake Tahoe, Nevada, USA, pp 1090–1098.

Kumar, Y.J., Salim, N., Abuobieda, A., & Albaham, A.T. (2014). Multi document summarization based on news components using fuzzy cross-document relations. Applied Soft Computing 21:265–279

Kuo, J.J., & Chen, H.H. (2008). Multidocument Summary Generation: Using Informative and Event Words. ACM Trans Asian Lang Inf Process (TALIP) 7(1):1–23.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '95. doi:10.1145/215206.215333

Kyoomarsi, F., Khosravi, H., Eslami, E., Dehkordy, P. K., & Tajoddin, A. (2008). Optimizing Text Summarization Based on Fuzzy Logic. Seventh IEEE/ACIS International Conference on Computer and Information Science (icis 2008). doi:10.1109/icis.2008.46

Larkey, L., Ballesteros, L., Connell, M. (2007). Light Stemming for Arabic Information Retrieval. Arabic Computational Morphology, 221–243

Larkey, L., Connell, M. (2002). Arabic Information Retrieval at UMass. In: TREC-10. NIST SPECIAL PUBLICATION SP, pp. 562-570

Larkey, S., Ballesteros, L., Connell, E. (2002). Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Aanalysis. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, Finland

Larsen, B., Aone, C.: Fast and Effective Text Mining Using Linear-Time Document Clustering. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining KDD, pp. 16–22 (1999)

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324 (1998). doi:10.1109/5.726791

Ledeneva, Y., García-Hernández, R. A., & Gelbukh, A. (2014). Graph Ranking on Maximal Frequent Sequences for Single Extractive Text Summarization. In: Computational Linguistics and Intelligent Text Processing. CICLing 2014. Lecture Notes in Computer Science, Springer, pp. 466–480. doi:10.1007/978-3-642-54903-8_39

Ledeneva, Y., Hernández, R. G., Soto, R. M., Reyes, R. C., & Gelbukh, A. (2011). EM Clustering Algorithm for Automatic Text Summarization. Lecture Notes in Computer Science, 305–315. doi:10.1007/978-3-642-25324-9_26

Lee, J.-H., Park, S., Ahn, C.-M., & Kim, D. (2009). Automatic generic document summarization based on non-negative matrix factorization. Information Processing & Management, 45(1), 20–34. doi:10.1016/j.ipm.2008.06.002

Lee, S., Belkasim, S., Zhang, Y. (2013). Multi-document text summarization using topic model and fuzzy logic. In: Machine learning and data mining in pattern recognition, Springer, pp 159–168

Li Chengcheng. (2010). Automatic Text Summarization based on Rhetorical Structure Theory. In: 2010 International Conference on Computer Application and System Modeling (ICCASM 2010). doi:10.1109/iccasm.2010.5622918

Li, F., Zhang, M., Tian, B., Chen, B., Fu, G., & Ji, D. (2017a). Recognizing irregular entities in biomedical text via deep neural networks. Pattern Recognition Letters. doi:10.1016/j.patrec.2017.06.009

Li, H., Misra, S.: Prediction of Subsurface NMR T2 Distributions in a Shale Petroleum System Using Variational Autoencoder-Based Neural Networks. IEEE Geoscience and Remote Sensing Letters 14(12):2395–2397 (2017). doi:10.1109/lgrs.2017.2766130

Li, H., Wang, S., & Kot, A. (2017). Image Recapture Detection with Convolutional and Recurrent Neural Networks. Electronic Imaging, 2017(7), 87–91. doi:10.2352/issn.2470-1173.2017.7.mwsf-329

Li, S., Ouyang, Y., Wang, W., & Sun, B. (2007). Multi-Document Summarization Using Support Vector Regression. In Proceedings of the 7th Document Understanding Conference. DUC.

Li, Y., Li, H., Cai, Q., & Han, D. (2012). A novel semantic similarity measure within sentences. In Computer Science and Network Technology (ICCSNT), 2012 2nd International Conference on , vol., no., pp.1176-1179, 29-31 doi: 10.1109/ICCSNT.2012.6526134.

Li, Y., Zhang, X., Jin, H., Li, X., Wang, Q., He, Q., & Huang, Q. (2017b). Using multi-stream hierarchical deep neural network to extract deep audio feature for acoustic event detection. Multimedia Tools and Applications, 77(1), 897–916. doi:10.1007/s11042-016-4332-z

Lin, Z. (2007). Graph-Based methods for automatic text summarization. Ph.D. thesis, school of computing National University of Singapore 2006–07.

Lin, C.-Y. (1999). Training a selection function for extraction. Proceedings of the Eighth International Conference on Information and Knowledge Management – CIKM '99. doi:10.1145/319950.319957

Lin, C.Y. (2004). ROUGE: A package for automatic evaluation of summaries. In Proceedings of workshop on text summarization branches out, post-conference workshop of ACL, pp 74–81

Lin, D. (1998). An information-theoretic definition of similarity. In Proceedings of the 15th International Conference on Machine Learning. Madison, WI, vol. 98, pp. 296–304.

Lin, X., Liu, J., & Kang, X. (2016). Audio Recapture Detection With Convolutional Neural Networks. IEEE Transactions on Multimedia, 18(8), 1480–1487. doi:10.1109/tmm.2016.2571999

Liu, T., Liyanaarachchi Lekamalage, C. K., Huang, G.-B., & Lin, Z. (2018). Extreme Learning Machine for Joint Embedding and Clustering. Neurocomputing, 277, 78–88. doi:10.1016/j.neucom.2017.01.115

Llopis, F., Vicedo, J. L., & Ferrández, A. (2002). Passage selection to improve Question Answering. Proceeding of the 2002 Conference on Multilingual Summarization and Question Answering - COLING-02. doi:10.3115/1118845.1118851

Lloret, E., & Palomar, M. (2012). Text summarisation in progress: A literature review. Artificial Intelligence Review, 37(1), 1–41.

Loza, V., Lahiri, S., Mihalcea, R., & Lai, P.-H. (2014). Building a dataset for summarization and keyword extraction from emails. In Proceedings of the ninth international conference on language resources and evaluation (LREC'14).

Lu, S., Lu, Z., Yang, J., Yang, M., & Wang, S. (2016). A pathological brain detection system based on kernel based ELM. Multimedia Tools and Applications, 77(3), 3715–3728. doi:10.1007/s11042-016-3559-z

Luhn, HP.: The automatic creation of literature abstracts. IBM Journal of Research Development 2(2):159-165 (1958)

Ma, Y., Wu, J. (2014). Combining n-gram and dependency word pair for multi-document summarization. In: 2014 IEEE 17th international conference on computational science and engineering (CSE), IEEE, pp 27–31

Malik, R., Subramaniam, V., and Kaushik, S. (2007) Automatically Selecting Answer Templates to Respond to Customer Emails. In Proceedings of IJCAI'07, Hyderabad, India, 1659-1664.

Mani, I. & Maybury, M.T. (1999). Advances in Automatic Summarization. MIT Press, Cambridge, MA.

Mann, W.C., Thompson, S.A. (1987). Rhetorical Structure Theory: toward a functional theory of text organization. Interdisciplinary Journal for the Study of Discourse. 8 (3): 243–281. doi:10.1515/text.1.1988.8.3.243

Manning, C.D., Raghavan, P., Schutze, H.: Introduction to Information Retrieval, vol. 1. Cambridge University Press, Cambridge (2008)

Mashechkin, I. V., Petrovskiy, M. I., Popov, D. S., & Tsarev, D. V. (2011). Automatic text summarization using latent semantic analysis. Programming and Computer Software, 37(6), 299–305. doi:10.1134/s0361768811060041

Mazdak, N. (2004). FarsiSum-a Persian text summarizer. Master thesis, Department of linguistics, Stockholm University.

McKeown, K., & Radev, D. R. (1995). Generating summaries of multiple news articles. Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '95. doi:10.1145/215206.215334

McKeown, K., Klavans, J., Hatzivassiloglou, V., Barzilay, R., & Eskin, E., (1999). Towards Multidocument Summarization by Reformulation: Progress and Prospects. In Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence (AAAI '99/IAAI '99). American Association for Artificial Intelligence, Menlo Park, CA, USA, 453-460.

Mendoza, M., Bonilla, S., Noguera, C., Cobos, C., León, E. (2014). Extractive single-document summarization based on genetic operators and guided local search. Expert Syst Appl 41(9):4158–4169

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Spain, pp 404–411.

Mihalcea, R., & Tarau, P. (2005) An Algorithm for Language Independent Single and Multiple Document Summarization. In Proceedings of the International Joint Conference on Natural Language Processing, Korea.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Miller, E., Shen, D., Liu, J., Nicholas, C. (1999). Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System," Computer Journal of Digital Information, vol. 1, no. 5, pp. 257-265, 1999.

Miller, G. A. (1995). WordNet: A lexical database for English. Communications of the ACM, 38(11), 39–41

Minhas, R., Baradarani, A., Seifzadeh, S., & Jonathan Wu, Q. M. (2010). Human action recognition using extreme learning machine based on visual vocabularies. Neurocomputing, 73(10-12), 1906–1917. doi:10.1016/j.neucom.2010.01.020

Mittal, V. O., Kantrowitz, M., Goldstein, J., Carbonell, J. G. (1999). Selecting text spans for document summaries: heuristics and metrics. In AAAI/IAAI, pp 467–473

Moawad, IF, Aref M (2012) Semantic graph reduction approach for abstractive Text Summarization. In: Proceedings of ICCES 2012, 2012 International Conference on Computer Engineering and Systems, pp 132–138. doi:10.1109/ICCES.2012.6408498

Mohammed, A. A., Minhas, R., Jonathan Wu, Q. M., & Sid-Ahmed, M. A. (2011). Human face recognition based on multidimensional PCA and extreme learning machine. Pattern Recognition, 44(10-11), 2588–2597. doi:10.1016/j.patcog.2011.03.013

Nguyen, N. T. H., Miwa, M., Tsuruoka, Y., & Tojo, S. (2015). Identifying synonymy between relational phrases using word embeddings. Journal of Biomedical Informatics, 56, 94–102. doi:10.1016/j.jbi.2015.05.010

Nguyen-Hoang, T.-A., Nguyen, K., & Tran, Q.-V. (2012). TSGVi: a graph-based summarization system for Vietnamese documents. Journal of Ambient Intelligence and Humanized Computing, 3(4), 305–313. doi:10.1007/s12652-012-0143-x

Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H. G., & Ogata, T. (2014). Audio-visual speech recognition using deep learning. Applied Intelligence, 42(4), 722–737. doi:10.1007/s10489-014-0629-7

Osama, A. Ghanem. & Wesam, M. Ashour. (2012). Stemming Effectiveness in Clustering of Arabic Documents. International Journal of Computer Applications, 49(5), 1–6. doi:10.5120/7620-0674

Oufaida, H., Nouali, O., Blache, P. (2014). Minimum redundancy and maximum relevance for single and multidocument arabic text summarization. Journal of King Saud University - Computer and Information Sciences 26(4):450–461 special Issue on Arabic NLP.

Ozsoy, M. G., Alpaslan, F. N., & Cicekli, I. (2011). Text summarization using Latent Semantic Analysis. Journal of Information Science, 37(4), 405–417. doi:10.1177/0165551511408848

PadmaPriya, G., & Duraiswamy, K. (2018). Multi-Document Based Text Summarization Through Deep Learning Algorithm. International Journal of Business Intelligence and Data Mining, 1(1), 1. doi:10.1504/ijbidm.2018.10011144

Pal, A. R., & Saha, D. (2014). An approach to automatic text summarization using WordNet. 2014 IEEE International Advance Computing Conference (IACC), pp 1169–1173. doi:10.1109/iadcc.2014.6779492

Pasha, A., Al-badrashiny, M., Diab, M. T., Kholy, A., El Eskander, R., Habash, N., Pooleery, M., Rambow, O., Roth, R.M. (2014). MADAMIRA : a fast, comprehensive tool for morphological analysis and disambiguation of Arabic. Proc. 9th Lang. Resour. Eval. Conf., pp 1094–1101.

Patil, K., & Brazdil, P. (2007). Sumgraph: Text summarization using centrality in the pathfinder network. In: IADIS International Conference Applied Computing 2007, pp. 3-10.

Pedersen, T. (2010). Information content measures of semantic similarity perform better without sense-tagged text. InHuman language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics, HLT '10 (pp. 329–332). Stroudsburg, PA, USA: Association for Computational Linguistics.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(8), 1226–1238. doi:10.1109/tpami.2005.159

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). doi:10.3115/v1/d14-1162

Pollack, J. B. (1990). Recursive distributed representations. Artificial Intelligence, 46(1-2), 77–105. doi:10.1016/0004-3702(90)90005-k

Porter, M. F. (1980). An algorithm for suffix stripping. Program, 14(3), 130–137

Pourvali, M., & Abadeh M. S. (2012). Automated Text Summarization Base on Lexicales Chain and graph Using of WordNet and Wikipedia Knowledge Base. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 3, January 2012.

Prochazka, S. (2006). Arabic. Encyclopedia of Language and Linguistics, volume 1. Elsevier, 2nd edition.

Radev, D. R., & McKeown, K. (1998). Generating Natural Language Summaries from Multiple On-Line Sources. Computational Linguistics, pp. 469–500.

Radev, D., Allison, T., Blair-Goldensohn, S., et al. (2004). MEAD - a platform for multidocument multilingual text summarization. In LREC 2004, Lisbon, Portugal.

Rahmani, M., & Akbarizadeh, G. (2015). Unsupervised feature learning based on sparse coding and spectral clustering for segmentation of synthetic aperture radar images. IET Computer Vision 9(5):629–638. doi:10.1049/iet-cvi.2014.0295

Ren, Y., Wang, R., & Ji, D. (2016). A topic-enhanced word embedding for Twitter sentiment classification. Information Sciences, 369, 188–198. doi:10.1016/j.ins.2016.06.040

Rezende, D.J., Mohamed, S., & Wierstra, D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In: Proceedings of the 31st International Conference on International Conference on Machine Learning (ICML'14), vol 32, Beijing, China, pp 1278–1286

Ribaldo, R., Akabane, A.T., Rino, L.H.M., & Pardo, T.A.S. (2012). Graph-based Methods for Multidocument Summarization: Exploring Relationship Maps, Complex Networks and Discourse Information. In: Caseli, H., Villavicencio, A., Teixeira, A., Perdigão, F. (eds.) PROPOR 2012. LNCS, vol. 7243, pp. 260–271. Springer, Heidelberg

Rong, W., Nie, Y., Ouyang, Y., Peng, B., & Xiong, Z. (2014). Auto-encoder based bagging architecture for sentiment analysis. Journal of Visual Languages & Computing, 25(6), 840–849. doi:10.1016/j.jvlc.2014.09.005

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536. doi:10.1038/323533a0

Saad, M., & Ashour, W. (2010). OSAC: Open Source Arabic Corpora. EEECS'10 the 6th International Symposium on Electrical and Electronics Engineering and Computer Science, pp. 118-123, European University of Lefke, Cyprus.

Sachdeva, P., Verma, S., & Singh, S.K. (2014). An improved approach to word sense disambiguation. In Signal Processing and Information Technology (ISSPIT), 2014 IEEE International Symposium on , vol., no., pp.000235-000240, 15-17.

Salton, G., Buckley, C. (1997). Improving Retrieval Performance by Relevance Feedback. In: Information Retrieval, pp. 355–364.

Samei, B., Estiagh, M., Keshtkar, F., Hashemi, S. (2014). Multi-document summarization using graph-based iterative ranking algorithms and information theoretical distortion measures. In: The Twenty-seventh international flairs conference

Savyanavar, P. A., & Mehta, B. (2016). Multi-Document Summarization Using TF-IDF Algorithm. International Journal Of Engineering And Computer Science. doi:10.18535/ijecs/v5i4.40

Schiffman, B., Mani, I., Concepcion, K. J. (2001). Producing biographical summaries: combining linguistic knowledge with corpus statistics. In: ACL, pp 450–457

Schilder, F., & Kondadadi, R. (2008). FastSum. Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies Short Papers - HLT '08. doi:10.3115/1557690.1557748

Schlesinger, J. D., O'Leary, D. P., & Conroy, J. M. (2008). Arabic/English Multi-document Summarization with CLASSY: The Past and the Future. Lecture Notes in Computer Science, pp. 568–581. doi:10.1007/978-3-540-78135-6_49

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y.: OverFeat: Integrated recognition, localization and detection using convolutional networks. In: Proceedings of the 2nd International Conference On Learning Representation (ICLR2014), Banff, Canada (2014)

Shaheen, M., & Ezzeldin, A. M. (2014). Arabic Question Answering: Systems, Resources, Tools, and Future Trends. Arabian Journal for Science and Engineering, 39(6), 4541–4564. doi:10.1007/s13369-014-1062-2

Sharma, AD., & Deep, S. (2014). Too long-didn‘t read a practical web based approach towards text summarization. In: Applied Algorithms, Springer, pp 198–208

Sobh, I., Darwish, N., & Fayek, M. (2007). An optimized dual classification system for Arabic extractive generic text summarization. In: Proceeding of the 7th conference on language engineering.

Song, S., Huang, H., & Ruan, T. (2018). Abstractive text summarization using LSTM-CNN based deep learning. Multimedia Tools and Applications. doi:10.1007/s11042-018-5749-3

Song, W., & Park, S. C. (2007). A Novel Document Clustering Model Based on Latent Semantic Analysis. Third International Conference on Semantics, Knowledge and Grid (SKG 2007). doi:10.1109/skg.2007.154

Spille, C., Ewert, S. D., Kollmeier, B., & Meyer, B. T. (2018). Predicting speech intelligibility with deep neural networks. Computer Speech & Language, 48, 51–66. doi:10.1016/j.csl.2017.10.004

Suanmali, L., Binwahlan, M. S., & Salim, N. (2009a). Sentence Features Fusion for Text Summarization Using Fuzzy Logic. 2009 Ninth International Conference on Hybrid Intelligent Systems. doi:10.1109/his.2009.36

Suanmali, L., Salim, N., & Binwahlan, M. S. (2009b). Fuzzy Logic Based Method for Improving Text Summarization. International Journal of Computer Science and Information Security, Vol. 2, No. 1.

Sun, S., Zhang, B., Xie, L., & Zhang, Y. (2017). An unsupervised deep domain adaptation approach for robust speech recognition. Neurocomputing, 257, 79–87. doi:10.1016/j.neucom.2016.11.063

Svore, K. M., Vanderwende, L., & Burges, C. J. (2007). Enhancing single-document summarization by combining RankNet and third-party sources. In: Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL). pp. 448–457.

Teng, Z., Liu, Y., Ren, F., Tsuchiya, S., & Ren, F. (2008). Single Document Summarization Based on Local Topic Identification and Word Frequency. 2008 Seventh Mexican International Conference on Artificial Intelligence. doi:10.1109/micai.2008.12

Ulrich, J., Carenini, G., Murray, G., & Ng, R. T. (2009). Regression-based summarization of email conversations. Icwsm

Ulrich, J., Murray, G., & Carenini, G. (2008). A publicly available annotated corpus for supervised email summarization. Association for the Advancement of Artificial Intelligence.

Varelas, G., Voutsakis, E., Raftopoulou, P., Petrakis, E. G. M., & Milios, E. E. (2005). Semantic similarity methods in wordnet and their application to information retrieval on the web. In Proceedings of the seventh annual ACM international workshop on web information and data management, WIDM ’05(pp. 10–16). New York, NY, USA: ACM.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th International Conference on Machine Learning - ICML ’08. doi:10.1145/1390156.1390294

Vodolazova, T., Lloret, E., Muñoz, R., & Palomar, M. (2013). The role of statistical and semantic features in single-document extractive summarization. Artificial Intelligence Research, 2(3):35.

Wang, L., Zhang, J., Liu, P., Choo, K.-K. R., & Huang, F. (2016). Spectral–spatial multi-feature-based deep learning for hyperspectral remote sensing image classification. Soft Computing, 21(1), 213–221. doi:10.1007/s00500-016-2246-3

Wang, Y., Ma, J. (2013). A comprehensive method for text summarization based on latent semantic analysis. In: natural language processing and Chinese computing, Springer, pp 394–401

Wei, T. T., Lu, Y. H., Chang, H. Y., Zhou, Q., & Bao, X. Y. (2015). A semantic approach for text clustering using WordNet and lexical chains, Expert Systems with Applications, 42 (4) 2264–2275.

White, M., Korelsky, T., Cardie, C., Ng, V., Pierce, D., & Wagstaff, K. (2001). Multidocument Summarization via Information Extraction. In Proceedings of HLT 2001, Human Language Technology Conference, San Diego, CA, 2001. Available at: http://dx.doi.org/10.21236/ada457772.

Wu, Z., & Palmer, M. (1994). Verbs semantics and lexical selection. In Paper presented at the proceedings of the 32nd annual meeting on association for computational linguistics.

Xiong, S., Lv, H., Zhao, W., & Ji, D. (2018). Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings. Neurocomputing, 275, 2459–2466. doi:10.1016/j.neucom.2017.11.023

Yang, L., Cai, X., Zhang, Y., & Shi, P. (2014). Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. Information Sciences, 260, 37–50. doi:10.1016/j.ins.2013.11.026

Yeh, J.-Y., Ke, H.-R., Yang, W.-P., & Meng, I.-H. (2005). Text summarization using a trainable summarizer and latent semantic analysis. Information Processing & Management, 41(1), 75–95. doi:10.1016/j.ipm.2004.04.003

Yousefi-Azar, M., & Hamey, L. (2017). Text summarization using unsupervised deep learning. Expert Systems with Applications, 68, 93–105. doi:10.1016/j.eswa.2016.10.017.

Yu, B., Xu, Z., & Li, C. (2008). Latent semantic analysis for text categorization using neural network. Knowledge-Based Systems, 21(8), 900–904. doi:10.1016/j.knosys.2008.03.045

Yu, J., Huang, D., & Wei, Z. (2018). Unsupervised image segmentation via Stacked Denoising Auto-encoder and hierarchical patch indexing. Signal Processing, 143, 346–353. doi:10.1016/j.sigpro.2017.07.009

Yu, L., Ma, J., Ren, F., & Kuroiwa, S. (2007). Automatic Text Summarization Based on Lexical Chains and Structural Features. Eighth ACIS International Conference on Software Engineering, Artificial

Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007). doi:10.1109/snpd.2007.385

Yu, L. C., Wang, J., Lai, K. R., & Zhang, X. (2018). Refining Word Embeddings Using Intensity Scores for Sentiment Analysis. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(3), 671–681. doi:10.1109/taslp.2017.2788182

Zajic, D., Dorr, B., & Schwartz, R. (2002). Automatic Headline Generation for Newspaper Stories. In the Proceedings of the ACL Workshop on Automatic Summarization/Document Understanding Conference (DUC), pp. 78--85.

Zhang, P., & Li, C. (2009). Automatic text summarization based on sentences clustering and extraction. 2009 2nd IEEE International Conference on Computer Science and Information Technology. doi:10.1109/iccsit.2009.5234971

Zhong, S., Liu, Y., Li, B., & Long, J. (2015). Query-oriented unsupervised multi-document summarization via deep learning model. Expert Systems with Applications, 42(21), 8146–8155. doi:10.1016/j.eswa.2015.05.034