جامعة سيدي محمد بن عبد الله بفاس
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵙⵉⴷⵉ ⵎⵓⵃⵎⵎⴰⴷ ⴱⵏ ⵄⴱⴷⵍⵍⴰⵀ ⵏ ⴼⴰⵙ
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH DE FES

كلية العلوم والتقنيات فاس
ⵜⴰⵖⵉⵡⴰⵏⵜ ⵏ ⵜⵎⴰⵙⵙⴰⵏⵉⵏ ⴷ ⵜⵙⵏⵉⴳⵉⵜⵉⵏ - ⴼⴰⵙ
FACULTÉ DES SCIENCES ET TECHNIQUES DE FÈS

Centre d'Etudes Doctorales : Sciences et Techniques de l'Ingénieur

N° d'ordre 51 /2021

THESE DE DOCTOART

Présentée par

Mr : Zakariae ALAMI MERROUNI

Spécialité : Informatique

Sujet de la thèse : Information Extraction and Summarization Solutions in IR and NLP Context

Thèse présentée et soutenue le 09 juillet 2021 devant le jury composé de :

Nom Prénom	Titre	Etablissement	
Mohammed Ouçamah Cherkaoui Malki	PES	Faculté des Sciences Dhar El Mehraz de Fès	Président
Said Ouatik ElAlaoui	PES	Ecole Nationale des Sciences Appliquées Kenitra	Rapporteur
EL Moukhtar Zemmouri	PH	Ecole Nationale Supérieure d'Arts et Métiers Meknès	Rapporteur
Ahmed Zinedine	PES	Faculté des Sciences Dhar El Mehraz de Fès	Rapporteur
Noureddine Ennahnahi	PH	Faculté des Sciences Dhar El Mehraz de Fès	Examinateur
Bouchra Frikh	PES	Ecole Supérieure de Technologie Fès	Directeur de thèse
Brahim Ouhbi	PES	Ecole Nationale Supérieure d'Arts et Métiers Meknès	Co-Directeur de Thèse

Laboratoire d'accueil : LIASSE, ENSA Fès.

Etablissement : Faculté des Sciences et Techniques de Fès (FSTF)

Titre de la thèse : Information Extraction and Summarization Solutions in IR and NLP Context.

Nom et prénom du candidat : Zakariae ALAMI MERROUNI
Spécialité : Informatique

Centre d'Etudes Doctorales : Sciences et Techniques de l'Ingénieur

Résumé de la thèse

De nos jours, l'information joue un rôle crucial et apporte des avantages compétitifs aux utilisateurs, lorsqu'elle est gérée correctement. Cependant, en raison du grand volume d'informations disponibles dans les différentes ressources sur l'Internet et les archives de données textuelles, les utilisateurs ne peuvent pas y faire face, et donc de nouvelles méthodes et approches basées sur le Traitement du Langage Naturel (LNP), Text Mining (TM), et la Recherche d'Informations (RI) sont apparues pour stocker, accéder, rechercher et extraire les informations requises par les individus d'une manière efficace et efficiente. Jusqu'à présent, plusieurs méthodes ont été développées pour faire face à ces défis émergents, parmi lesquelles nous trouvons : l'extraction automatique de mots-clés / phrases clés et le Résumé Automatique de Texte (ART) (y compris ses variantes : extractif et abstractif).

Les travaux de recherche réalisés dans cette thèse portent sur (1) la tâche d'Extraction Automatique des Phrases Clés (EAPC) pour récupérer des informations pertinentes, et (2) la tâche d'Automatisation du Résumé du Texte (ART) pour produire des résumés concis et fluides reflétant les informations les plus importantes tout en étant non répétitives, cohérentes et grammaticalement lisible. Ces tâches sont de plus en plus utilisées par de nombreuses plates-formes de contenu (par exemple, les applications de recherche d'informations textuelles, les moteurs de recherche) pour aider les utilisateurs à accéder aux informations nécessaires. En fait, extraire des informations pertinentes et les résumer correctement (pour être au plus près de ce qu'un humain pourrait écrire) est considéré comme une tâche complexe. En particulier, les tâches EAPC et ART impliquent l'utilisation de méthodes avancées de NLP et des algorithmes d'apprentissage automatique. Pourtant, la tâche ART est l'une des applications d'Intelligence Artificielle (IA) les plus difficiles, bien que prometteuses, en général, et des tâches NLP et de recherche d'information en particulier.

Afin de relever ces défis et d'extraire et de synthétiser les informations les plus pertinentes, nous avons apporté deux contributions principales dans cette thèse : 1) proposer une nouvelle méthode hybride d'extraction de phrase-clé (qui est indépendante du domaine traité) pour identifier et extraire des mots-clés en premier lieu et des phrases-clés informatifs et syntaxiquement corrects en deuxième lieu, et 2) présenter une nouvelle approche hybride ART qui relève les défis d'automatisation du résumé de texte dans un mode « extractif » et « abstractif ». En fait, nous proposons une approche ART indépendante du domaine d'application qui est capable de produire deux types de résumés : des extraits, qui reposent sur une approche purement 'extractive', et d'autres reposant sur l'approche abstraite. De plus, dans les deux contributions proposées (EAPC ou ART), nous avons intégré et proposé les meilleures techniques linguistiques, statistiques et sémantiques pour détecter et produire des informations pertinentes.

Pour démontrer l'intérêt des approches proposées, les systèmes EAPC et RAT ont été évalués quantitativement et qualitativement en utilisant différents ensembles de données en termes de couverture des contextes et des domaines. De plus, les recherches menées dans cette thèse montrent la pertinence des approches EAPC et RAT proposées, soit pour être utilisées seules ou pour améliorer les performances des applications émanant de RI, TM et NLP.

Mots clés : Traitement du Langage Naturel (NLP), Text Mining (TM), Recherche d'Informations (RI), l'extraction automatique de mots-clés / phrases clés (EAPC), L'automatisation du Résumé du Texte (ART)

Acknowledgments

Firstly, I would like to thank GOD for giving me the strength, the courage, and the patience to establish this thesis work. I would like express my sincere gratitude to my supervisor Prof. Bouchra FRIKH and Prof. Brahim OUHBI for their continuous support, invaluable advice and encouragement. Their guidance helped me to explore research challenges and thinking about scientific problems profoundly. I am really grateful to them. I would like to thank my supervisory committee members for their time and effort spent on thesis committee reporting. They provided very valuable feedback and helpful suggestions to my research. I also must thank the USMBA University for providing us various formations programs and support. I am forever thankful to my dear parents, you should know that your support, patience, encouragement was worth more than I can express on paper. Last but not the least, I would like to thank my wife, and my brother and sisters for their continuous and unparalleled love, help and support.

Abstract

In the current society, information plays a crucial role that brings competitive advantages to users, when it is managed correctly. However, due to the large amount of available information embedded within the various resources on the internet and textual data archives, users cannot cope with it, and therefore new methods and approaches based on Natural Language Processing (NLP), Text Mining (TM), and Information Retrieval (IR) are essential to store, access, search and extract the information required by individuals in an effective and efficient manner.
Up to the present time, several methods have been developed to deal with these emergent challenges, among them we find: Automatic keyword/keyphrase extraction (AKPE), and Automatic Text Summarization (ATS) (Including its variants: extractive and abstractive).

The research works carried out in this thesis deals with (1) Automatic Keyphrase Extraction (AKPE) task to retrieve relevant information and (2) Automatic Text Summarization (ATS) task to produce concise and fluent summaries of the most important information while being non-redundant, coherent and grammatically readable. These tasks are increasingly used by many content platforms (e.g., Text-based Information Retrieval applications, Search Engines) to assist users to access the needed information. In fact, extracting relevant information and summarizing it sufficiently (to be close as possible to what a human could write) is considered a complex task. Especially, the AKPE and ATS tasks involve the use of advanced NLP and computational methods. Yet, the ATS task is one of the most difficult, though promising, Artificial Intelligence (AI) applications in general, and NLP and IR more accurately. In order to address these challenges and to extract and synthesize more relevant information, in this thesis we made two main contributions: 1) proposing a new hybrid domain-independent keyphrase extraction method to identify and extract informative and syntactically correct keywords and keyphrases and 2) presenting a novel hybrid ATS approach which addresses the challenges of extractive and abstractive text summarization. In fact, we propose a domain-independent ATS approach that is able of producing two types of summaries: extracts, which rely on a pure extractive approach, and abstracts, an abstractive-oriented summary, that combines extractive and abstractive techniques. In addition, either for the proposed AKPE or the ATS methods, we incorporate and suggest the finest linguistic, statistic, and semantic techniques for detecting and producing relevant information.

To demonstrate the interest in the proposed approaches, both AKPE and ATS systems were evaluated quantitatively and qualitatively using different data sets in terms of coverage of contexts and domains. Furthermore, the research conducted in this thesis shows the appropriateness of the proposed AKPE and ATS approaches, either for being used on their own, as well as for improving the performance of IR, TM, and NLP applications.

Résumé

De nos jours, l'information joue un rôle crucial qui apporte des avantages compétitifs aux utilisateurs, lorsqu'elle est gérée correctement. Cependant, en raison du grand volume d'informations disponibles dans les différentes ressources sur l'Internet et les archives de données textuelles, les utilisateurs ne peuvent pas y faire face, et donc de nouvelles méthodes et approches basées sur le Traitement du Langage Naturel (LNP), Text Mining (TM), et la Recherche d'Informations (RI) sont essentielles pour stocker, accéder, rechercher et extraire les informations requises par les individus d'une manière efficace et efficiente. Jusqu'à présent, plusieurs méthodes ont été développées pour faire face à ces défis émergents, parmi lesquelles nous trouvons : l'extraction automatique de mots-clés / phrases clés et le Résumé Automatique de Texte (RAT) (y compris ses variantes : extractif et abstractif).

Les travaux de recherche réalisés dans cette thèse portent sur (1) la tâche d'Extraction Automatique des Phrases Clés (EAPC) pour récupérer des informations pertinentes, et (2) la tâche d'automatisation du Résumé de Texte (RAT) pour produire des résumés concis et fluides des informations les plus importantes tout en étant non-répétitives, cohérentes et grammaticalement lisible. Ces tâches sont de plus en plus utilisées par de nombreuses plates-formes de contenu (par exemple, les applications de recherche d'informations textuelles, les moteurs de recherche) pour aider les utilisateurs à accéder aux informations nécessaires. En fait, extraire des informations pertinentes et les résumer correctement (pour être au plus près de ce qu'un humain pourrait écrire) est considéré comme une tâche complexe. En particulier, les tâches EAPC et RAT impliquent l'utilisation de méthodes avancées de LNP et des algorithmes d'apprentissage automatique. Pourtant, la tâche RAT est l'une des applications d'Intelligence Artificielle (IA) les plus difficiles, bien que prometteuses, en général, et la LNP et l'RI en particulier. Afin de relever ces défis et d'extraire et de synthétiser les informations les plus pertinentes, nous avons apporté deux contributions principales dans cette thèse : 1) proposer une nouvelle méthode hybride d'extraction de phrase-clé (qui est indépendante du domaine traité) pour identifier et extraire des mots-clés en premier lieu et des phrases-clés informatifs et syntaxiquement corrects en deuxième lieu, et 2) présenter une nouvelle approche hybride RAT qui relève les défis d'automatisation du résumé de texte dans un mode « extractif » et « abstractif ». En fait, nous proposons une approche RAT indépendante du domaine d'application qui est capable de produire deux types de résumés : des extraits, qui reposent sur une approche purement 'extractive', et des abstrait, un résumé orienté vers l'abstraction. De plus, dans les deux contributions proposées (EAPC ou RAT), nous avons intégré et proposé les meilleures techniques linguistiques, statistiques et sémantiques pour détecter et produire des informations pertinentes.

Pour démontrer l'intérêt des approches proposées, les systèmes EAPC et RAT ont été évalués quantitativement et qualitativement en utilisant différents ensembles de données en termes de couverture des contextes et des domaines. De plus, les recherches menées dans cette thèse montrent la pertinence des approches EAPC et RAT proposées, soit pour être utilisées seules ou pour améliorer les performances des applications émanant de RI, TM et NLP.



Contents

Acknowledgments **1**

Abstract **2**

Résumé **3**

List of Tables **9**

List of Figures **10**

List of Abbreviations **11**

Chapter 1: General Introduction

1.1. Research Context & Motivations **12**

1.2. Research Issues **14**

1.3. The Contribution of The Thesis **16**

1.4. Overview of Thesis Organization **18**

Chapter 2: State of The Art

2.1 Introduction **20**

2.2 IR, IE, NLP, TM, and the AKPE **21**

2.2.1 Background and Research Methodology 22

2.2.1.1 From Keyword to Keyphrase 22

2.2.1.2 Keyphrases in IR, NLP and TM Tasks 22

2.2.1.3 Research Methodology and Criteria 25

2.2.2 The Automatic Keyphrase Extraction Task 26

2.2.2.1 The Automatic Keyphrase Extraction Process 26

2.2.2.2 Documents Pre-Processing and Candidate Phrases Identification Stage 27

2.2.2.3. Feature Engineering Methods 27

2.2.3. Review of Supervised, Unsupervised and Deep Learning Ranking Approaches 29

2.2.3.1 Supervised Approaches 30

2.2.3.2 Unsupervised Approaches 33

2.2.3.3 Deep Learning Approaches 40

2.2.4 Evaluation Methods 41

2.2.4.1 Manual Evaluation 41

2.2.4.2 Automatic Evaluation 42

2.2.5 Comparative Study of Existing Approaches 42
2.2.5.1 Results and Discussion 43
2.2.5.2 Recommendations 49
2.3. Contextual Information Retrieval 52
2.3.1. Introduction 52
2.3.2. The Notion of Context In IR 52
2.3.3. Contextual Information Retrieval Systems 53
2.3.3.1 Context Taxonomies 54
2.3.3.2 Context in IR Systems 57
2.3.3.3 Context Modeling 58
2.3.3.4. Contextual Retrieval 58
2.3.3.5 Uses of Context in IRS 59
2.3.3.6 Context Levels in IRS 60
2.3.4. Discussion and Recommendation 60
2.5. Summary and Conclusion 61
Chapter 3: HAKE: An unsupervised Approach for Automatic Keyphrase Extraction
3.1. Introduction 63
3.2. Aims and Related Works 64
3.2.1. Candidate Keyphrase Identification Methods 66
3.2.2. Candidate Keyphrases Scoring and Ranking Approaches 66
3.3. A New AKPE System 71
3.3.1. Identification of Candidate Keyphrases 73
3.3.1.1. Documents Pre-processing 74
3.3.1.2. Candidate Keyphrases Identification Technique 74
3.3.2. Feature Selection 77
3.3.3. Hybrid Keyword Scoring 82
3.3.4. Hybrid Keyphrase Scoring 82
3.4. Experimental Results and Evaluation 83
3.4.1. Datasets 83
3.4.2. Baseline Methods 85

3.4.3. Evaluation Metrics 85
3.5. Results and Discussion 86
3.5.1. Parameter Values Selection and Analysis 87
3.5.2. Linguistic Filter Comparison 90
3.5.3. Overall Performance (Against Baselines) 90
3.6. Conclusion 96

Chapter 4: EXABSUM: A hybrid Text Summarization System for Generating Extractive and Abstractive Summaries

4.1. Introduction 98
4.2. The ATS Task: A Review 99
4.2.1. ATS Systems Classifications 100
4.2.2. ATS Systems Applications 101
4.2.3. Automatic Summarization Approaches 102
4.2.3.1. Extractive Summarization 102
4.2.3.2. Abstractive Summarization 105
4.2.3.3. Hybrid Summarization 107
4.2.4. Available Corpora 108
4.2.5. Summary Evaluation 109
4.3. Related Works 112
4.4. EXABSUM: A hybrid Text Summarization System for Generating Extractive and Abstractive Summaries 115
4.4.1. The System's Architecture 115
4.4.2. $EXABSUM_{Extractive}$ Core Stages 115
4.4.2.1 Surface Linguistic Analysis 116
4.4.2.2 Redundancy Detection and Removal 117
4.4.2.3 Sentence Relevance 119
4.4.2.4 Summary Generation 124
4.4.3. $EXABSUM_{Abstractive}$ Core Stages 125
4.4.3.1 Word Graph Generation and Re-Ranking: 125
4.4.3.2 Paths Filtering 128
4.4.3.3 Re-Ranking Candidate Sentences Using Keyphrases 128
4.4.4. $EXABSUM_{Hybrid}$ Hybrid Summary Generation 130
4.4.5. Experimental Setup 131
4.4.5.1. Datasets 132
4.4.5.2. Evaluation Method 133

4.4.6. Experiments Results and Discussion 134

4.4.6.1. Parameter Values Selection and Appropriateness of The Analysis of The Used Scoring Techniques 135

4.3.6.2. Comparison to Related Works 142

4.5. Conclusion 148

Chapter 5: Conclusion and Work in Progress

5.1. Summary of Contributions.....151

5.2. Work in Progress and for the Future 154

5.3 Relevant Publications.....156

Bibliography……158

List of Tables

Table 2.1 Electronic databases investigated in this survey……25
Table 2.2 Some target journals-conferences-workshops investigated in this survey……26
Table 2.3 Some evaluated datasets in this survey……44
Table 2.4 Characteristic summary of some aforementioned AKPE techniques along with the used features, dataset, target domain, approach and performance rates……45
Table 2.5 Characteristic summary of the most representative AKPE techniques and their performance on the shared "SemEval-2010 dataset" ……48
Table 2.6 The strengths and weaknesses of supervised, unsupervised and DL approaches for the AKPE task……51
Table 2.7 Differences between classic information retrieval systems versus contextual information retrieval systems……59
Table 3.1 Some prominent systems for keyphrase extraction……67
Table 3.2 Detailed statistics about evaluated datasets……84
Table 3.3 The performance of HAKE system while changing the weighting parameters……88
Table 3.4 Performances of HAKE system with different noun phrase filters according to precision (P), recall (R) and F1-Score (F) ……90
Table 3.5 The performance of HAKE using (α =0,2), and (β =0,2) and n<=5 compared with Baselines. F1@10……93
Table 3.6 The performance of HAKE using (α =0,2), and (β =0,2) and n<=5 compared with 23 other methods according to precision (P), recall (R) and F1-score (F) on the SemEval2010 Dataset……95
Table 4.1 Statistics on the CNN and DUC datasets. ……133
Table 4.2 Example of summaries generated with EXABSUMExtractive and EXABSUMAbstractive, respectively for document WSJ891019-0021 (DUC 2002 corpus, cluster d062j), with 50% ratio of the original text. ……136
Table 4.3 ROUGE results for EXABSUMExtractive on DUC2001 dataset while analyzing its features, Comb denotes the combination of the selected scoring techniques……137
Table 4.4 ROUGE results for EXABSUMExtractive on DUC2002 dataset while analyzing its features, Comb denotes the combination of the selected scoring techniques……139
Table 4.5 ROUGE results for EXABSUMExtractive and EXABSUMAbstractive on CNN dataset while analyzing its features, Comb denotes the combination of the selected scoring techniques……140
Table 4.6 User satisfaction results for the different text summarization approaches……142
Table 4.7 F-measure results of our proposed approaches against baselines for the DUC 2001 collection on single-document summarization task ……144
Table 4.8 F-measure results of our proposed approaches against baselines on the DUC 2002 collection on single-document summarization task……145

List of Figures

Figure. 2.1 Keyphrases interaction in various TM, IR and NLP tasks……23
Figure. 2.2 Automatic keyphrase extraction process stages……27
Figure. 2.3 Baseline features that characterize a keyphrase……28
Figure. 2.4 Automatic keyphrase extraction ranking approaches……29
Figure. 2.5 Supervised machine learning models……31
Figure. 2.6 Unsupervised approaches for automatic keyphrase extraction……34
Figure. 2.7 Deep learning approaches……41
Figure. 2.8 Evolution of information retrieval systems……54
Figure. 2.9 The Ambisense model of context……55
Figure. 2.10 Nested model of context types centered on Information Objects of an IR system……56
Figure. 2.11 The multi-dimensional concept of context in an IR system (Tamine et al. 2009) ……56
Figure. 2.12 A basic architecture of the contextual IRS (Tamine et al. 2009) ……57
Figure. 3.1 Schematic diagram of HAKE system for extracting keyphrases……73
Figure. 3.2: Schematic diagram of the proposed technique for candidate keyphrases identification……75
Figure. 3.3 Example of parse tree of a sentence……76
Figure. 3.4 Identified candidate keyphrases from a simple sentence……77
Figure. 3.5 The F1 scores of the top 5,10 and 15 keyphrases while varying the weighting parameters on the SemEval2010 Dataset. ……88
Figure. 3.6: The F1 scores of the top 5,10 and 15 keyphrases while varying the weighting parameters on the on the evaluated datasets……89
Figure. 3.7: Effectiveness of HAKE against Baselines. F1@10 SCORE……93
Figure. 3.8: Effectiveness of HAKE on top of large texts (Krapivin2009, SemEval2010, and Wiki20 datasets), and different domains (Nguyen2007 dataset). F@10, P@10, R@10……94
Figure. 4.1 Classifications of the ATS systems……101
Figure. 4.2 ATS approaches and their associated methods……102
Figure. 4.3 The architecture of a hybrid text summarization system.……105
Figure. 4.4 An architecture example of a hybrid text summarization system……107
Figure. 4.5 Taxonomy of summary evaluation measures ……110
Figure. 4.6 $EXABSUM_{Extractive}$ stages for extractive summary……116
Figure. 4.7 $EXABSUM_{Abstractive}$ stages for generating abstractive summary……125
Figure. 4.8 Word graph built from sentences (1-4) and a possible compression path is given……126
Figure. 4.9 Hybrid Summary Process……131
Figure. 4.10 ROUGE-1 and ROUGE-2 results of EXABSUM on the DUC 2001 collection while varying the scoring techniques and summary types……138
Figure. 4.11 ROUGE-1, ROUGE-2, ROUGE-SU4, ROUGE-L results of EXABSUM on the DUC 2002 collection while varying the scoring techniques and summary types……139
Figure. 4.12 ROUGE-1 and ROUGE-2 results of EXABSUMExtractive and EXABSUMAbstractive against various baseline systems on the DUC 2001 dataset……140
Figure. 4.13 ROUGE-1 and ROUGE-2 results of our proposed approach for the large CNN dataset varying the scoring techniques and summary types. ……145
Figure. 4.14 ROUGE-1 and ROUGE-2 results of EXABSUMExtractive and EXABSUMAbstractive against various baseline systems on the DUC 2002 dataset……146
Figure. 4.15 ROUGE-1 and ROUGE-2 results of EXABSUMExtractive and EXABSUMAbstractive against various baseline systems on the large CNN dataset……147

List of Abbreviations

- **IR**: Information Retrieval.
- **IRS**: Information Retrieval Systems.
- **CIRS**: Contextual IR Systems.
- **IE**: Information Extraction.
- **NLP**: Natural Language Processing.
- **ML**: Machine Learning.
- **TM**: Text Mining.
- **AKPE**: Automatic Keyphrase Extraction.
- **ATS**: Automatic Text Summarization.
- **TF-IDF:** Term Frequency Inverse Document Frequency.
- **VSM:** Vector Space Model.
- **LSA**: Latent Semantic Analysis.
- **LSI**: Latent Semantic Indexing.
- **HFSM:** Hybrid Features Selection Method.
- **RF**: Relevance Feedback.
- **POS**: Part of Speech.
- **DNNs**: Deep Neural Networks.
- **DBNs**: Deep Belief Networks.
- **RNNs:** Recurrent Neural Networks.
- **DRNNs**: Deep Recurrent Neural Networks.
- **AE** Automatic Evaluation.

1

General Introduction

1.1 Research Context & Motivations

The pace at which data is generated nowadays and often made available for on-line access to almost everyone has encouraged the introduction and application of intelligent and automated ways to process data. Data items today are generated by an extremely wide audience and range, including geo-locational, acceleration measurement data, mostly due to smart devices. All the diverse and immense data that is continuously being produced is often referred to as big data nowadays. This new scale of data not only called for the creation of novel, distributed techniques with respect to their storage but in their processing as well.

Despite the abundant appearance of novel types of data, it is still extensively produced in the textual format. These textual data items may take many forms, ranging from blog review and forum posts to scientific literature and news portal entries. Documents originating from the different genres and context can differ significantly in their length, degree of structure, writing style, semantic context, and so on. However, what they all have in common is their large size is accessible and that their detailed processing is practically impossible without automatic approaches.

As the amount of digitally available textual data (e.g., reviews, news, blogs, social media content, books, biomedical documents, scientific papers, etc.) continues to grow exponentially, the need for automated techniques to process, retrieve, and synthesize information becomes increasingly important.

One effective way of representing the main contents of documents is by describing them in the form of keyphrases, and synthesizing them in the form of a summary.

Indeed, prospective readers can quickly determine whether a document is relevant to their information need if the significant phrases (keyphrases) or fragments (sentences) in this document are provided. The characteristic of keyphrases for describing and summarizing the contents of documents in a compressed form makes them very crucial for several natural language processing (NLP), Information Retrieval (IR), and Text Mining (TM) tasks.

They can be extremely useful in the categorization, summarization, and retrieval of textual documents. Analyzing the relative importance of keyphrases over time offers the possibility of performing trend detection, and the aggregation of document-level keyphrases to provide keyphrases for multiple documents can be utilized in intelligent visualization tools, as will be demonstrated later on. However, despite their potential utility, on one hand, only a small minority of documents have keyphrases assigned to them, which makes readers consume much time to find the relevant information. On the other hand, users cannot even read and comprehend all the textual content of search results due to many repeated or insignificant portions of texts. Thus, condensing this extracted relevant information in the form of a summary would be very beneficial for several users, and can also be used as content repositories for extracting valuable facts or information. In fact, without summaries, it would be practically impossible for human beings to get access to the ever-growing mass of information available online. Nevertheless, manual keyphrase extraction or manual text summarization is time-consuming and costly, hence methods for their automatization become urgent as well as a pressing need. For this reason, the extraction of keyphrases from documents and text summarization has gained academic interest recently. These concerns have initiated interest in the development of (1) Automatic KeyPhrase Extraction (AKPE) systems to retrieve relevant information, and (2) Automatic Text Summarization (ATS) systems to produce concise and fluent summaries of the most important information while being non redundant, coherent and grammatically readable. Although several AKPE algorithms have been proposed, some of them are domain-dependent (tied to a specific domain while extracting keyphrases), and others have used purely statistical techniques to evaluate their performance. Nevertheless, depending only on the statistical analysis is no longer enough, especially with the growing text resources, where precision becomes more important than recall. Thus, involving or gathering more linguistic and semantic analysis to the statistic ones can enhance the task in general.

Regarding the ATS task, a good text summary should retain the most important content of the original document or a cluster of documents. Generally, ATS approaches can be classified into three categories (1) extractive which identify the important sections of the text and produce them verbatim, (2) abstractive that aims to produce important information that includes words and phrases different from the ones occurring in the source document, and (3) the hybrid one that fuses extractive and abstractive techniques. The need for time is to produce high-quality summaries that are grammatically correct, readable, coherent, concise, and information-rich. However, extractive summarization has been a very extensively investigated topic. Now the research has shifted towards abstractive summarization. Abstractive summarization helps resolve the dangling anaphora problem and thus helps generating readable, concise, and cohesive summaries. Abstractive summarization helps reducing the sentence size as it uses fusion to merge the sentences, thus helps achieving more non-redundancy in the summary as compared to the extractive summaries, where even the non-relevant part of the sentence also gets included due to the fact that it extracts the sentences and arranges them. Abstractive summaries also give higher precision than extractive summaries.

Abstractive summarization techniques not only help for getting good quality summaries for the textual data but also for the non-textual data and cross-language data. The complexity, underlying the natural language text, makes abstractive and hybrid summarization a difficult, challenging, and a new promising research direction.

This thesis will introduce two new methods: the first one for the extraction of keyphrases from textual documents and the second for the text summarization. Both methods perform on various document genres and domains. First, we investigate existing works and approaches proposed for each task in order to highlight the major challenges that such related techniques face and depict the specific complexities they address. Then, we propose two innovative methods for AKPE and ATS. These methods deal with several revealed limitations and challenges of the existing systems. The obtained results are very satisfactory and prove that our methods are effective at increasing the performance of these two difficult tasks. The following section presents the research questions that we addressed through this thesis.

1.2. Research Issues

As we have previously stated, AKPE is designed to extract relevant information, whereas ATS is designed to summarize and produce relevant information in a concise and fluent manner to a reader. To this end, both tasks use various techniques to extract or summarize relevant information.

On one hand, in the AKPE TASK, the use of selection and ranking approaches is quite important, but the challenge is to improve the practices and methods used to make the AKPE system more precise, efficient through different domain contexts and text genres. According to the most recent existing literature in the AKPE task, different research questions (problems) have been raised and should be addressed which can be articulated on the following axes:

- First, **the candidate keyphrase selection issue**. Before doing any operation, a system should firstly generate a list of quality candidate phrases from each document, where a quality phrase means a continuous sequence of words with coherent semantics. This step is considered as a main challenge of AKPE since most systems fail at predicting candidate keyphrases with different sequential orders, or encounter keyphrases that contain synonyms or abbreviations. A novel methodology to select more linguistically informative and grammatically correct keyphrases should be considered.

- Second, **the adopted analysis issue**. Most studies in the field of AKPE have only focused on statistical analysis or use external knowledge bases to select relevant keyphrases. The Linguistic, structural, and semantic analysis must be sought with advanced statistical methods to select more precise keyphrases.

- Third, **the quality of keyphrase properties and overgeneration errors**. Existing unsupervised methods suffer from overgeneration error since they typically identify key" words" that are statically relevant and then return phrases that contain these keywords as keyphrases. Therefore, three questions can be revealed: how to select statically, structurally, and semantically relevant keywords and use them later as a basis to select more relevant keyphrases? What kinds of properties (features) make a sequence of words an informative phrase linguistically? what kinds of properties make a phrase a relevant keyphrase? An approach exploring and using important properties of keywords such as informativeness and positional preference should be considered.

- Forth, **the ranking and the overall performance of the system**. Even though there is a vast literature on the automatic keyphrase extraction task, the existing methods do not achieve good performances. The related performance factors such as (1) incorporating background knowledge: keyphrase extractors need to have a deeper "understanding" of a document in order to reach the next level of performance. Such an understanding can be facilitated by the incorporation of background knowledge; (2) how to handle long documents (to handle a large number of candidates in long documents)? Employing sophisticated features is needed, especially those that encode background knowledge, will enable keyphrases and non-keyphrases to be distinguished more easily even in the presence of a large number of candidates; (3) handling a good performance while ranking relevant keyphrases across different document domains (or topics) and types.

On the other hand, in the ATS task, even with the many advances in information technology, summarization is still a field where improvements are much needed. There are several main problems in text summarization which can be expressed as the following:

- First, **Text Relevancy Detection**. The flow of information in a given document is not uniform, which means that some parts are more important than others. Hence, a common problem is the selection
  of the most relevant information and the distinction of the informative parts from a source document or documents (e.g., based on their relevant keywords or keyphrases). To this end, sentence ranking is a primary step in any ATS. Thus, it is necessary to improve this step and propose new approaches based on effective scoring techniques.

- Second, **lack of coherence and redundancy problems**. The extractive summarization suffers from the lack of cohesion and coherence resulting summaries exhibit, partly due to the redundancy problem (sentence with similar meaning), the wrong link between sentences, and non-resolved co-reference relationships. Therefore, instead of typically selecting the most relevant sentences, a new method must be proposed to produce a final coherent summary without semantic redundancy.

- Third, **the well-formed summaries, and the Abstractive and Hybrid summarization challenge**. Most attention has been paid to the problem of extracting verbatim information from a document, however, research has also been carried out on how to create well-formed summaries. To this end, abstractive or hybrid ATS approaches must be involved. This kind of approach is still a growing and challenging field. So far, it is still not easy to generate an

effective abstractive summary. Some general guidelines and effective approaches should be developed to move from extracts to abstracts summaries and to benefit from the advantages of both ATS approaches.

1.3. The Contribution of The Thesis

The main goal of the research, carried out in this thesis, is to analyze, develop new approaches and techniques for the AKPE and ATS tasks. These approaches will contribute towards the extraction and summarization of relevant information, which can be integrated in a framework of an information retrieval environment. In order to achieve this general goal, the following contributions are described:

- **Survey about AKPE**: one among the objectives of this thesis is conducting a detailed survey of state-of the-art on AKPE, in order to distinguish the strengths and weaknesses of AKPE approaches and techniques (supervised, unsupervised, deep learning, and hybrid) and finally to identify best practices, known and proven that achieve the best performance in term of high accuracy and quality.

- **An overview of the use of context information in IR systems**: we present an overview of the theoretical and experimental works that address the use of context in Information Retrieval IR. The notion of context is defined and explained in detail, including its implications. This overview allows an understanding of how the context is modeled in such an IR field and exhibits some related challenges.

- **State-of-the-art of ATS**: the second objective is to conduct a comprehensive review of the various components and techniques that are used to design and implement ATS systems. It allows us to highlight and understand the various used approaches (Extractive, abstractive, or Hybrid) and the used evaluation methods. Hence, the appropriateness of different used approaches and techniques will be identified, and the best revealed guidelines would be followed.

Both AKPE and ATS studies contribute to these research fields successively:

- **A new AKPE system has been developed:** We propose a novel unsupervised domain-independent keyphrase extraction system. The new system allows us to identify and extract informative and syntactically correct keyphrases, and addresses several APKE challenges. Our system solves:

  - **The candidate keyphrase selection issue**: in order to tackle this issue, our system proposes a novel technique (based on a parse tree approach, part of speech tagging, and filtering) for candidate keyphrases identification and extraction. This technique allows to generate a comprehensive and meaningful list of candidate keyphrases and reduce the candidate set's size without increasing the computational complexity.

  - **The correctness of the adopted analysis, and the overgeneration issues:** another important issue, on which we focused, is the detection of statistically, semantically, and structurally relevant keyphrases. We handled these issues by suggesting a new model that takes full advantage of mining simultaneously hybrid (linguistic knowledge, statistical, structural, and semantic) text features extracted from

documents to select the most relevant keywords that are used later to rank relevant keyphrases. This method computes keyphrases relevance based on their hybrid component keywords scores instead of their only statistical ones. This will avoid the over generation problems and produce quality keyphrases.

- T**he ranking and the overall performance of the system issues:** another common issue in the existing AKPE systems is the ranking and the low performance issues. We handled these issues by putting no restriction on any document length, nor requiring any training on a particular set of documents, nor depending on external corpora, dictionaries, domain, or text size.

- **A new hybrid ATS SYSTEM**. As already stated, abstractive and hybrid summarization is a very challenging task. **Our contributions regarding this task are**:

  - We introduce **a novel hybrid summarization approach** that is able to produce three types of summaries: extracts, which rely on a pure extractive approach, and abstracts, an abstractive-oriented summary, which is the result of an abstractive approach, and hybrid, which is the combination of extractive and abstractive summaries. This approach addresses various challenges of both extractive and abstractive summarization approaches as follows:

    - Contrary to some existing extractive approaches, that rely only on the statistical scoring techniques to produce verbatim of the source document's sentences**. We propose a novel unsupervised extractive approach** that solves the **Text Relevancy Detection issue.** The new approach incorporates statistic and semantic scoring techniques for detecting relevant information while suggesting a new one.

    - **The semantic redundancy is a critical problem in ATS**. To solve this problem, we suggested the use of textual entailment for avoiding semantic and context equivalent redundant information in the resulting summaries.

    - We **address the generation of an abstractive summary challenge** by proposing a new graph-based summarization model that produces robust abstractive summaries. It adapts and extends a novel multi-sentence compression and fusion algorithm, and a keyphrase-extraction based re-ranking method. The proposed approach performs without relying on any training data, nor learn the structure or domain of the document.

    - The choice of the hybrid ATS approach can generate high-quality summaries which leads to satisfied readers. The experimental results are presented to compare the effectiveness of our approach to the most recent ATS techniques (extractive and abstractive).

1.4. Overview of Thesis Organization

The current introductory chapter described the theoretical background of AKPE and ATS tasks, introduced the main focus and research scope, presented the research questions and objectives, and explained the contribution of the thesis.

- In **the first part of Chapter 2**, we present a detailed survey of the state-of-the-art of the AKPE task. The used research methodology for conducting this survey is presented. The common process of this task is highlighted while illustrating its various used approaches (supervised, unsupervised, and Deep Learning) and release techniques. Furthermore, the major challenges that such techniques face are investigated, and the specific complexities they address are depicted. Besides, a comparison study of the best performing techniques is provided, and finally, some focal recommendations to improve each stage of the AKPE process for future works are presented.

- ***In the second part of chapter 2,*** an overview of using context information in IR systems is presented. The notion of context is defined and explained in detail, including its implications. Some practical recommendations are then suggested.

This chapter is concentrated on the theoretical contribution of the thesis. The next two chapters detail the main practical contributions of this research work.

- ***Chapter 3*** introduces the first practical contribution that tackles the AKPE issues, which consists of the 'HAKE' System. The unsupervised domain-independent keyphrase extraction algorithm to identify and extract informative and syntactically correct keyphrases. First, in this chapter, brief background and related works on the task of AKPE are provided.

  Then, the HAKE system and its mains steps are introduced, including the proposed keyphrase identification technique, and the followed methodology to select and rank the relevant keyphrases. Each of the HAKE System stages is explained in detail. In the final part of this chapter, the extensive evaluation experiments are presented, and the finding is discussed***.***

- **Chapter 4** presents the second contribution to tackle the challenge of hybrid ATS.

  - ***The first part of chapter 4*** provides a study of the state-of-the-art about ATS and related works. The various components and techniques that are used to design and implement the ATS systems are presented. Besides, the various used approaches (Extractive, abstractive, or Hybrid) and the evaluation methods are also presented. A discussion of some of the most common drawbacks and challenges of ATS are also presented. Finally, the related works to our proposed approach takes place at the end of this chapter part.

  - ***The second part of chapter 4*** consists of the 'EXABSUM' system. A detailed description of EXABSUM architecture and stages is provided, including the three approaches for generating extractive and abstractive and hybrid summaries, respectively. First, it describes how the extractive approach extracts and ranks the document's relevant sentences and addresses the semantic redundancy issue. Then, the abstractive summarization approach is presented, by introducing the new graph-based summarization model which consists of the novel multi-

sentence compression and fusion algorithm, and a keyphrase-extraction-based re-ranking step. Then the combination of the extractive and abstractive approaches is present trough the hybrid approach. The final part of the chapter consists of the evaluation environment, the experiments carried out, and the results obtained by EXABSUM.

- The last chapter, **Chapter 5,** draws the main summary of this thesis, presents the conclusions derived from the whole research work, and makes suggestions and recommendations about potential improvements for current and future work. In addition, researches and publications derived from this thesis have been also presented.

2

Chapter 2: State of The Art

2.1 Introduction

This chapter deals with the state-of-the-art of Automatic Keyphrase Extraction (AKPE) and the context information retrieval which belong to NLP and IR research fields. It covers reviewing different approaches and techniques used in this regard, different methods of evaluation, challenges, and issues comforted in AKPE.

This chapter presents a survey of the AKPE task. This part clarifies the scientific materials and methods used to overview previous AKPE systems related work. The main findings of previous reviews, the research methodology used for conducting our survey, and the proposed classification of selected studies are all explained in the next sections of this chapter's first part. The key success of any research work is in the validity and usability of the research methodology as will be demonstrated. We also perform the detailed, in-depth analysis of selected studies, comparative analysis. Recommendations are suggested to overcome the limitations identified in current AKPE literature.

The second part of the chapter presents a review of the theoretical and experimental works that address the use of context information in Information Retrieval IR field. First, the notion of context, its importance, implications, and uses in the information retrieval field are highlighted. Then, the salient variety of context categories or taxonomies, modeling context, context uses, and implications to IR are presented. An overview of using context information in the IR systems is presented. The notion of context is defined and explained in detail, including its implications, and some practical recommendations are then suggested.

2.2 IR, IE, NLP, TM, and the AKPE

Due to the exponential growth of textual data and web sources, automatic mechanisms are required to identify relevant information embedded within them. This automation process has called for research fields such as:

- **The Information retrieval (IR):** is the process of obtaining information system resources that are relevant to an information need from a collection of those resources. Information retrieval is the science of searching for information in a document, searching for documents themselves, and also searching for the metadata that describes data, and for databases of texts, images or sounds. Automated information retrieval systems are used to reduce what has been called information overload.

- **The information Extraction (IE):** is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents and other electronically represented sources. In most of the cases this activity concerns processing human language texts by means of natural language processing (NLP).

- **Natural Language Processing (NLP):** aims of studying problems in the automatic generation and understanding of natural language. It is a subfield of linguistics, computer science, and artificial intelligence concerned to process and analyze large amounts of natural language data. The result is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

- **Text Mining (TM):** is the process of deriving high-quality information from text. It involves "the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources". Written resources may include websites, books, emails, reviews, and articles. High-quality information is typically obtained by devising patterns and trends by means such as statistical pattern learning.

The utility of **Automatic Keyphrase Extraction (AKPE)** in these fields cannot be overstated, given its widespread adoption in many Information Retrieval (IR), Natural Language Processing (NLP), and Text Mining (TM) applications, and its potential ability to solve difficulties related to extracting valuable information.

In recent years, a wide range of AKPE techniques has been proposed. However, they are still impaired by low accuracy rates and moderate performance. This section provides a comprehensive review of recent research efforts on the AKPE task and its related techniques. More concretely, we highlight the common process of this task, while also illustrating the various approaches used (supervised, unsupervised, and Deep Learning) and released techniques. We investigate the major challenges that such techniques face and depict the specific complexities they address.

Besides, we provide a comparative study of the best performing techniques, discuss why some perform better than others, and propose recommendations to improve each stage of the AKPE process.

2.2.1 Background and Research Methodology

2.2.1.1 From Keyword to Keyphrase

A keyword is "a single word that is highly relevant". A keyphrase is "a sequence of one or more words that are considered highly relevant" (Hammouda et al. 2005). In IR systems and search engines, keyphrases are much more convenient for users because they can express more precise information. Keyphrases, key terms or keywords have the same roles that characterize and capture the main topics of a single large document, or data collection. Since keyphrases are considered more significant and informative than keywords in documents and search systems, users prefer keyphrases over keywords (Zhang et al. 2004). They not only provide the main idea of the document, but also aid readers in deciding whether to continue further reading or search for additional details. For a single document, keyphrases can serve as a condensed summary (e.g., keyphrases of a journal article). They enable the reader to decide quickly whether the given article is interesting or not. For a collection of documents (e.g., digital libraries, scientific articles, web pages, email messages, magazine articles, business papers, and news reports). Although these known benefits of keyphrases, the manual assignment of them is expensive, time-consuming and difficult especially with large documents or web data. Thus, there is a need to automate this task. The task of Automatic KeyPhrase Extraction (AKPE) is the process of determining the most relevant, descriptive and expressive terms for a given document *(Merrouni et al. 2020)*. The extraction of these terms enables fast and accurate searching of documents and can be applied in many fields such as the **improvement of text summarization** (Zhang et al. 2004), categorization (Hulth and Megyesi 2006), opinion mining (Berend 2011), and document indexing (Gutwin et al. 1999).

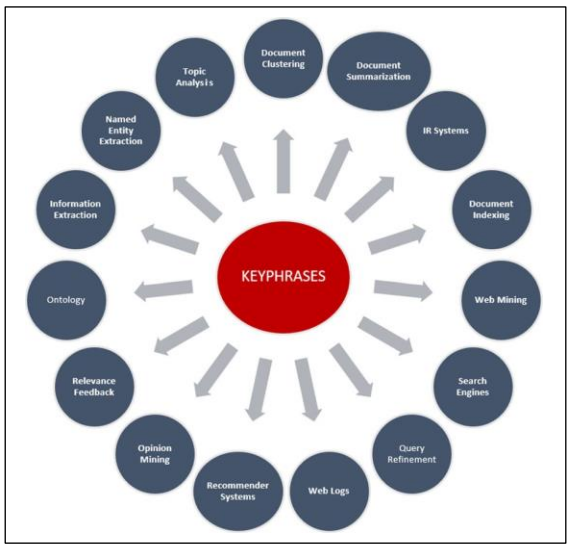
2.2.1.2 Keyphrases in IR, NLP and TM Tasks

Keyphrase extraction is a challenging task where the state-of-the-art performs lower than many core NLP tasks (*Merrouni et al. 2020,* Liu et al. 2010). Hence, AKPE techniques have received much consideration in the past decades due to their importance. Keyphrases are widely used in various TM, IR and NLP tasks (see Figure 2.1). In this section, we describe some uses of keyphrases in these tasks to emphasize on their importance.

- **IR systems:** keyphrases are used extensively to describe documents returned by a query (Jones and Staveley 1999; Moldovan et al. 2000), as the list of keyphrases can help to decide whether a given document is relevant to the user's interest. They improve the functionality of IR systems by using them to build an automated index for document collection, or instead of using them for document representation in classification or categorization tasks (Hulth and Megyesi 2006).

- **Text or document summarization:** given the overload of information that is available on textual databases and the Web, the core application area of keyphrases is focused on Document Summarization (DS). DS involves the creation of an informative summary from a document (or a collection of documents) by distilling and extracting the most relevant parts of the text, which can be presented as a set of keyphrases. Thus, keyphrases are useful in DS. They can serve as a form of semantic metadata indicating the significance of sentences and paragraphs in which they appear (Zha 2002; Zhang et al. 2004; D'Avanzo and Magnini 2005). Web Search Engines present a list of documents and informative summaries (parts of the text matching query terms). Such summaries help the user to decide which documents are relevant. They facilitate users to better understand large amounts of text information, especially when the documents contain various topics.

Figure. 2.1 Keyphrases interaction in various TM, IR and NLP tasks



- **Document indexing:** in large text data (for example Digital Libraries (DLs)), readers face difficulties in locating their desired sections. Keyphrases are widely used in organizing Digital Library holdings and providing thematic access to documents. The DLs are often supplemented with an extra section called the index. This section consists of important topics described in the document and their locations. The index can be built using an alphabetically categorized list of keyphrases, extracted from parts of a single long document or from a collection of documents (Medelyan and Witten 2006; Newman et al. 2012; Gutwin et al. 1999). Moreover, keyphrases can also be used for the task of thesaurus creation for these DLs (Medelyan and Witten 2006).

- **Document clustering:** document clustering is the act of gathering similar documents into groups (or clusters), where similarity is decided by functions (for example Euclidean Distance) on these documents. Text clustering can be in different levels of granularities where clusters can be documents, paragraphs, sentences, keyphrases or terms. Text clustering approaches based on key phrases help to achieve more effective clustering results (Zhang and Dong 2004; Osinski et al. 2004; Hammouda et al. 2005; Han et al. 2007; Liu et al. 2009). Indeed, using keyphrases as document features has several benefits: (1) it can improve the clusters' quality by leveraging more information presented in the document, (2) it is helpful in building concise and accurate descriptions (labels) for the produced clusters (Zamir and Etzioni 1998), (3) it can aid in producing clusters that group the documents that are relevant to the user's query apart from the irrelevant ones (Zamir and Etzioni 1998).

- **Web mining:** the rapid growth of the Web has led to an increase in the web data (e.g. page contents, usage logs, and web hyperlinks). Web mining aims to discover and extract valuable information from these various types of data. For example, using keyphrases in web mining techniques can aid in extracting document topics and ranking web documents. Thus, users can gather information both efficiently and comfortably (Turney 2003; Chen et al. 2005).

- **Search engines, Web logs:** keyphrases are used beneficially in (1) Search Engines (SE) which help to extract information through pattern matching, using keyphrases or keywords for a given user query. SE provides a conceptualization of a large text collection. Thus, keyphrases of web pages can serve as metadata for indexing and retrieving these web pages efficiently. And (2) weblogs, where keyphrases can be used to analyze the usage patterns of a user and his or her trends and interests (Gong and Liu 2009).

- **Query expansion, Query suggestion, Query refinement, Relevance feedback:** Query expansion, query suggestion, and query refinement are some of the most well-known query modification techniques in IR systems and search engines. They can add, suggest, reformulate, refine or transform relevant terms (to form keyphrases) that will aid the user in removing the ambiguities of natural language and express the information idea in a more detailed way into the original query (Krovetz and Croft 1992; Chandrasekar et al. 2006; Song et al. 2006). The majority of these techniques are based on the user relevance feedback or the result list of the previous query. Relevance Feedback (RF) is a proven approach for expanding a query by selecting important keyphrases or expressions attached to documents retrieved from the original query. Users in the result set then marked them as relevant, or the system deduced that the top-ranked documents were relevant (Song et al. 2006).

- **Recommender systems, opinion mining:** Content-Based Recommender Systems (CBRS) (Lops et al. 2011; Ferrara et al. 2011) and Advertisement Targeting (AT) (Yih et al. 2006) are based on keywords automatically extracted from the text of the page. They help users in discovering relevant information to their previously expressed interests. Opinion mining and sentiment analysis deal with the treatment of opinion, sentiment, and subjectivity in the text. Keyphrases aid in representing and understanding these opinions (for example, in web reviews, how an opinion-holder reacts toward a given product) (Berend 2011).

- **Ontology**: ontologies ease knowledge sharing and re-use. More recently, the notion of ontology has become widespread in information retrieval and knowledge management. Keyphrases are used to facilitate: the automatic construction of concept maps (Leake et al. 2003), ontology learning and building (Fortuna et al. 2006; Frikh et al. 2011; El Idrissi et al. 2014; Do and Ho 2015). Precisely, they help in identifying relevant concepts and provide a basis for building ontologies.

- **Information extraction and Named entity extraction**: Information Extraction (IE) is the task of automatically extracting information from unstructured or semi-structured documents. Such information can be present as a set of keyphrases or a block of text that include various named entities in different domains (e.g., biomedical, News, Business, Politics, etc.) (Dostal and Jezek 2011). A named entity is a term or phrase that identifies a real-world object. These entities categorize people, geographic locations, addresses, organizations, products, etc.

- **Topic analysis**: due to the exponential growth of textual data, the complexity of document corpora has led to considerable interest in using topics to facilitate tasks like browsing and searching. The topic represents an underlying semantic theme of a document. Topic keyphrases briefly describe and represent the key content of topics. They help users grasp the key content of a topic and make the decision to keep reading the document or not (Liu et al. 2009; Grineva et al. 2009; Bougouin et al. 2013).

Additionally, AKPE task is used in several real-world domains such as medicine (Sarkar 2013; Liu et al. 2015), agriculture (El-Beltagy and Rafea 2009), national security and terrorism (Elovici et al. 2010), social media (Matsuo et al. 2007; Mori et al. 2007), and law (Jungiewicz and Lopuszynski 2014).

2.2.1.3 Research Methodology and Criteria

Relevant works are retrieved by querying multiple electronic databases (see Table 2.1). In addition, most of the related high-profile journals, conferences and workshops (see Table 2.2) are investigated manually to analyze AKPE works from different perspectives.

More generally, the selection process in both databases and target journals-conferences-workshops consists of three steps: (1) a set of major keywords are used including: “keyphrase extraction” or “automatic keyphrase extraction”, next, (2) formal searches are performed sequentially on (i) the selected databases, and (ii) the target domain journals and conferences. Then, (3) in each formal search, titles, abstracts, and full-texts of potential works are analyzed against predefined criteria to decide whether each paper should be included or not.

Table 2.1 Electronic databases investigated in this survey

#	Electronic databases
EDB1	Science Direct
EDB2	IEEE Xplore
EDB3	ACM Digital Library
EDB4	SpringerLink
EDB5	Google Scholar

Effectively, articles and conferences papers are reviewed and analyzed to ensure that they are, in fact, present theoretical or/and practical solution (s) to this task. More than 200 papers are collected when applying the first step. Given this scale of papers, a presentation of every existing AKPE work in this survey is not possible. The second step was to identify additional criteria to lessen the number of selected papers.

Therefore, papers from the past two decades are considered, focusing on the most referenced, relevant, and recent case works. More precisely, works that implement an AKPE technique in TE, NLP or IR domains are prioritized. Applying these criteria, a total of 34 techniques are analyzed and listed chronologically, and more than 85 references are chosen to conduct the overall survey.

Table 2.2 Some target journals-conferences-workshops investigated in this survey

#	*Name*	*Publisher*
J1	Knowledge and Information Systems Journal	Springer
J2	Information Retrieval Journal	Springer
J3	Information Systems Journal	Elsevier
J4	Journal of The Association for Information Science and Technology	Wiley Online Library
J5	Information Processing & Management	Elsevier
J6	Information Sciences	Elsevier
C1	Symposium on Document Engineering (DocEng)	ACL
C2	International Conference on Computational Linguistics (COLING)	ACL
C3	International Conference on World Wide Web (WWW)	ACM
C4	International Conference on Web-Age Information Management (WAIM)	Springer
C5	International Conference on Data Mining ICDM	IEEE
C6	International ACM SIGIR Conference on Research and Development in Information Retrieval	ACM
C7	Conference on Empirical Methods in Natural Language Processing (EMNLP)	ACL
C8	ACM/IEEE-CS Joint Conference on Digital libraries (JCDL)	ACM
WS1	International Workshop on Semantic Evaluation (SemEval)	ACL
WS2	Workshop on Multiword Expressions (MWE)	ACL

2.2.2 The Automatic Keyphrase Extraction Task

2.2.2.1 The Automatic Keyphrase Extraction Process

As previously stated, the task of AKPE is defined as the process of automatically extracting relevant, descriptive and expressive phrases from a document or web data. Generally, most AKPE systems identify a set of words and phrases called “candidates” that could convey the topical content of a document, then these candidates are scored and ranked. Finally, the “best” ones are selected as a document’s keyphrases. A typical AKPE process consists of five main stages (See Figure. 2.2): (1) pre-processing the input data, (2) identification of candidate phrases, (3) feature selection and scoring, and (4) keyphrase ranking. Depending on the goal of every system, the ranked extracted keyphrases can be presented in two sets: single document keyphrases or document collection keyphrases, and finally (5) the evaluation of the extracted keyphrases. Figure 2.1 depicts the overall process of automatic keyphrase extraction, whose main stages are analyzed and explained in the following subsections.

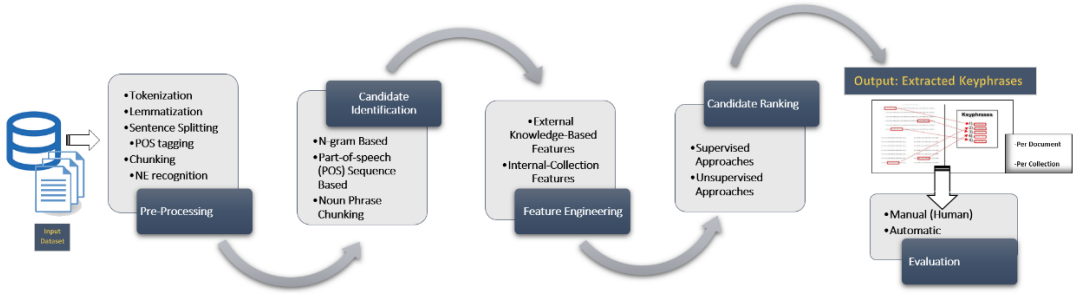


Figure. 2.2 Automatic keyphrase extraction process stages

2.2.2.2 Documents Pre-Processing and Candidate Phrases Identification Stage

The document(s) preprocessing stage consists of formatting the document (s) into a machine-readable format to decreasing its (their) complexity. This requires some text operations including sentence segmentation, stemming (reduction of word to grammatical root), tokenization (converting a sequence of characters into a sequence of tokens), part-of-speech tagging, noisy symbols and stop words removal, and named entity recognition (that seeks to locate and classify named entity mentions in text into predefined categories such as names, locations, quantities, time expressions, values). After this stage, candidate phrases are identified from the full-text document. A set of phrases are typically extracted as candidate keyphrases using heuristic rules which aim to reduce the number of false-positive candidates while maintaining the true-positives. To detect all candidate keyphrases, most methods usually fall into three categories: (1) N-Gram based, (2) Part-Of-Speech (POS) sequence based, or both. Category (1) was adopted in the pioneer and earlier systems (Witten et al. 1999; Turney 2000), where the input text is separated according to phrase boundaries (e.g., punctuation marks) and to a limited length (bigram, trigram, etc.). Then, some simple rules are applied to candidate phrases to filter meaningless sub-sequences (for example, candidate phrases cannot begin or end with a stop word) (Huang et al. 2006; Liu et al. 2009). The commonly used category (2) allows words with definite part-of-speech tags (for example, adjectives, nouns, verbs) to be candidate keywords by using a POS tagger (Barker and Cornacchia 2000). The noun phrase-chunking category is also performed especially when we search for text chunks matching to individual noun phrases. Hulth (2003) compares three different phrase identification methods and concludes that extracting noun phrase- chunks and word sequences (that match any set of a POS tag pattern) both give better precision than n-grams. Grineva et al. (2009) use n-grams that appear in Wikipedia article titles as candidates. Haddoud et al. (2015) identified five existing POS tag sequence definitions of a noun phrase and

proposed a new POS tag sequence that shows significant improvements over other filters. In case the results of this stage generates a long list of candidates, Huang et al. (2006), Kumar and Srinathan (2008), El-Beltagy and Rafea (2009), You et al. (2009), and Newman et al. (2012) have designed different pruning heuristics to limit candidates that are unlikely to be keyphrases. Using these pruning heuristics increases the recall rates in extracting relevant keyphrases.

2.2.2.3. Feature Engineering Methods

The feature engineering step involves developing properties that could characterize individual keyphrase candidates from other terms. These properties are called features. Features perform in both (1) supervised learning approaches (feature weights rely on priors or statistics attained in the training process), such as *tf–idf*, Accuracy, F1-measure, Mutual Information, Term Strength, Chi-Square, Information Gain, and Odds Ratio, and in (2) unsupervised learning approaches (features which are extracted without training like structural rules (tags or positions) or frequency-based information).

Regardless, features have been used extensively as an instance for super-vised keyphrase extraction and reflect how well a candidate phrase represents the topic and content of the document. Practically, several features are calculated for each candidate phrase to measure its importance during the feature engineering stage. The majority of the used features combine: (1) frequency statistics within a single document and across an entire collection, (2) semantic similarity among keyphrases (i.e., keyphrase cohesion), (3) popularity of keyphrases among manually assigned sets, (4) heuristics, such as locality and the length of phrases (Medelyan and Witten 2006), and (5) lexical and morphological analysis (for example, hyphenated phrases and morphological affixes). Usually, features are classified according to their origin (see Figure. 2.3): phrase-level features (such as structural features, syntactic and morphological ones), document-level features (e.g. statistical and baseline set features within a single document), corpus-level features (applied across an entire collection), and external knowledge-based features (such as article data-banks banks (Wikipedia[1]), terminological databases (MeSH[2], GRISP[3], etc.), linguistic resources (WordNet[4], etc.), and Web Search Query Log.

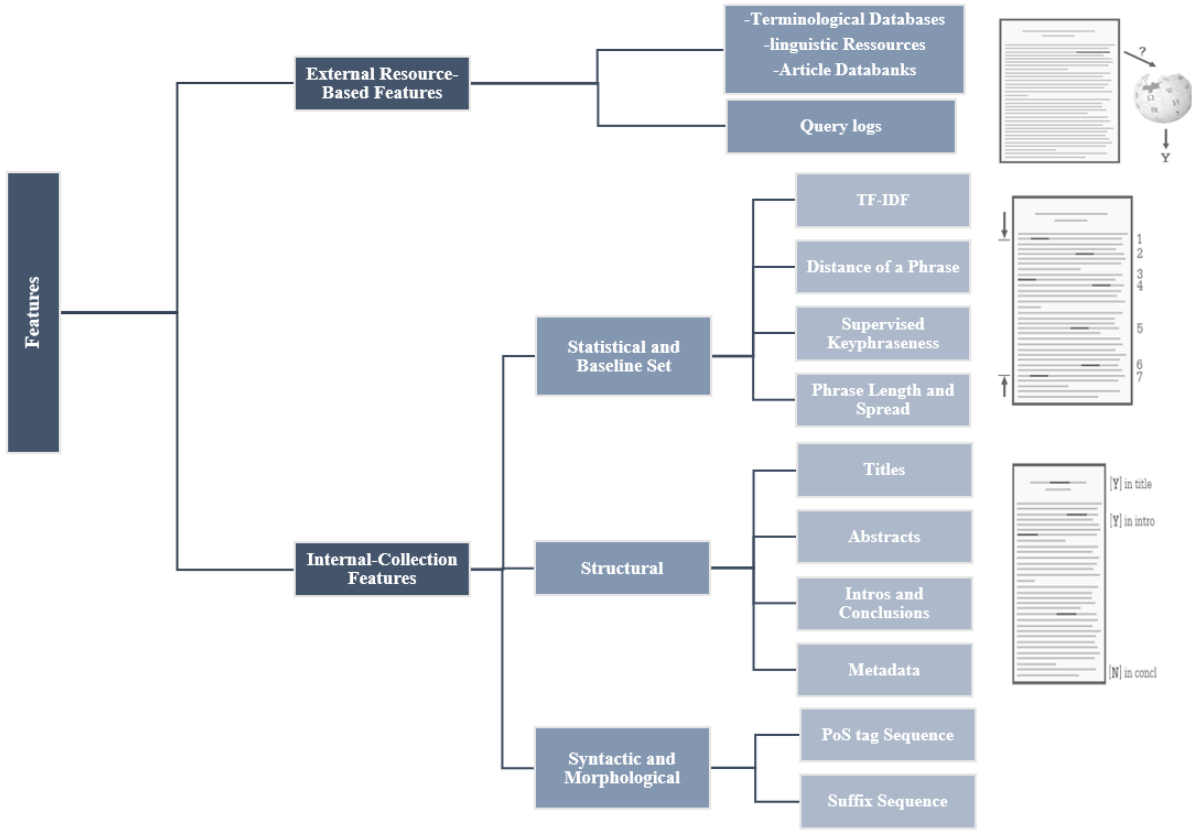


Figure. 2.3 Baseline features that characterize a keyphrase

[1] https://en.wikipedia.org/

[2] https://www.ncbi.nlm.nih.gov/mesh

[3] https://hal.archives-ouvertes.fr/inria-00490312/en/

[4] https://wordnet.princeton.edu/download

The most popular features are: tf–idf (that select those candidate phrases which are frequent in a given document, but less frequent in the whole document collection) (Frank et al. 1999; Witten et al. 1999; Nguyen and Kan 2007; Liu et al. 2009), and first occurrence (Witten et al. 1999) which are often used as a baseline by most existing AKPE systems. The KEA++ system (Medelyan and Witten 2006) adds the feature of phrase length in words and the node degree (representing the number of thesaurus links that connect the term to other candidate phrases) as a refinement of KEA system. Keyphrase cohesion is another extensively-used feature. Turney (2003) introduces two new sets of features and measures keyphrase cohesion (the statistical association) within the top-N keyphrase candidates against the remaining candidates using web frequencies. Context, document type, and structure are also considered boosting information for feature calculation. Zhang et al. (2006) propose to use not only 'global' but also 'local' context information, such as the linkage feature. Kumar and Srinathan (2008) use the position of sentences and position of phrases in a sentence as a feature. Chen et al. (2005) use three new features considering the structure of web pages. In scientific publications, Nguyen and Kan (2007), Kim and Kan (2009), Lopez and Romary (2010), and Nguyen and Luong (2010) propose the section occurrence vectors as a feature. They are based on the idea that keyphrases are distributed non-uniformly in different logical sections of a paper, prefer-ring sections such as abstracts, titles, introductions, and related works. Berend and Farkas (2010) proposed features regarding the document level, the corpus level, and the knowledge-based level. They rely on external knowledge sources too (i.e., Wikipedia) to achieve additional enhancements in the system. Based on the semantic relations among candidate phrases, Kelleher and Luz (2005) used the "Semantic Ratio" (SR) feature that divides the number of occurrences of a phrase in the present document by the number of times it occurs in all documents directly linked to that document.

Huang et al. (2006) treat each document as a semantic network and analyze its connectedness and compactness to symbolize the importance of a node, which represents a candidate phrase. Wan and Xiao (2008) use a set of neighborhood documents to improve the single document keyphrase extraction. Other authors have used the term co-occurrence of candidates (Mihalcea and Tarau 2004; Matsuo and Ishizuka 2004; Liu et al. 2009).

2.2.3. Review of Supervised, Unsupervised and Deep Learning Ranking Approaches

Keyphrase extraction is generally considered a ranking problem (i.e., keyphrase candidates are ranked based on their feature values, and the top-N ranked ones are returned as keyphrases). The ranking approaches aim to develop machine-learning models that determine which of the candidate keyphrases generated in a prior stage are relevant and then rank them. Three learning approaches can be used to achieve this aim: the supervised, unsupervised, and deep learning (see Figure. 2.4).

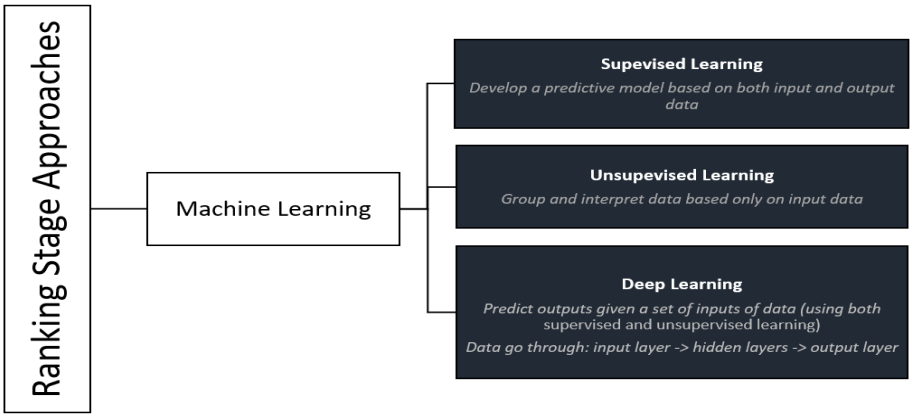


Figure. 2.4 Automatic keyphrase extraction ranking approaches

2.2.3.1 Supervised Approaches

The supervised approaches keyphrase extraction is considered a binary classification problem (Frank et al. 1999; Turney 2000). To cope with this kind of situation, different supervised machine learning (ML) models are used (see Figure. 2.5). In ML terminology, the phrases in a document are "examples" and the learning problem is defined as a mapping from the examples to the two classes: "keyphrase" and "not-keyphrase". Supervised ML models can automatically generate this mapping if they are provided with a set of training examples. Hence, by training a classifier on documents annotated with keyphrases, we can determine whether a candidate phrase is a keyphrase. Different supervised ML models are used such as: bagged C4.5 (Turney 2000), Naïve Bayes (Frank et al. 1999), neural networks (Kamal Sarkar Mita Nasipuri 2010), SVM (Jiang et al. 2009), maximum entropy (Yih et al. 2006). Below we examine some representative AKPE systems that adopt the supervised approach.

- KEA System (Witten et al. 1999): in KEA, the candidate keyphrases are identified using lexical methods, then, for each candidate phrase, a set of features is calculated. Finally, a machine-learning model (Na¨ıve Bayes) is used to predict and classify can-didates as good keyphrases or not. This model is used on a candidate phrase with the following feature values: (1) t (for tf–idf) a measure of a phrase frequency in a document compared to its rarity in general use (general usage is represented by document frequency which is the number of documents containing the phrase in some large corpus), (2) d (for distance) first occurrence which is calculated as the number of words that precede the phrase's first appearance, divided by the number of words in the document.
  The result can be a number between 0 and 1 that represents how much of the document precedes the phrase's first appearance. Generally, the model determines the overall probability that each candidate is a keyphrase, and then a post-processing operation selects the best set of keyphrases. Two probabilities are calculated: P [yes] (the probability that a candidate phrase is a keyphrase) and P [no] (the probability that a candidate phrase is not a keyphrase), calculated similarly (1):

$$P[yes] = \frac{Y}{Y+N} P_{TF\times IDF}[t|yes] P_{distance}[d|yes] \quad (1)$$

  where $P_{TF\times IDF}[t|yes]$ is the proportion of positive examples that have a discrete $TF \times IDF$ value, $P_{distance}[d|yes]$ is the probability of the first appearance of a keyphrase into the document. Y (that is, the author-identified keyphrases) is the number of positive instances in the training files, and N (that is, candidate phrases that are not keyphrases) is the number of negative instances (the Laplace Estimator is used to avoid zero probabilities), this simply replaces Y and N by Y + 1 and N + 1. The overall probability (2) that the candidate phrase is a keyphrase can then be calculated and ranked according to this value:

$$p = \frac{P[yes]}{P[yes]+P[no]} \quad (2)$$

  However, the KEA system depends on the training set and may provide poor results when the training set does not properly fit the processed documents. KEA also has a limited configuration of keyphrase selection and filtering. Moreover, it does not offer capabilities for changing the entire pre-process or adding filters.

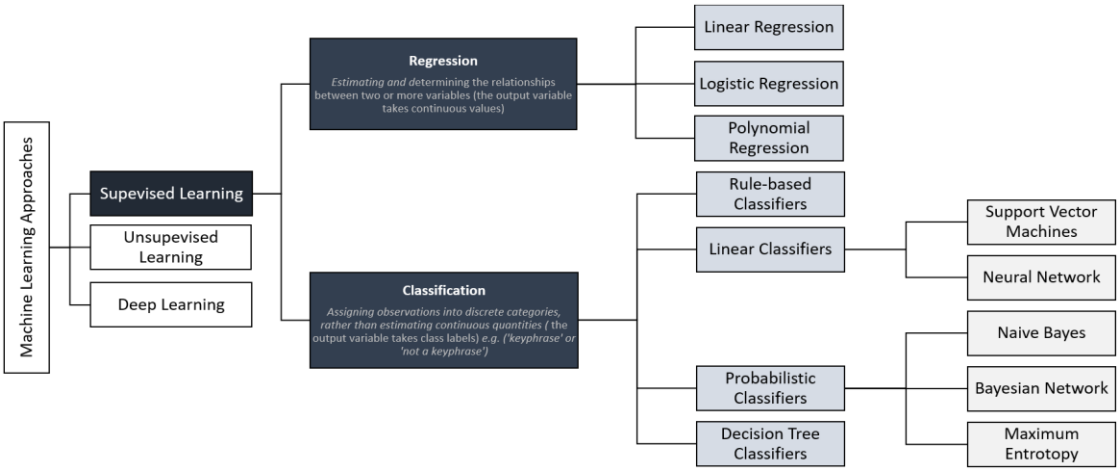


Figure. 2.5 Supervised machine learning models

- GenEx Algorithm, Turney (2000): GenEx, is one of the most representative AKPE techniques. It is based on a genetic algorithm which optimizes the number of correctly identified keyphrases in the training documents. GenEx has two components: The Genitor genetic algorithm and the Extractor keyphrase extraction algorithm. Extractor takes a document as input and produces a list of keyphrases as output. It uses three features: term frequency (tf), phrase length, and first occurrence information. It is based on the C4.5 decision-tree-like process that achieves both candidate selection and weighting. The parameters of the Extractor are tuned by the Genitor genetic algorithm, to maximize performance (fitness) on the training data. GenEx has the ability to preserve its performance across different domains by not using any domain-dependent attributes for classification and weighting purposes. Later, Turney (2003) proposed to model the coherence of an entire set of candidate phrases.

  This approach uses the degree of statistical association among candidate keyphrases as evidence that they may be semantically related, which is done by using the pointwise mutual information (PMI) between a candidate and *k* previously selected phrases. The PMI can be used in conjunction with a web search engine in order to exploit the extra-large corpus. Experiments demonstrate that this enhancement improves the quality of the extracted keyphrases (Turney 2003). However, it is difficult to obtain the PMI of these sets without using sufficiently large datasets. Additionally, calculating the coherence between features is time-consuming.

- Hulth System (Hulth 2003): unlike the KEA and GenEx systems, the Hulth system places no limit on the length of the extracted keyphrases. It uses four different features in conjunction with a bagging technique. These features are term frequency, collection frequency, the relative position of the first occurrence of a term and the part-of-speech tag of a term. In the identification of candidates, Hulth considers the 56 most frequently occurring POS tag sequences among keyphrases in the training data. Hulth argues that using a POS tag as a feature significantly improves the results of keyphrase extraction, as certain POS-patterns are more likely to denote keyphrases. Experimentation carried out by Hulth has shown that using a combination of several prediction models applied on noun phrase candidates produced the best results. Unfortunately, in the Hulth system there is no relation between the different POS tag feature values, (2) this system has not been evaluated on any of the GenEx and KEA datasets, and (3) the recall value reported in this work is low. Consequently, there is no common basis for comparing the results of the three systems (Hulth, KEA, and GenEx).

- Huang et al. Algorithm (2006): Huang et al. propose an AKPE algorithm using two novel weight features that can be adapted in both supervised and unsupervised tasks. This system performs well on digital book datasets. However, due to the lack of phrase redundancy in short documents, the system does not perform well with all types of documents.

- Maui Algorithm (Medelyan et al. 2009): Maui is a general algorithm for automatic top-ical indexing, which is based on the KEA system (Frank et al. 1999). Maui extended the KEA system to integrate information from Wikipedia. Two stages are performed in this system: candidate selection and machine learning. In the first stage, Maui determines textual sequences defined by orthographic boundaries and splits these sequences into tokens. Then all n-grams up to a maximum length of 3 words that do not begin or end with a stop word are extracted as candidate tags. In the second stage, several features are computed for each candidate, then, they are combined in a machine-learning model to obtain the probability that the candidate is indeed a tag. Maui added a novel feature set that consists of document-specific "keyphraseness", a knowledge-based feature that represents the likelihood of a phrase being a link in the Wikipedia corpus. However, a shortcoming of Maui is the lack of any evaluation capabilities.

- Core Word Expansion Algorithm (You et al. 2009): You et al. developed and evaluated an AKPE technique for scientific documents. This technique can be adapted to both supervised and unsupervised tasks. The core word expansion algorithm is used to generate new candidate phrases. First, a core word set is used to localize potential positions of keyphrases, then, candidate phrases are obtained by expanding the core words. In the feature calculation step, You et al. introduce an *idf* related feature for selecting the proper granularity when a phrase and its sub-phrases coexist as candidates. Experimental results show the efficiency and effectiveness of the refined candidate set and demonstrate an overall performance of this system compared with KP-MINER (El-Beltagy and Rafea 2009) and KEA (Witten et al. 1999) systems.
- HUMB System (Lopez and Romary 2010): In HUMB, three kinds of features have been used: (1) structural features (the position of a term with respect to the document structure for each candidate, and the position of the first occurrence), (2) the content features (Phraseness, Informativeness, Keywordness) and finally, (3) the Lexical/Semantic features. In the ranking stage, HUMB combined 3 learning models with boosting and bagging techniques. These models are the decision tree (C4.5), Multi-Layer Perceptron (MLP) and Support Vector Machine (SVM). The selected model for the final run was bagged decision tree. However, HUMB used external knowledge bases (GROBID/TEI, GRISP, and HAL) which are associated with scientific domains.

- DPM-index (Haddoud 2014): Haddoud et al. define the document phrase maximality index (DPM-index), a new measure to discriminate overlapping keyphrase candidates in a text document. They developed a supervised learning system that uses 18 statistical features, among them the DPM-index and five other new features which are transformed into input attributes for a logistic regression learning algorithm. This system offers the possibility of evaluating the efficiency of keyphrase candidate selection based on linguistic features. By not using any external knowledge or structural document features, the results of this system demonstrate an improvement over almost all the other compared systems. Later Haddoud et al. (2015) investigate the impact of candidate term-filtering using linguistic information on the accuracy of AKPE from scientific papers. They proposed a new definition for personalized tag sequence.

- The DPM-index system showed prominent results on the SemEval-2010/Task-5 dataset[5] (a workshop on Semantic Evaluation) (Kim et al. 2010) compared to six of the finest systems (Turney 2000; Kim and Kan 2009; Liu et al. 2009; Nguyen and Kan 2007; Lopez and Romary 2010).

- CeKE model (Gollapalli and Caragea 2014): CeKE is a supervised citation-enhanced keyphrase extraction model. CeKE built on a combination of novel features that capture information from citation contexts and existing features from previous works. The novel designed features consider citation network information to build supervised models that classify (using Naïve Bayes classifiers) candidate phrases as keyphrases, or non-keyphrases. They divided features into three categories: (1) existing features for keyphrase extraction including: tf−idf , relative position, POS tag, (2) other novel features like: Citation Network Based (inCited and inCiting (Boolean features that are true if the candidate phrase occurs in cited and citing contexts)), Citation tf−idf feature (the tf−idf score of each phrase computed from the citation contexts), and (3) other features that include: first position of a candidate phrase, tf−idf − Over (a Boolean feature, which is true if the tf−idf of a candidate phrase is greater than a threshold 0), firstPosUnder (a Boolean feature which is true if the distance of the first occurrence of a phrase from the beginning of a target paper is below some value). Later, Bulgarov and Caragea (2015) showed that adding the keyphraseness feature (related to how often a candidate phrase appears as a tag or a keyphrase in the training dataset) to document textual content and textually-similar neighbors features, improves the keyphrase extraction results. This CeKE+ keyphraseness model shows interesting results in the authors' experiments compared to other systems.

- KeyEx Method (Xie et al. 2017): is a sequential pattern mining based document-specific keyphrase extraction method. The key innovation of this method is to use wildcards (or gap constraints) to help in extracting sequential patterns, so the flexible wildcard constraints within a pattern can capture semantic relationships between words, giving the system full flexibility to discover different types of sequential patterns as candidates for keyphrase extraction (Xie et al. 2017). This method discovers a rich set of keyphrase candidates and further uses a supervised learning approach to build a classification model for keyphrase extraction.
  The author's experiments showed that the KeyEx system is effective in improving the quality of the extracted keyphrases. Furthermore, their method outperforms other sequential pattern mining methods (Xie et al. 2017).

2.2.3.2 Unsupervised Approaches

The unsupervised approaches: in unsupervised approaches, the keyphrase extraction task is considered a ranking problem and performs without prior knowledge. These approaches are grouped as either statistical-based or graph-based (see Figure. 2.6).

[5] http://semeval2.fbk.eu/semeval2.php?location=data

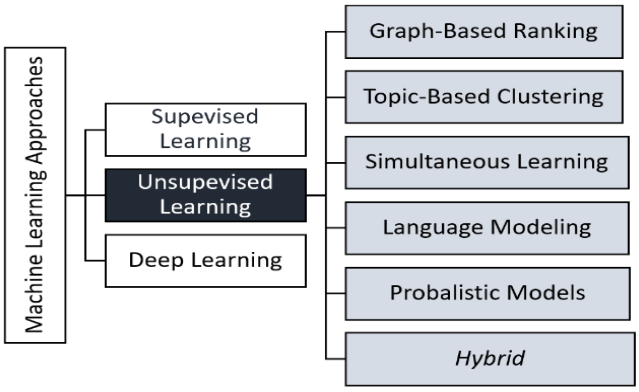


Figure. 2.6 Unsupervised approaches for automatic keyphrase extraction

In statistical-based models, texts are usually represented as matrices in which the statistical metrics are applied to rank the words, such as *tf–idf* term weighting metric (Zhang et al. 2006; Liu et al. 2009). Matsuo and Ishizuka (2004) and Frikh et al. (2011) present a study that applies word co-occurrence distribution. Frantzi et al. (1998) combined linguistic and statistical information to extract technical terms from documents in digital libraries.

In recent years, many graphical representation models of the text have been raised. The graph-based ranking models are based on the idea of “recommendation” or “voting”. They build a graph from the input documents. Each document is represented as a graph where vertices or nodes represent phrases and/or words and the edges are connected based on lexical or semantic relations, such as a co-occurrence relation (Mihalcea and Tarau 2004). Nodes or vertices are then ranked using graph centrality measures such as PageRank and its variants (Page et al. 1999; Mihalcea et al. 2004; Wan and Xiao 2008; Gollapalli and Caragea 2014). These measures assign weights to the node-words reflecting their semantic significance in the text. In Probabilistic models, candidate keyphrases are typically scored based on the simple multiplication of feature values (Frikh et al. 2011; Liu et al. 2009). To rank candidates, other authors include the PageRank family measures (Mihalcea et al. 2004; Wan and Xiao 2008; Gollapalli and Caragea 2014).

Hybrid models combine two or more models in a single model such as ’probabilistic and graph-based’ to take advantage of two or more models (Danesh et al. 2015). In topic based clustering models, grouping the candidate keyphrases is highly suggestive of an underlying semantic theme, which is called a topic. Based on the assumption that candidate keyphrases that are related to a topic are more likely to occur together, it is possible to attribute phrases to a specific topic (Liu et al. 2009; Grineva et al. 2009; Bougouin et al. 2013).

Various methods have been pro-posed concerning these models such as Latent Dirichlet Allocation (LDA) (Blei et al. 2003), which is a representative of topic models, Latent Semantic Analysis (LSA) (Landauer et al. 1998) and probabilistic LSA (pLSA) (Hofmann 1999). In language modeling models, several probabilistic models have been developed to assign a probability to each sequence of words in a document, and assign a probability for the likelihood of this given sequence to follow a sequence of words (Tomokiyo and Hurst 2003).

In simultaneous learning models, the text summarization and keyphrase extraction tasks can benefit from one another if they are performed simultaneously.

Consequently, they will make full use of the reinforcement between summary and keyphrases (Zha 2002; Wan et al. 2007). Below are some representative AKPE systems that adopt the unsupervised approach.

- Tomokiyo and Hurst Approach (Tomokiyo and Hurst 2003): Tomokiyo and Hurst pro-pose an approach that extracts keyphrases based on statistical language models by using the pointwise KL-divergence between multiple language models. Two features have been used in this system: (1) phraseness: describes the degree to which a given word sequence is considered to be a phrase, and (2) informativeness that refers to how well a phrase captures or illustrates the key ideas in a set of documents. Intuitively, a phrase with high phraseness and informativeness scores is likely to be a keyphrase. However, the results of this approach are difficult to evaluate. Moreover, this approach provides no qualitative evaluation and uses a language model instead of heuristics to identify phrases. Consequently, the system's performance is decreased.

- The TextRank Algorithm (Mihalcea and Tarau 2004): TextRank algorithm is one of the most well represented graph-based models for AKPE. It represents a document as a graph. Each vertex in the graph corresponds to a word, and there is an edge between any two words occurring together in the text. A weight, w_{ij}, is assigned to the edge connecting two vertices, ϑ_i and ϑ_j, and its value is the number of times the corresponding word co-occur within a window of W words in the document. TextRank adapts the original PageRank (Brin & Page ,1998) algorithms to calculate word ranks. The original PageRank algorithm works on directed unweighted graphs, G = (V, E). Let In(ϑ_i) be the set of vertices that point to a vertex vi, and Out(ϑ_i) be the set of vertices to which ϑ_i point, the score (3) which reflects the importance of ϑ_i is calculated by PageRank as:

$$S(\vartheta_i) = (1-d) + d \times \sum_{J \in In(\vartheta_i)} \frac{1}{|Out(\vartheta_j)|} S(\vartheta_j) \quad (3)$$

  where d is a damping factor that can be set between 0 and 1, which has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph. In TextRank, the in-degree of a vertex equals to its out-degree, since the graph is undirected. Formally, let d denote a document, and *w* denote a word, then $d = \{w_1, w_2, ..., w_n\}$.

  The weight (4) of a vertex calculated by TextRank is:

$$WS(v_i) = (1-d) + d \times \sum_{\vartheta_j \in In(\vartheta_i)} \frac{w_{ij}}{\sum_{v_k \in Out(v_j)} w_{jk}} WS(v_j) \quad (4)$$

  where w_{ij} is the strength of the connection between two vertices ϑ_i and ϑ_j, and d is the dumping factor, usually set to 0.85 (Brin and Page, 1998). Intuitively, a vertex will receive a high score if it has many high-scored neighbors. As noted, before, after convergence, the top-scored vertices are selected as keywords. Adjacent keywords are then collapsed and output as a keyphrase.

TextRank can be run without syntactic filtering during the formative stages of document representation, and therefore, all words can be considered.
However, the TextRank does not make full use of the statistical information, such as the length and the position of the phrase, and its best score is achieved when only nouns and adjectives are used to create a uniformly weighted graph for the text under consideration.

- CorePhrase Algorithm (Hammouda et al. 2005): CorePhrase is an algorithm for topic discovery that uses keyphrase extraction from multi-document sets, and clusters that are based on frequent and significant shared phrases between documents. CorePhrase (1) first constructs a list of candidate keyphrases by comparing every pair of documents (using the Document Index Graph (DIG) (Hammouda and Kamel 2002), (2) it scores each candidate keyphrase according to some criteria (df: document frequency, aver-age weight, average phrase frequency, and average phrase depth), (3) ranks keyphrases according to their score, and finally selects a number of the top ranked keyphrases for output. CorePhrase achieves high accuracy in identifying the topic of a document cluster on a web dataset. Compared with other keyword-based cluster labeling algorithms, CorePhrase did not achieve any significant improvements in performance.

- SingleRank Algorithm (Wan and Xiao 2008): SingleRank algorithm follows the same approach as the TextRank algorithm but differs in the following aspect: while in Tex-tRank, phrases containing the top-ranked words are selected, SingleRank does not filter out any low-scored words (vertices). Each candidate key phrase in SingleRank is given a score by summing the scores of its constituent words obtained from the graph. Then, the Top-N highest-scored phrases are extracted as keyphrases.

- ExpandRank Algorithm (Wan and Xiao 2008): In ExpandRank Wan & Xiao propose using a small number of nearest neighbor documents to provide more knowledge in improving single document keyphrase extraction. A graph-based ranking algorithm is applied on the expanded document set to make use of both: (1) the local information in the specified document and (2) the global information in the neighbor documents. Each document is represented by a term vector where each vector dimension corresponds to a word type present in the document, and its weight is computed by tf–idf. Specifically, for a particular document, the approach first finds its k nearest neighbors from a text corpus adopting the widely-used

  cosine measure. Then, the graph for the document is built using the co-occurrence statistics of the candidate words collected from a larger document set of k+1 document, D = {d0, d1, d2, ..., dk}. In the graph, each vertex v_i corresponds to a candidate word type in D, and each edge connects two vertices v_i and v_j if the corresponding word types co-occur within a window of w units in the document set. The weight (5) of an edge $w(v_i, v_j)$ is computed as follows:

$$w(v_i, v_j) = \sum_{d_p \in D} sim_{doc}(d_0, d_p) \times count_{dp}(v_i, v_j) \quad (5)$$

  where $sim_{doc}(d_0, d_p)$ is the cosine similarity to reflect the confidence value for document d_p in the expanded document set and $count_{dp}(v_i, v_j)$ is the count of the co-occurence between words v_i and v_j in d_p. Once the graph is constructed, the rest of the procedure can be performed operations are similar to TextRank.

However, ExpandRank builds upon SingleRank by incorporating neighboring documents but without significant performance improvements.

- KP-Miner System (El-Beltagy and Rafea 2009): The keyphrase extraction in the KP-Miner system is a three-step process: candidate keyphrases selection, candidate keyphrases weight calculation and finally keyphrase refinement. KPMiner is a non-learning, ranking-based system, which operates on n-grams and uses a modified version of tf−idf, where the document frequency for n-grams with n greater than one is assumed to be one. KP-Miner also boosts the weights of multiword candidates in pro-portion to the ratio of the frequencies of single word candidates to all candidates.
  In a reranking step, the tf−idf of each term is recalculated based on the number of times it is subsumed by other candidates in the top 15 candidates list. KPMiner has the advantage of being configurable as the rules and heuristics adopted by the system are related to the general nature of documents (Arabic and English) and keyphrases.
  This requires the users of this system to understand the document(s) being input to the system and adjust it to their particular needs. Nevertheless, KP-Miner does not subtract the overlap-ping count during its weight stage calculation and only considers terms as candidates which occur on their own in the text (i.e., candidates surrounded by punctuation marks or stop words). A further improvement to the system would entail allowing certain stop words to appear within the produced keyphrases.

- Keycluster Technique (Liu et al. 2009): Liu et al. involve an unsupervised clustering technique (henceforth Keycluster). They assign an importance score to a word, and then pick the top ranked words as keywords by clustering semantically similar candidates using Wikipedia and co-occurrence-based statistics. The Wikipedia and co-occurrence-based statistics ensure that the extracted keyphrases cover the entire document. The main idea is that each cluster represents a unique aspect of the document and takes a representative word from each cluster so that the document is covered from all aspects. However, Keycluster gives each topic equal importance scores while extracting keyphrases from each topic cluster process. Later, Liu et al. (2010) propose Topical PageRank (TPR), an approach that overcomes the above-mentioned weakness of Key-Cluster. It runs TextRank multiple times for a document, once for each of its topics generated by a Latent Dirichlet Allocation LDA (Blei et al. 2003). Consequently, TPR ensures that the extracted keyphrases cover the main topics of the document. The final score of a candidate is computed as the sum of its scores for each of the topics, weighted by the probability that a given topic is in the document.

- CommunityCluster Method (Grineva et al. 2009): Grineva et al. present a novel method for key term extraction from text documents, where a document is modeled as a graph of semantic relationships between terms of that document. In this method, the terms related to the main topics of the document tend to bunch up into densely interconnected subgraphs or communities, while non-important terms fall into weakly interconnected communities, or even become isolated vertices.

This technique introduces a criterion function to select groups that contain key terms discarding groups with unimportant terms. Weighing terms and determining the semantic relatedness between them exploits the information extracted from Wikipedia. Using such an approach creates two advantages. First, it allows for the effective processing of multi-themed documents. Second, it reduces the noisy unstructured text, such as navigational bars or headers in web pages, online chats, SMS, emails, newsgroups, and blogs. Method evaluations show that Com-munityCluster outperforms Keycluster algorithm, Topical PageRank Algorithm and Yahoo! term extractor (in recall scores without losing precision).

- Sarkar et al. approach (Kamal Sarkar Mita Nasipuri 2010): In this novel Multilayer Perceptron (MLP) based approach, Sarkar et al. present a neural network-based approach to keyphrase extraction from scientific articles. This method consists of three primary components: (1) document preprocessing, (2) candidate phrase identification directed by a classifier based on a set of features (phrase frequency, phrase links to other phrases and Inverse Document Frequency, Phrase Position, Phrase Length and Word Length), (3) keyphrases are extracted using a neural network by training the MLP to predict whether a phrase is a keyphrase or not. Experiment results of the proposed system reveal that this system performs better than the publicly available keyphrase extraction system KEA. Later, Sarkar (2013) proposed a hybrid approach to extract keyphrases from medical documents, which combines two methods: the first one assigns weights to candidate keyphrases based on an effective combination of features such as term position, term frequency, *idf* , while the second one assigns weights to candidate keyphrases using some knowledge about their similarities to the structure, and characteristics of keyphrases available in the memory (stored list of

  keyphrases). The proposed approach results outperform some state-of-the-art keyphrase extraction systems like KEA and KP-Miner.

- TopicRank Algorithm (Bougouin et al. 2013): TopicRank is an enhanced variant of the TextRank algorithm (Mihalcea and Tarau 2004). It extracts noun phrases that rep-resent the main topics of a document. The noun phrases are clustered into topics and used as vertices in a complete graph. The resulting graph stands as a topical representation of the document. Topics are scored using the TextRank ranking model and keyphrases are then extracted by selecting the most representative candidate from each of the top-ranked topics. This approach offers several advantages over existing graph-based keyphrase extraction methods: (1) first, as redundant keyphrase candidates are clustered, extracted keyphrases cover the main topics of the document better, (2) the use of a complete graph captures the relations between topics without any manually defined parameters, and (3) it induces better or similar performance than the state-of-the-art connection method that uses a co-occurrence window (Bougouin et al. 2013). However, in keyphrase selection, the experiments show that the current method did not provide the best solution that could have been achieved with the ranked clusters.

- CiteTextRank Algorithm (Gollapalli and Caragea 2014): CiteTextRank extracts keyphrases from research articles. It is a graph-based algorithm that incorporates evidence from both (1) document content, and (2) contexts in which the document is referenced within a citation network. This evidence is used in a flexible manner to score keywords that are later used to score keyphrases.

Unlike other PageRank-based keyphrase extraction models, CiteTextRank builds a graph to compute scores (of nodes or vertexes) using information from multiple contexts (Mihalcea and Tarau 2004; Wan and Xiao 2008). Moreover, CiteTextRank effectively captures the notion of word importance across multiple global and citation contexts. CiteTextRank improves precision by 9-20% over state-of-the-art baseline algorithms such as TextRank, Keycluster, and Topical PageRank.

- Cho and Lee (2015) propose a new approach to select latent keyphrases (keyphrases that do not appear in the document). To do this, they extract candidate phrases by referencing neighbor documents, which are similar to the given document and evaluate the individual words of each candidate by considering the topic. They use Latent Dirichlet Allocation (Blei et al. 2003) to evaluate the importance of single words by considering the document's topics. The candidates are then measured by calculating the harmonic mean of their component's importance. By averaging, this method mitigates the overlapping problem. However, during averaging, this method does not consider the relationship between components. Consequently, performance deteriorates with more candidates.

- SGRank Algorithm (Danesh et al. 2015): Danesh et al. present a hybrid statistical-graphical algorithm that capitalizes on the heuristics of two major algorithm families of unsupervised keyphrase extraction: statistical and graph-based. The SGRank algorithm processes an input document in four stages. In the first stage, all possible N-grams (including the removal of the unlikely keyphrases) are extracted from the input text. In the second stage, the remaining n-grams are ranked based on a modified version tf–idf, in the third stage, the top-ranking candidates from the second stage are reranked based on additional statistical heuristics such as the position of the first occurrence and term length. A Position of First Occurrence factor (PFO) is defined according to the following formula:

$$PFO(t,d) = \log\left(\frac{cutoffPosition}{p(t,d)}\right) \qquad (6)$$

where $p(t,d)$ is the position of term t's first occurrence in a document d, a $cutoffPosition$ value is set to 3000. Experiment results on 2 datasets show that the average performance is considerably better than graph-based algorithms (KP-Miner and TextRank).

- Smatana & Butka Alogorithm (Smatana and Butka 2016): Samatana and Butka pro-pose a modification of single document keyphrase extraction models (without external information). This modification is based on hierarchical concepts. A Formal Concept Analysis (FCA) is used for the creation of a hierarchical concepts' method, which organizes objects into a concept lattice (structure of hierarchically organized clusters known as formal concepts). In this technique, after preprocessing the input documents, the objects extracted are sentences or paragraphs, and the attributes are frequencies of terms in particular objects. For this input data, a conceptual model is created using an FCA-based algorithm known as a generalized one-sided concept lattice. Hierarchical concepts from this model are used for the extraction of keyphrases from a document.

Compared with other standard keyphrase extraction weight metrics (tf, tf−idf, Chi-square, Information Gain), Smatana & Butka have shown improvement (in precision-recall) in keyphrase extraction, especially when unigrams, bi-grams and tri-grams are extracted in the preprocessing step from documents. However, the authors did not compare their method with some of the state-of-the-art keyphrase extraction systems.

- KeyphraseDS Algorithm (Yang et al. 2017): KeyphraseDS is a novel **document summarization** mechanism that can organize scientific articles into multi-aspect and informative scientific project summaries by exploiting keyphrases. KeyphraseDS consists of three steps: keyphrase graph construction, semantic aspect generation, and content selection. First, keyphrases are extracted through a CRF-based model exploiting various features, such as syntactic features, correlation features, etc. Spectral clustering is then performed on the keyphrase graph to generate different aspects, where the semantic relatedness between keyphrases is computed through knowledge-based similarity and topic-based similarity.

  Significant sentences are then selected with respect to the generated aspects through integer linear programming (ILP), which takes semantic relevance, semantic diversity, and keyphrase salience into consideration (Yang et al. 2017). Extensive experiments, including automatic evaluation and human evaluation, demonstrate the effectiveness and the feasibility of this method. Compared with other models (KEA, SVM), a CRF-based model yields the best performances in both precision and recall rates in keyphrase extraction.

2.2.3.3 Deep Learning Approaches

The deep learning approaches: Deep learning (DL) (also known as hierarchical learning or deep structured learning) is part of the machine learning methods family based on learning data representations (see Figure. 2.7). DL allows computational models that are composed of multiple processing layers to learn these representations of data with multiple levels of abstraction. The DL architectures models include: Deep Neural Networks (DNNs), and Deep Belief Networks (DBNs), Recurrent Neural Networks (RNNs), Deep Recurrent Neural Networks (DRNNs). DL (LeCun et al. 2015) techniques present promising approaches for various NLP tasks. As a result, DL models have recently gained popularity and interest in AKPE systems (Zhang et al. 2016; Meng et al. 2017). Recurrent neural networks (RNNs) (Elman 1990) have been applied to several sequential prediction tasks. RNNs and bidirectional RNNs are considered the most suitable model in NLP tasks. They deal with a keyphrase as a sequence of tokens and incorporate information from previous ones.

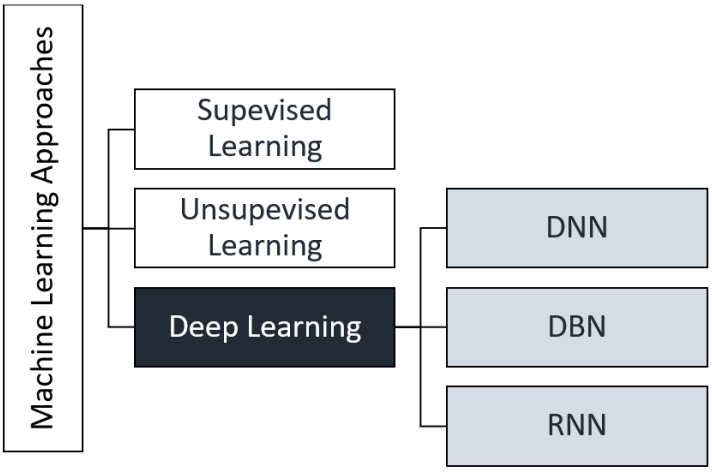


Figure. 2.7 Deep learning approaches

They are beneficial, especially when making a decision on the current token. The information provided by the subsequent tokens is generally valuable. Below, we examine some representative AKPE systems that adopt the DL approach.

- Meng et al. (2017) apply an RNN-based generative model (deep keyphrase generation) to predict keyphrases. They incorporated a copying mechanism in RNN, which enables the model to successfully predict "absent keyphrases" that rarely occur (phrases that do not match any contiguous subsequence of source text). The authors built a dataset comprised of 20,000 scientific documents to train and evaluate the system.

- Zhang et al. (2016) proposed a novel RNN model to tackle the problem of extracting relevant keyphrases from tweets, where the length limitations of Twitter-like sites make the performance of existing AKPE systems decrease. To solve this problem, the RNN model combines keywords and context information. In addition, a massive dataset of tweets with gold standard keyphrases was constructed to evaluate the proposed approach.

2.2.4 Evaluation Methods

The goal of this final step is to evaluate the extracted keyphrases. The familiar metrics for evaluating keyphrase extraction systems are:

2.2.4.1 Manual Evaluation

The Manual Evaluation (ME), is based on human judgments. Various authors have used human judges to evaluate the results of AKPE techniques (Turney 2000; Matsuo and Ishizuka 2004), as well as to determine if the retrieved keyphrases are representative of a document's content. However, human evaluation of the extracted keyphrases is very expensive and time-consuming, and it is not appropriate for any kind of parameter tuning.

2.2.4.2 Automatic Evaluation

The Automatic Evaluation (AE), relies on (1) matching and (2) scoring. (1) Matching a set of annotated gold standard keyphrases with a ranked list of extracted keyphrases (output of a keyphrase extraction system). This evaluation approach is also adopted by the SemEval-2010 shared task on keyphrase extraction (Kim et al. 2010). This exact matching can be true or false, depending on whether gold standard keyphrases and the extracted ranked keyphrases are equivalent according to the matching strategy. After matching, (2) scoring the output using evaluation metrics is performed. Among these metrics, there are the well-known: (1) precision (P), recall (R), and F-score (F), (2) keyphrase ranking (as an evaluation metric) to distinguish between systems that score the same number of exact matches (Liu et al. 2010), or to determine which systems rank the most correct keyphrases above incorrect candidate keyphrases (Zesch and Gurevych 2009), (3) the information retrieval R-precision (R-p) (Zesch and Gurevych 2009), which is defined as the precision when the number of retrieved documents equals the number of relevant documents in the document collection. An R-precision of 1.0 is equivalent to per-fect keyphrase ranking and perfect recall. This means that a system that achieves a perfect (R-p) value ranks all the keyphrases on top of the non-keyphrases (Zesch and Gurevych 2009). However, the exact matching of keyphrases is an excessively strict condition, which is problematic since it overlooks near matches that are basically semantically identical (such as different grammatical forms, and synonyms). To overcome this issue, in some cases, near matches have also been considered. Some researchers have developed a number of systems for automatic evaluation via near or partial matching. Mihalcea and Tarau (2004) and Jarmasz and Barriere (2004) have suggested treating semantically-similar keyphrases as correct based on similarities calculated over a large corpus. Similar to Mihalcea and Tarau (2004) and Jarmasz and Barriere (2004), Medelyan and Witten (2006) used semantic relations defined in a thesaurus to treat near match keyphrases. Later, Zesch and Gurevych (2009) used an n-gram based approach relative to the gold standard to compute near matches. Additionally, they proposed some metrics that consider semantic similarity and character n-grams in order to differentiate between probable near matches and completely erroneous keyphrases (Zesch and Gurevych 2009; Kim et al. 2010). Nevertheless, these systems only give a limited solution as they can only detect a subset of near-misses to exact match terms (Kim et al. 2010).

2.2.5 Comparative Study of Existing Approaches

Reviewing the 34 noteworthy keyphrase extraction techniques revealed interesting trends in the various stages of keyphrase extraction. The performance of these techniques is presented in Table 2.4. The evaluated datasets are presented in Table 2.3. Table 2.4 presents a characterization summary of the aforementioned AKPE along with the used dataset, target task, ranking approach, features, and best evaluation scores achieved by each technique as it is described in their papers. In Table 2.5, we compare the most prominent and representative of these afore-mentioned AKPE techniques.

The evaluation was carried out on the well-known shared dataset "SemEval-2010"[6], which is composed of 284 technical and scientific documents (40 trial documents, 144 training documents, and 100 test documents) (Kim et al. 2010). Additionally, in this dataset, a set of gold-standard keyphrases is available for the evaluation. For each article, three sets of gold-standard keyphrases are provided: author-assigned keyphrases, reader-assigned keyphrases, and a combination of the two previous sets. In the two tables, P, R, and F denote precision, recall, and F-score respectively.

2.2.5.1 Results and Discussion

The top-performing systems return F-scores in the upper forties (see Table 2.5). Apparently, this number is low, which indicates that much more improvement can be made. How-ever, since keyphrase extraction is a subjective task, an F-score of 100% is not possible. In Table 2.4, the F-score of the best-performing systems on their testing data is achieved by: KeyphraseDS (Yang et al. 2017), Keycluster Algorithm (Liu et al. 2009), Community Cluster Algorithm (Grineva et al. 2009) and the Maui Algorithm (Medelyan et al. 2009). In Table 2.5, Kp-Miner system (El-Beltagy and Rafea 2009), HUMB system (Lopez and Romary 2010), DPM-Index system (Haddoud 2014) showed the best performance on the shared 'SemEval-2010/Task-5' dataset (Kim et al. 2010). Given these results and the analysis of several aforementioned AKPE techniques, various challenges are revealed in different stages of keyphrase extraction: candidate keyphrases identification, feature engineering, candidates ranking and evaluation. First, most of these systems apply pre-processing before candidate identification.
In **the candidate identification stage**, numerous systems used either POS-based or n-grams, noun phrase chunks, or a combination of the two of them. However, in this step, there is a difficulty in dealing with the structure of a keyphrase which is sometimes irregular and comprised of only a single word or multi-word noun phrase or multiple multiword noun phrases connected by prepositions.

Additionally, most systems face different challenging tasks whereby they: (1) fail at predicting candidate keyphrases with different sequential orders, or (2) encounter keyphrases that contain synonyms or abbreviations.

In **feature engineering stage**, some authors applied a variety of features: statistical, structural and syntactic/morphological. Other authors added external resource-based features, such as Wikipedia and external terminological corpora databases. Nevertheless, the use of external knowledge bases is more computationally expensive than baseline features. Information such as document structure or document statistics appears to be effective for automatic keyphrase extraction across the results (see Tables 2.4 and 2.5). However, these features only detect the relevance of each word in a document based on the statistics of word occurrence and co-occurrence, and some of them are unable to benefit from the full semantics that underlies the document content.

[6] http://semeval2.fbk.eu/semeval2.php?location=data

Table 2.3 Some evaluated datasets in this survey

Name or source	*Number of files*	*File's type*	*Dataset domain*
NZDL (New Zealand Digital Library)	1800	Technical reports	Multi domains
Inspec	500 or 2000	Paper abstracts	Scientific
uw-can-data	314	Web pages	Multi domains
DUC 2001	308	News Articles	News
citeulike.org collection	–	Scientific documents	Scientific
CSTR, NASA, FIPS, Journals, Ali web	502	Documents	Multi domains
ACM Digital Library, Cogprints, Springerlink, Wikipedia	570	Scientific papers	Scientific
Geeking with Greg, DBMS2, Stanford Infoblog	252	Blogs	Technical
Elsevier, Springer	150	Full journal articles	Scientific
CiteSeerx	790	Papers published in WWW and KDD conferences	Scientific
ACL Anthology Network (AAN) corpus	+ 20000	Research Papers	Scientific
Reuters collection	21578	NewsWire articles	News
SemEval-2010 Task-5	284	Scientific articles	Technical and Scientific

To rank candidates, supervised systems have used different machine learners such as maximum entropy, logistic regression, the learn-to-rank classifier based on SVMrank, Naïve Bayes and C4.5 bagged decision trees. However, supervised approaches depend on the predetermination of features before running the system. They use off-line analysis and they need a considerable amount of computation time for training. Consequently, these features cannot be modified or learned during the system process. It is difficult to enumerate all the features associated with a specific domain because the linguistic, statistical and external knowledge of the text should be exploited. Additionally, some keyphrases are positively recognized as candidate keyphrases but fail to be ranked at the top for extraction. In unsupervised systems, as shown in Tables 2.4 and 2.5, the F-score of the best-performing systems was for those that used topic clustering methods or simple probabilistic ones. For systems that used graph-based methods, there is a limitation in covering the various relevant topics of a document. Nevertheless, they work well for single documents.

Table 2.4 Characteristic summary of some aforementioned AKPE techniques along with the used features, dataset, target domain, approach and performance rates

Technique/ Contributor	Target/ Domain	Dataset	Approach	Features		Top N/ P R F			
				Internal	*External*	N	P	R	F
KEA System (Witten et al. 1999)	Keyphrase Extraction	NZDL	Supervised (Naïve Bayes model)	*First occurrence position*, $tf-idf$	–	5 10 15	0.300 – 0.165	– – –	– – –
GenEx Algorithm (Turney 2000)	Keyphrase Extraction	75 Journal Articles, 311 Email Messages, 266 Web Pages	Supervised (C4.5 Decision Tree)	*Term Frequency* (tf), *first occurrence position, phrase Length*	–	5 10 15	0.239 – 0.128	– – –	0.118 – –
TextRank Algorithm (Mihalcea and Tarau 2004)	Keywords and Sentence Extraction	Inspec	Unsupervised (graph-based)	*POS, distance between word occurrences*	–	5 10 15	– 0.142 –	– 0.125 –	– 0.127 –
CorePhrase Algorithm (Mihalcea and Tarau 2004)	Topic Discovery using Keyphrase Extraction	uw-can-data	Unsupervised (graph-based)	*DF-average weight, Phrase frequency, Average phrase depth*	–	– – –	– 0.41 –	– – –	– – –
Single Rank Algorithm (Wan and Xiao 2008)	Single Document Keyphrase Extraction	DUC 2001	Unsupervised (graph-based)	*POS tagging, neighborhood, Level saliency score*	–	5 10 15	0.34 0.35 0.26	0.25 0.28 0.31	0.26 0.31 0.30
Maui Algorithm (Medelyan et al. 2009)	Tags Extraction	citeulike.org collection	Supervised (Naïve Bayes and bagged decision trees models)	$tf-idf$, *position of first occurrence, Keyphraseness, Phrase length, spread, node degree*	*Wikipedia based keyphraseness, spread, semantic relatedness, Inverse Wikipedia linkage*	– – –	– 0.457 –	– 0.486 –	– 0.471 –

Table 2.4 (continued)

Technique/ Contributor	Target/ Domain	Dataset	Approach	Features		Top N/ P R F			
				Internal	*External*	N	P	R	F
KP-Miner System (El-Beltagy and Rafea 2009)	keyphrase Extraction (Agriculture domain)	CSTR, NASA, FIPS, Journals, Aliweb	Unsupervised (Weighting model)	$tf - idf$, *first occurrence position, Boosting Factor for weight bias*	–	7 15 20	0.214 0.143 0.214	0.277 0.358 0.277	0.241 0.205 0.241
Core Word Expansion Algorithm (You et al. 2009)	keyphrase Extraction	ACM digital library, Cogprints, Springer-link, Wikipedia	Supervised; Unsupervised	*Inverse document frequency* (idf), *N-gram*	–	5 15 20	0.272 0.192 0.169	0.189 0.267 0.353	0.223 0.223 0.228
Community Cluster Algorithm (Grineva et al. 2009)	Keyphrase Extraction	Geeking with Greg, DBMS2, Stanford Infoblog	Unsupervised (semantic graph-based)	*N-gram*	*Wikipedia-based keyphraseness*	– – –	– 0.351 –	– 0.615 –	– 0.447 –
Keycluster Algorithm (Liu et al. 2009)	Keyphrase Extraction	Inspec	Unsupervised (topic clustering model)	*Co-occurrence-based term relatedness* (idf), *N-gram*	*Wikipedia-based term relatedness*	– Using spectral clustering –	– 0.350 –	– 0.660 –	– 0.457 –
MLP based Algorithm (Kamal Sarkar Mita Nasipuri 2010)	Keyphrase Extraction	Elsevier, Springer	Supervised (MLP model)	*Noun Phrase, Phrase frequency, phrase links to other phrases,* idf, *phrase position, phrase length and word length*	–	5 10 15	0.34 0.22 0.17	0.35 0.46 0.51	0.344 0.297 0.255

Table 2.4 (continued)

Technique/ Contributor	Target/ Domain	Dataset	Approach	Features		Top N/ P R F			
				Internal	*External*	N	P	R	F
CeKE Algorithm (Bulgarov and Caragea 2015)	Keyphrase Extraction	CiteSeerx	Supervised (Naïve Bayes model)	$tf - idf$; *relativePos, POS; inCited, inCiting; citation tf-idf, first position, tf-idf-Over, firstPosUnder, keyphraseness*	–	WWW KDD –	0.25 0.254 –	0.46 0.440 –	0.32 0.321 –
KeyphraseDS System (Yang et al. 2017)	Document Summarization	ACL Anthology Network (AAN) corpus	Supervised (CRF-based model, ILP-based optimization, LDA model)	*Syntactic features(POS and Relation), correlation features (MutInfoPre, MutInfoNext), semantic relatedness*	*InWiki*	– – –	– 0.38 –	– 0.62 –	– 0.47 –
KeyEx Method (Xie et al. 2017)	Keyphrase Extraction	Reuters 21578 collection, SemEval-2010	Supervised ()	$tf-idf$; *POS; first occurrence position, statistical pattern features (pattern support, pattern length,number of closed patterns,* $PFIDF$ *value)*	–	– – –	– 0.134 –	– 0.35 –	– 0.19 –

Table 2.5 Characteristic summary of the most representative AKPE techniques and their performance on the shared "SemEval-2010 dataset"

Technique/ Contributor	Approach	Features		Top N/ P R F			
		Internal	*External*	N	P	R	F
KP-Miner (El-Beltagy and Rafea 2009)	Unsupervised	$tf-idf$, *first ccurrence position, boosting factor for weight Bias*	–	5 10 15	0.296 0.233 0.253	0.123 0.205 0.261	0.253 0.261 0.258
SZTERGAK (Berend and Farkas 2010)	Supervised	$tf-idf$, *first occurrence, phrase length, POS, suffix, acronymity, PMI, syntactic,* $sf-isf$, *keyphraseness*	*Wikipedia*	5 10 15	0.342 0.285 0.248	0.117 0.194 0.254	0.174 0.231 0.251
HUMB (Lopez and Romary 2010)	Supervised	*Position of a term, position of the first occurrence, phraseness, informativeness, keywordness, length of the term*	*Wikipeia keyphrase-ness, GRISP based term, GRO-BID/TEI*	5 10 15	0.300 0.320 0.272	0.133 0.218 0.278	0.198 0.260 0.275
DPM-index (Haddoud 2014)	Supervised	*Length n of the n-gram t in words;* tf; idf; $tf-idf$, *first position, first sentence, head frequency, average sentence length, substrings frequencies sum, generalized dice coefficient, maximum likelihood estimate; kullback–leibler divergence; document phrase maximality index(DPM-Index); DPM-index cross tf-idf, tf-idf ratio of the term and its main compound, k-Means of the normalized positions (k = 1, 2)*	–	5 10 15	0.448 0.362 0.283	0.153 0.247 0.289	0.228 0.294 0.286
KeyEx (Xie et al. 2017)	Supervised	$tf-idf$; *POS; first occurrence position, statistical pattern features (pattern support, pattern length,number of closed patterns,* $PFIDF$ *value)*	–	5 10 15	0.720 0.600 0.510	0.150 0.280 0.360	0.248 0.381 0.422

However, unsupervised keyphrase extraction techniques sometimes produce less relevant keyphrases than supervised ones, and they are less specific. In *deep learning systems*, most of them used the RNN model. It was used for the joint processing of keyword ranking, keyphrase generation, and keyphrase ranking. Especially, this model enhanced the generation of embedded keyphrases. Despite the fact that DL reduces the need for engineering features, they are computationally expensive in training. However, DL approaches require a large amount of data to perform better. Additionally, there are just a few AKPE systems using DL.

In the evaluation stage, traditionally, most AKPE systems have been evaluated using the proportion of Top-N candidates that exactly match the gold standard keyphrases (Frank et al. 1999; Witten et al. 1999; Turney 2000). In most papers, the authors commonly manually assign keyphrases based on their semantic meaning, instead of following the written content in the document. Consequently, current evaluation methods do not cover keyphrases that are semantically relevant. In this case, the exact matching of keyphrases can be very strict and problematic because it ignores near matches that are largely semantically identical. Furthermore, it is difficult to measure the relative superiority of different machine learning approaches over the task, as they were combined with different candidate selection techniques and feature sets. There is no doubt, however, that there is certainly room for improvement on the APKE task. Generally, some limits and drawbacks of existing AKPE systems can be summarized as follows:

(1) The redundancy is induced by semantically equivalent keyphrases,
(2) The over-generation: frequency of keyphrases alone cannot be a criterion for extracting keyphrases,
(3) The infrequency: many systems overlook the infrequent relevant keyphrases, and focus on frequent words in the associated documents,
(4) As the length or the number of documents increases, the global performance of all ranking algorithms can drop,
(5) The evaluation is insufficient since many studied systems have evaluated their approaches on only one dataset and
(6) The evaluation of exact matches is deemed insufficient to cover all relevant keyphrases.

2.2.5.2 Recommendations

The aforementioned discussion reveals significant trends across different stages of AKPE. The objective of this study is to (i) enhance the understanding of the AKPE process and differences between the various approaches and techniques used, (ii) provide the guidelines and necessary components to be considered in future AKPE solutions. The analysis regarding every stage, approach, and technique have already been given in the above sections. Thus, for future AKPE works (depending on the final systems' objectives), and to overcome the mentioned limitations, we suggest some recommendations for each stage of the AKPE process.

- Enhancing **the preprocessing stage and candidate selection stages** will ameliorate the selection of good (the morphologic form) candidate phrases that can improve the time, quality and complex efficiency of a keyphrase extraction system. Apparently, using a combination

between POS tag sequence (noun phrase based) and n-grams showed worthwhile results in candidates' identification.

- **In the feature engineering stage**, the success of AKPE systems in this stage depends primarily on the quality, variety, and quantity (structural, linguistic, statistic, semantic) of the used features. Apparently, tf–idf, co-occurrence, length of the candidate phrase, the relative position of the phrase, the keyphraseness, have revealed worth-while results in different systems (Lopez and Romary 2010; Haddoud 2014).

  Involving semantic information is also very important in reducing redundancy problems, and exploiting semantic relationships between words may potentially help in extracting more relevant and informative keyphrases. While the use of external knowledge bases is computationally expensive, using them has many benefits: (1) they will make the keyphrases more grammatically understandable, (2) they can be used to boost the infrequent relevant keyphrases which may solve the infrequency problem and (3) they can increase the precision, recall and exact match rates.

- **In the candidates ranking stage**, supervised approaches are more precise in extracting the exact match keyphrases, especially when the concerned classes are determined. They can improve the quality of keyphrases with repeated training. It is notable that the use of bagged decision trees C4.5 with a variety of good features performs better than other supervised models (Haddoud 2014). Otherwise, using unsupervised approaches has the benefit of being more widely applicable, since their models do not require any pre-knowledge of the domain or trained data. These models have some practical advantages such as (1) notable time saving, (2) computing resource saving and (3) the ability to produce a large high-quality keyphrase list, which makes the system more effective especially in real-time and query refinement applications. In the case of small and medium-sized documents (≈ 100 − 1000 tokens), using co-occurrence graph-based ranking is highly recommended, and it outperforms supervised systems by a wide mar-gin (Zesch and Gurevych 2009). Awareness of the domain or topic is necessary in order to obtain robust systems. Topic-based clustering models have notable potential (see Table 2.4). They use meaningful concepts or contents of documents, and they undertake an important role in predicting and extracting latent keyphrases (keyphrases that do not appear in documents) by using methods such as LDA. Similarly, topic models can aid to solve the problem of infrequency. Indeed, future systems should take into account keyphrases based on an understanding of the text, regardless of the presence or absence of keyphrases in the document text. Recently *deep learning approaches* have attracted the attention of the research community. Indeed, APKE can gain significant benefits from using DL. DL models aid in combining keywords and context information to per-form the keyphrase extraction task. We recommend the use of DL models, especially for extra-large document sets. There is no doubt that we will continue to see a growth in the application of DL methods on the APKE.

Table 2.6 illustrates some of the strengths and weaknesses of supervised, unsupervised and DL approaches. Depending on the application at hand, this table can give an idea of the approaches that can be used before building an AKPE system.

- **In the evaluation stage**, we believe that the context and topics of keyphrases should be considered for two reasons: (1) better performance in extracting keyphrases and sufficient coverage of each document. To better improve near matches, we suggest treating semantically similar keyphrases as correct based on similarities computed over a large corpus (Mihalcea and Tarau 2004; Jarmasz and Barriere 2004), or using semantic relations defined in a thesaurus (Medelyan and Witten 2006). Indeed, more semantically-motivated evaluation received less attention and should be engaged to give more accurate keyphrase suitability. A suitable keyphrase should satisfy the following properties: (1) clear and understandable to readers, (2) semantically relevant to the document topic, (3) sufficient coverage (cover all the document entities). Additionally, it is necessary to evaluate keyphrase extraction systems on multiple and variant datasets to gain a better view of performance. Certainly, evaluating the same system can vary depending on the type, volume, and content of the dataset.

Table 2.6 The strengths and weaknesses of supervised, unsupervised and DL approaches for the AKPE task

Approaches	Strengths	Weaknesses
Supervised	(+) More precision and very specific (in the definition of the labels).	(-) Complex models (need to understand very well and label the inputs).
	(+) The input data is well known and is labeled.	(-) Needs to relearn and establish the model every time the domain changes (impractical to label training set from time to time by users).
	(+) Better decision boundary.	
	(+) Parameterized heuristic rules.	(-) Not suitable for real-time tasks (off-line analysis).
	(+) Ability to determinate the number of classification classes.	(-) Need gold standard keyphrases and training data to compare and evaluate.
	(+) More accurate results.	(-) Often dependent on the domain.
		(-) Not suitable for dynamic and growing data.
		(-) Time-consuming in training (computation time) and extraction phases.
		(-) Uses absolute feature values.
Unsupervised	(+) Less complexity (no labeling of data inputs is needed) and more flexible.	(-) Not specific in data sorting and output definitions.
	(+) Suitable for small and medium-sized documents.	(-) No coverage guarantees for all main topics.
	(+) Takes place in real time.	(-) Low accuracy of the results.
	(+) Does not exploit any manually tagged corpus or training data.	(-) input data is not known and not labeled.
	(+) Time efficient.	
	(+) Good solution for real-time applications.	
	(+) Simplicity-syntactic representation requires almost no language-specific, advanced knowledge linguistics or processing.	
	(+) Takes into account term co-occurrence patterns.	
	(+) Domain and language independent.	
	(+) More availability of unlabeled data.	
Deep Learning	(+) Performs very well on text data.	(-) Not suitable as general-purpose algorithms.
	(+) Reduces the need for feature engineering.	(-) Requires a very large amount of data.
	(+) Adaptable architecture.	(-) Computationally intensive to train.
	(+) Can be easily updated with new data (best-in-class performance on problems).	(-) Requires much more expertise to tune.
	(+) Can learn extremely complex patterns.	(-) Needs high-performance hardware.
	(+) Learns from high-dimensional data.	(-) Needs high-performance hardware.
		(-) Does not have much in the way of strong theoretical foundation (the learning process is considered a Black-Box).
		(-) What is learned is not easy to comprehend.

2.3. Contextual Information Retrieval

2.3.1. Introduction

With the continuous growth of the size and the availability of electronic data and the expansion of internet technology, users can laboriously find and satisfy their information need in some situations, which may pose a permanent and serious challenge to current search technologies in the future. This challenge leads to an increasing research interest in contextualizing the process of information search. This operation improves the limitations of many IR systems that have been mostly context-free. The context consists of many cognitive and social factors related to a user's intentions, tasks, and needs. Within the interaction between users and these systems, that divulgation of user context models is revealed. The concept of context has been long known as being very important in a variety of fields **(**McCarthy (1993), Edmonds (1999)), such as IR (Lindsay and Rajaram, 2017), Finkelstein et al. (2002), Haveliwala (2002), Lawrence (2000), (Sarwar et al. 2018), (Lindsay and Rajaram, 2017), (Tseng and Braginsky, 2018), (Jones et al. 2018), (Flake et al. 2017)**)**, automatic image analysis ((Dasiopoulou et al. 2005), (Mylonas and Avrithis, 2005)), mobile and pervasive computing ((Brown et al. 1997), (Coutaz et al. 2005), (Heer et al. 2003), (Hong and Landay, 2001)), computational linguistics (Eijck (2000), Finkelstein et al. (2002), (Wiebe et al. 1996)). Since context covers a vast landscape of research fields, we will only focus on a part of this landscape in this part of chapter, considering only related works of context with information retrieval (IR).

2.3.2. The Notion of Context In IR

Earlier, Dervin Allen (1997) considered context as an "unruly beast" because of its difficulties in defining methodological control. Vakkari Vakkari (1997) describes context as a 'situation' in his first International Conference on Information Seeking in Context (ISIC). In these past decades, the concept of context in information retrieval has been studied and discussed through numerous initiatives such as the IRIX workshops (Tseng and Braginsky, 2018); the monograph by (Ingwersen and Järvelin, 2017) Ingwersen and Järvelin (2005), the HARD and ciQA tracks at TREC (Sarwar et al. 2018) and the IIiX conference series (Jones et al. 2018). Recently, many IR systems and search engines have integrated the concept of context to improve the retrieval performance ((Tseng and Braginsky, 2018), (Sarwar et al. 2018), (Jones et al. 2018), (Lindsay and Rajaram, 2017), (Flake et al. 2017), O' donnell (2017), Bharat (2000), (Shen et al. 2005)).

Using context for retrieving valuable information is not a new idea. Several existing IR systems have utilized the concept for enhancing their IR systems performance ((Flake et al. 2017), O' donnell (2017), Bharat (2000), (Shen et al. 2005), (Tseng and Braginsky, 2018), (Sarwar et al. 2018), (Jones et al. 2018), (Lindsay and Rajaram, 2017)).

In Edmonds (1999), Edmonds defines context as an abstraction of an aspect of a heuristic for the learning and application of knowledge: *"the abstraction of those elements of the circumstances in which a model is learned, that are not used explicitly in the production of inference or prediction when the model is later applied, that allow the recognition of new circumstances where the model can be usefully applied.*".

Dey & Abowd (Dey and Abowd, 2000) define context as follows: *"any information that can be used to characterize the situation of entities (e.g., a person, a place or an object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves"***.** Another definition of context is given by Sato Sato (2001) as "a pattern of behavior or relations among variables that are outside of the subjects of design manipulation and potentially affect user behavior and system performance". According to Dourish Dourish (2004), *the context* may be defined in two perspectives: (1) the first one describes it as a representational problem where it is viewed as a form of delineable, independent, and stable of the activity. It consists of implicit attributes that describe the user and the environment in which information activities occur; (2) The second perspective defined it as an interactional problem and presented context as arising from the activity. It cannot be separated. In (Crestani and Ruthven, 2007), Crestani and Ruthven define context, particularly in the information retrieval field, as follows: *'context affects all aspects of information retrieval. A searcher's context affects how they interact with a retrieval system, what type of response they expect from a system, and how they make decisions about the information objects they retrieve'.* Later, Ruthven Ruthven (2011) provides another definition: context is *"a complex set of variables describing our intentions, personal characteristics, the data and systems available for searching, and our physical, social and organizational environments".* Ingwersen & Jarvelin (Ingwersen and Järvelin, 2017) say that *"There are social, organizational, cultural as well as systemic contexts, which evolve over time"***.** Thus, since context depends on previous knowledge, we can assume that *"the context is not explicit, not trivial and not static, but it is a dynamic concept composed of many factors like users, situations, devices, locations, environments, time and more". T*hese factors will influence how we use an information retrieval system and how we evaluate its performance.

2.3.3. Contextual Information Retrieval Systems

Most classical information retrieval systems "IRS" use statistical information to retrieve relevant information. However, several systems' design does not involve the concept of user context (see Figure. 2.8). The information is authored and exploited by millions of different user profiles, each with different knowledge, backgrounds, and preferences leading to different search contexts. For example, suppose a child search for an information. In that case, the IRS should provide an adapted answer to the child's age and his understanding level. Thus, it is necessary for IRS to face such challenges associated with the differences that coexist between users, their tasks and goals. There too many factors like history of interaction, the task in hand, time, place, surrounding environment that we must take them into consideration in the search process. These factors can be graded as dimensions, taxonomies, or categories of context. In the following paragraph, we give an overview of some prominent previous related works.

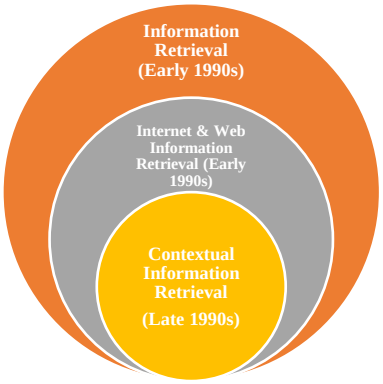


Figure. 2.8 Evolution of information retrieval systems

2.3.3.1 Context Taxonomies

In 2000, Dey & Abowd (Dey and Abowd 2000) classify the context information based on the entities in which the context is evaluated and the categories of context. Three entities are defined: (1) people (including individual or groups, co-located or distributed), (2) things (e.g., physical objects or software), (3) places (geographical space).

Latern, in 2002, Goker & Myrhaug (Goker and Myrhaug, 2002) classified a context in five main categories: (1) the environmental context, that captures the entities that surround the user (things, services, temperature, light, humidity, noise, and persons), (2) the personal context, consists of physical and mental information about the user, (3) the task context, is about what the user is doing, tasks, activities, and goals, (4) the social context, which describes the social aspects of the current user context, such as information about friends, neutrals, enemies, neighbors, co-workers, and (5) the Spatio-temporal context that describes aspects of the user context related to the time and spatial extent for the user context. In 2003, Kofod-Petersen and Aamodt (Kofod-Petersen and Aamodt, 2003) proposed an open context model as a part of the “Ambisense project” concerned by information retrieval via mobile devices. The taxonomic structure of context includes five main categories (see Figure. 2.9):

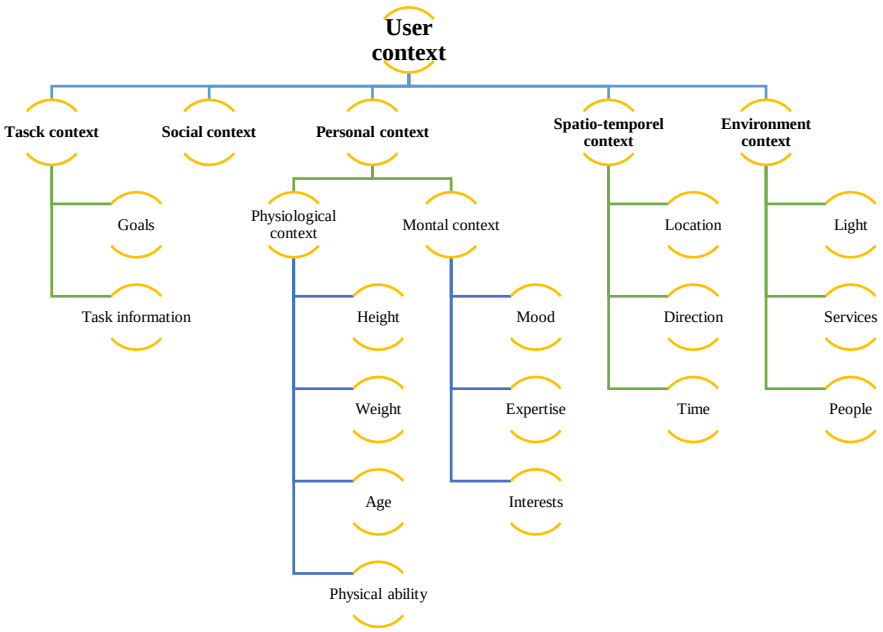


Figure. 2.9 The Ambisense model of context (Kofod-Petersen and Aamodt, 2003)

i) *Task context*: The task context describes what the user is doing, it can describe the user's goals, tasks, activities, etc. *ii) Social context:* describes the social aspects of the user, such as information about friends and relatives, the role the user plays, etc. *iii) Personal context*: this part describes the mental and physical information about the user, such as mood, expertise, disabilities, and weight. iv) *Spatio-temporal context:* this type of context is concerned with attributes like: time, location, and movement. *v) Environmental context*: This part captures the users surrounding, such as things, services, light, people, and information accessed by the user (Kofod-Petersen and Aamodt, 2003).

In 2006, a nested model of context for IR with six categories or "dimensions" (see Figure. 2.10) was presented by Ingwersen & Jarvelin Ingwersen (2006). According to Ingwersen (2006), the (1) *Intra-object structures* refer to context obtained from terms, phrases, image features, pixels, sentences, paragraphs, sections; (2) *Inter-object contexts* are concerned with the properties of documents, like links, citations, document clusters defined of various features; (3) *Interaction/session context* which is expressed by search/authoring process (*evidence*, e.g., eye/mouse movements, Work Task descriptions, explicit Relevance Feedback, search task path); features of algorithmic IR processes, auto-indexing keys-strings; (4) *Individual* context, dependent on central components (subjective): actor, current work, task perception … ;(objective): engine logic/algorithms, interface functionality; (5) *Collective* context, dependent on central components: (local) sociocultural and organizational structures/conditions: like domain vocabulary, natural work tasks, organizational preferences, several searchers' Work Task perceptions, socio-cognitive relevance assessments and recommendations, implicit Relevance Feedback behavior; (6) *Infrastructural contexts :*infrastructures influencing (not necessarily always in a remote way) all actors, components and interactive sessions, network type, speed, censorship, economic constraints …; and the (7) *historic context* operates across this

stratification, i.e., the history of *all* participating actors' experiences, forming their expectations. All information retrieval processes and activities are under the influence of this temporal form of context (Ingwersen (2006)).

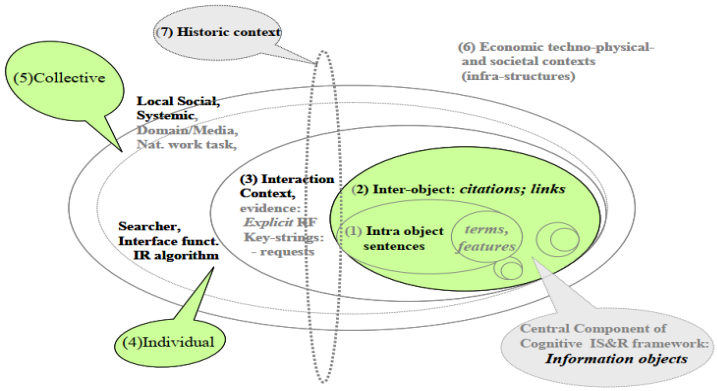


Figure. 2.10 Nested model of context types centered on Information Objects of an IR system Ingwersen (2006)

Later in 2008, Mansourian (Mansourian (2008)) identified five categories as the main contextual elements that affect end-users' search performance. These categories include:

1. *Web users' characteristics* that include three subcategories which are feelings, thoughts, and actions of the user during the search procedure and after having search results, which might be satisfactory or unsatisfactory for him/her;
2. *Type of the search tool*;
3. *Search topic* which can be divided into work-related and everyday life searches;
4. Search situation including four subcategories which are a place of search, type of search, immediacy, and importance of search; and
5. *The retrieved information resources* refer to three subcategories: level of provision, searchability, and presentation format.

Tamine et al. (Tamine et al. 2009) proposed five context-specific dimensions, listed below (see Figure. 2.11):

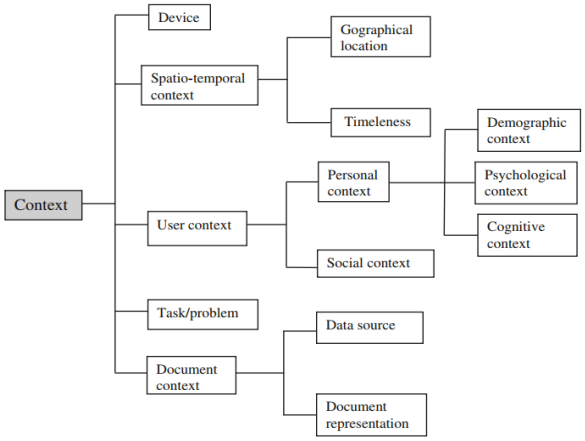


Figure. 2.11 The multi-dimensional concept of context in an IR system (Tamine et al. 2009)

In Figure. 2.11, Tamine et al. (Tamine et al. 2009) introduce the context of the information access device and the document context, the Spatio-temporal context, the user context, and the task context. This taxonomy (dimensions) covers all the past discussed categories in a more organized and detailed way. In 2012, Melucci (2012) considered four categories of context in his computational framework: (1) content refer to the informative content and relationships between queries and documents, (2) geographical: any variable that is having some relationship with space location, (3) interaction that is observed overtime during the interaction between users and IR systems, and (4) social variables refer to user communities or groups. In 2017, O' donnell O' donnell (2017), in his advanced context data search, considers many context categories like user or community characteristics context, content context, user cognitive context, physical or virtual location context, time, virtual object or idea, device, philosophy, goal. More recently, in 2018, Tseng & Braginsky (Tseng and Braginsky, 2018) invent a system that provides context-relevant search for a user based on his: interest, location, content, associated timing.

Generally, several viewed literatures share the same context taxonomies. However, as mentioned above, the taxonomy of Tamine et al. (Tamine et al. 2009) covers almost all the discovered and studied dimensions in the Contextual IR Systems (CIRS).

2.3.3.2 Context in IR Systems

The need of precise and better results in IRS and search engines has led to the development of sophisticate contextual IRS (context-aware IR). In IRS, the context depends mainly on some of the related categories (listed above). Figure. 2.12 presents the basic architecture of a context-aware, also called contextual IRS given by (Tamine et al. 2009). We notice that this architecture is based on two main axes: (1) context modeling and (2) contextual retrieval.

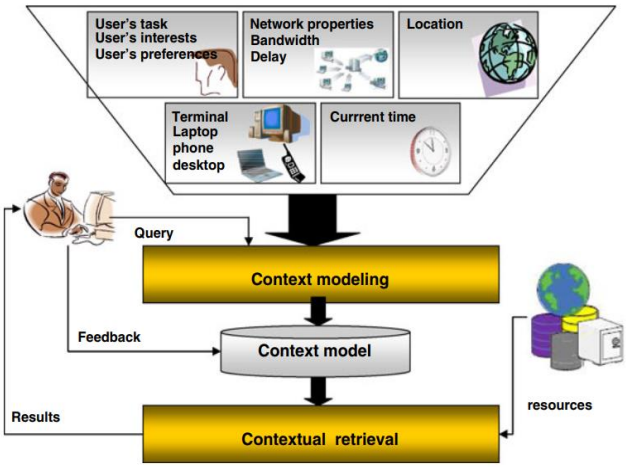


Figure. 2.12 A basic architecture of the contextual IRS (Tamine et al. 2009)

In the following paragraphs, we give an overview of previous related works.

2.3.3.3 Context Modeling

Context modeling has been long acknowledged as a key aspect in information retrieval. This task aims to address two main problems: how to model the user's context and how to exploit it in the retrieval process to provide the user with information that is of most value in his context. In contextual IRS, the user's behavior can be predictably grounded on his past interactions with the system. Consequently, to predict the user needs, the system must select and use only those preceding interactions that were offered within the same context (Anand and Mobasher, 2007). Thus, the system's effectiveness depends strongly on the available quality and quantity of information and its preferences about the user. In order to make this information available, three key points characterize a context modeling approach that should be undertaken as processes: (1) *the acquisition process* for capturing the information characterizing the user context which is called usually user profile (2) *the representation process* aimed at representing formally the acquired information and facilitate the access to the information, and (3) *updating process* is finalized at learning the changes of the user preferences in time.

2.3.3.4. Contextual Retrieval

In Allan (2002), contextual retrieval is defined *as' combining search technologies and knowledge about query and user context into a single framework to provide the most appropriate answer for user's information needs*. Thus, contextual IR aims at optimizing the retrieval accuracy by involving two related steps: appropriately defining the context of user information needs, commonly called search context, and then adapting the search by taking it into account in the information selection process. Moreover, IRS will be more intelligent, dynamic, flexible, and effective at changing their results in different contexts. They will also be more adaptive to the user's information need and preferences, making the search task easier. Indeed, the inclusion of context in the IRS reduces the complexity of the retrieval process and increases its performance, especially when quality contextual variables or factors are used (Mylonas and Avrithis, 2005), Dourish (2004), (Ingwersen and Järvelin, 2017). In Table 2.7, we illustrate a comparison between classic information retrieval systems versus contextual information retrieval systems.

Table 2.7. Differences between classic information retrieval systems versus contextual information retrieval systems

Classic Information Retrieval Systems	Contextual Information Retrieval Systems
– Context free. – Focus on the topic of the search – Ignore the specific user's search context – No supporting of individual differences that exist between users. – Static. – Low understanding of results. – No cognitive view – Difficulty in evaluation	– More intelligent. – Need a Pre-knowledge information. – Better match results. – Individual or group based – More flexible with differences of the context. – Dynamic – Cognitive; Meaningful, Comprehensive. – Reduce the complexity of IRS process. – Support the predicting of information need. – More sophisticated methods to measure the relevance – Short queries are disambiguated – Optimize the retrieval accuracy – Social interaction is encouraged – High retrieval precision is improved

2.3.3.5 Uses of Context in IRS

The taxonomy presented by Ingwersen & Jarvelin's Ingwersen (2006) is the most appropriate to many context IRS. However, it does not cover uses of context, only Dey & Abowd (Dey and Abowd, 2000) has proposed such use. In this section, we present a summary of some categories of uses of context in IR.

The context can be used in: (1) *ranking*; (2) *user interfaces*, which is divided into two subcategories: the interface associated with the specification of the user's information need and the presentation of the result set. This last category is also divided in document surrogates (e.g., snippet - short summary of the document), query term hits within a document (e.g., keyword-in-context snippets), categories for results set the context, and another type of strategies; (3) *indexing & searching*; (4) *relevance feedback* (RF) which is divided into global and local methods, global methods include query expansion/reformulation based on collection-independent knowledge structures and local methods, like pseudo relevance feedback, implicit relevance, adjust the query with information from the documents that belong to the result set of the initial query and (5) query expansion.

2.3.3.6 Context Levels in IRS

To better understand the integration of context in IRS, it can be seen in many essentials' levels or aspects such as:

- *Information environment level*: context can be interpreted as the information environment within which information behaviors take place.

- *Information seeking level*: the information-seeking context level comprises the goal(s) that a person is trying to achieve or his intentions related to a problematic situation.

- *IR interaction level*: explores the user-system interaction within search sessions, Where the role of feedback relevance, judgments as related to characteristics of users' situations can be investigated.

- *Query level*: can be described as a linguistic level of context which explores IR system performance on user queries, including query disambiguation as terms in the query are interpreted correctly or incorrectly by the system depending on the context assigned to them.

2.3.4. Discussion and Recommendation

According to the previous analysis, contextual IRS are based on different aspects. The core concept of is about the user's context, where modeling approaches must provide the most useful information needed to enhance the searching results. A common major challenge that can be seen is the of the user context modeling.

In fact, the availability of a user model, where the relevant information that characterizes the user context is represented, is necessary to define any process meant at adapting, based on this context description, the results proposed as an answer to a user's query. The quality of the personalization process is strongly correlated to the quality of the user's model, e.g., to its accuracy and reliability.

There is difficulty capturing and representing knowledge about user context, especially that the context is extremely dynamic and can continuously change even for the same user or during the same search in the process. This means that even based on correctly learned user preferences, the IRS could make wrong suppositions or get importunate. The same case can happen when users have multiple interests. Independently of the decision about how to capture and exploit the knowledge of the user's preferences, another interesting aspect that has emerged in contextual IRS, is the availability and accessibility of a context model. It may represent both user's preferences and other contexts factors. Since contextual IRS is firmly based on user-related and user personal information, the discussed aspect raises another issue: the user's privacy that should be preserved. Many users do not prefer providing their private information.

In the following, we suggest some context model's requirements that address some of the discussed issues and others.
First, in order to improve retrieval results, using the long-term user interest profiles can be efficient. To cope with this, a context-modelling framework has to identify all relevant and useful contextual categories. An interesting research direction related to personalized search is to make the user an active player in determining such an aggregation scheme: this could be simply done by making the aggregation dependent on the user's preferences over the single relevance dimensions.

In this way, for a same query and a same profile, different results can be obtained based on the user's preference over the relevance dimensions. Thus, the system will allow users to explicitly provide context information, and at the same time a context should basically be recognized automatically. The knowledge about the actual context of a user should be used for two aims: first to enhance any information created, modified, or published by the user, and second, to offer this information in contexts that are similar to the current user's context. Moreover, the time is a very important context dimension. Taking into consideration this dimension, can enhance more the personalized search. It defines the user needs according to the time as: time spent (batch), instant need (interactive) and persistent need (personalization). The time factor depends strongly on the task dimension and it is used in order to treat the evolution of tasks over time.

To address the problem of context modeling, user-dependent aggregation schemes should be defined as linear combinations where weights of relevance context dimensions are automatically computed based on the user-specified priority order over the context dimensions (Da Costa Pereira et al. 2009). Concerning the privacy issue, client-side applications can be useful where some architectures are used to enforces user privacy on social networking websites by shielding a user's personal information from the site and from other users that were not explicitly authorized by the user (Ghosh and Swaminatha, 2001) Kobsa (2007) (Luo et al. 2009).

Researches on the contextual IR field have proven that the state when the user conducts a search has a perceptible effect on the user's search behavior. The search context may include several dimensions such as time, location, history of interaction, current task, etc. In the IR field, it has taken a very important part of research aiming to improve the relevance of the search results. The evolution of IR offers opportunities to explore new research path. For this reason, the search context should be studied further in order to discover new dimensions and to improve further the information search.

2.5. Summary and Conclusion

In this chapter, we have presented our thesis' examined research fields. It presents interesting findings that can be useful for future works regarding AKPE. The studied reviews show that there is still a long trail to walk in the AKPE field.

First, we have given a survey and trends on the automatic keyphrase extraction task. We discussed some areas to which keyphrases could be beneficial. Our analysis focuses on the overall automatic keyphrase extraction process. Many approaches were analyzed (supervised, unsupervised and deep learning) and have proven their efficiency on different datasets. However, during this study, more than 34 existing techniques revealed that there are several major challenges (related to every stage of the AKPE process) ahead.

We have suggested several recommendations regarding each stage of this task to (i) address these challenges, and to (ii) enhance this task in future works. As a result, we propose in the next chapter work to develop and implement a new method for the AKPE task.

The context becomes a common notion in Information Retrieval, and has been recognized as a significant challenge in the field. **In the second part of this chapter**, we have provided a review about IR systems' evolution from the classic to the contextual scope. We have presented the concept of context, the variety of its categories, levels, and uses. We also showed how to model the context in the IR. Although many works focused on contextual IR systems, some related significant challenges are still remaining. For this reason, we discussed some context IR modeling issues, and proposed some practical recommendations to enhance these issues for future research.

Despite having more than 50 years old, AKPE is still rising, with great interest among the research community. Indeed, this research field is very dynamic, since it is continuously adapting itself to the new needs and challenges. Although the performance of AKPE is still moderate, it can be seen how their combination with other systems (e.g., IR and search engine) leads to more improvement, and helping in the development of more intelligent systems. All the possibilities that AKPE offers together with the extensive application it has in the real world make this task an interesting research area. Therefore, the acquisition of the background of AKPE, which was the main goal of this chapter, is essential.

3

HAKE: An Unsupervised Approach for Automatic Keyphrase Extraction

3.1. Introduction

In the first part of the previous chapter, we have analyzed in depth the state-of-the-art of the AKPE task. Not surprisingly, the number of publications related to AKPE has been significantly increased in the last decade. This inconsistency may be due to the AKPE importance in many mentioned research fields. Definitely, retrieving relevant information from a large textual data scale and dealing with its complicated content has affected interested researchers who try to take benefits from the available knowledge in NLP and machine learning to improve the AKPE task.

Despite the huge literature dealing with keyphrase extraction, recent studies have shown that these methods' accuracy remains moderate (Merrouni et al., 2019) (Hassan, 2014) (KIM, 2010), and several examined techniques revealed that there are quite major challenges (related to every stage of the AKPE process) ahead. These issues remain as essential preliminary challenges to be handled during the establishment of an AKPE method. One of the significant issues related to the automatic keyphrase extraction is the selection of good (the morphologic form) candidate keyphrases; the redundancy induced by semantically equivalent keyphrases; the infrequency of semantically related keyphrases where many systems overlook the infrequent, relevant keyphrases, and focus on frequent words in associated documents; some techniques depend on large annotated data, which is time-consuming and not always available. This finding has important implications for developing a new AKPE system that tackles most of these challenges.

As we noticed in the first part of chapter 2, all of the AKPE techniques require the presence of a candidate keyphrases selection stage to produce accurate and good (the morphologic form) candidate keyphrases. However, the selection of these candidate keyphrases is not a trivial task. It must be carried out carefully since the performance of an AKPE depends mainly on the usefulness of the used method(s). Besides, it is commonly known that the AKPE process works much better for those systems on which they used robust candidate selection methods. This stage presents an important challenge for an AKPE, which has to be addressed to enhance the AKPE system performance. Generating statically and semantically relevant keyphrases from any domain of a document is a crucial step too. During this step, it is crucial to acquire informative and pertinent keyphrases that represent the real information need of the user. These two stages will be tackled in this chapter.

Therefore, our proposed methodology is twofold: first, we propose a novel candidate keyphrases' selection method, based on a parse tree approach, part of speech tagging, and filtering. This approach allows us to generate a comprehensive and meaningful list of candidate keyphrases and reduce their set size without increasing the computational complexity. Second, a novel method is proposed to takes full advantage of mining simultaneously hybrid (linguistic, statistical, structural, and semantic) text features extracted from documents to select the most relevant of these selected keyphrases. This method considers the relevance of keywords within candidates that are used later for ranking the top k keyphrases.

Our methodology does not require any training on a particular set of documents, nor does it depend on external corpora, dictionaries, domain, or text size.

Section 3.2 presents the aims of the proposed method and some related works of AKPE in which we compared our proposed work. In section 3.3, we present our approach to select informative candidate keyphrases and extract the relevant ones. To show the effectiveness of our proposition, section 3.4 is dedicated to experimentations and evaluation strategies to evaluate the proposed approach, which is conducted on some of the top available benchmark datasets from different domains. The obtained results and discussion are covered. In section 3.5. Finally, we discuss the contribution of the approach while giving some perspectives in section 3.6.

3.2. Aims and Related Works

As mentioned in the literature review, AKPE is defined as the automatic extraction of relevant phrases from a document. However, several challenges facing AKPE techniques. On one hand, there are challenges such as techniques which are adopting only statistical analysis in the extracting of keyphrases and miss the semantic ones. Several existing supervised approaches use different features and machine learners that classify keywords as relevant or non-relevant. Whereas these models have some advantages, they require a long training process and large manually annotated data. They depend on the pre-determination of features before running the system. They use offline analysis and require a considerable amount of computing time for training.

As a result, these features cannot be modified or learned during the system process. It is difficult to list all the features associated with a specific domain because it is necessary to exploit the text's textual, statistical, and external information (Merrouni et al., 2019). On the other hand, unsupervised models perform without any prior knowledge or manually labeled data, and they can be easily applied to a document across different domains. We have mentioned in the first part of Chapter 2 that the task of keyphrase extraction includes three crucial steps. First, candidate keyphrases identification: all the phrases of a document are filtered to identify candidate terms that could be keyphrases.

Second, feature selection: a set of features (properties) are used to discriminate keyphrases and their relevance from other terms. Third, keyphrase ranking: these features are combined using machine learning models (supervised or unsupervised) to give a synthetic score for each keyphrase. This score is used to rank the candidate keyphrases, and the k-top-ranked keyphrases are output as the keyphrases we look for.

In every proposed method, identifying and selecting candidate keyphrases is an essential task, and thus the robustness and effectiveness of methods depend primarily on the underlying candidate keyphrase selection technique. A comprehensive and syntactically correct list of candidate keyphrases may ensure a robust set of keyphrases (Merrouni et al., 2019). Conventionally, n-gram or noun-phrase based techniques are used for the extraction of candidate keyphrases. However, there are certain limitations on existing approaches that make it difficult to obtain candidate phrases that are grammatically correct.

An intuitive idea is that keyphrase(s) must contain keyword(s) (You et al., 2013). Here, keywords mean the most important single words in the document. They can be used as the basis for measuring candidate keyphrases. Weighting words first, instead of phrases, offers many advantages. They are usually much easier to extract, match, and weight, especially in documents where many phrases may not be statistically frequent (Kim et al., 2010) (Liu et al., 2010). They can reduce the over-generation errors for AKPE by ensuring that the keyphrases are selected according to their component words.

Motivated by these finding, we present in this chapter 'HAKE', a novel unsupervised Automatic Keyphrase Extraction method, to identify and extract relevant keyword and keyphrase(s) from text documents. It deals with documents in complex and multi-domain topics. Precisely, our work mainly improves the first two steps of the AKPE process by adopting a twofold approach. First, we identify and form comprehensive and syntactically correct candidate keyphrases using a novel method that relies on a parsing-based technique and part of speech tagging (POS) technique. Second, several features are calculated and combined to identify relevant keyword(s) using a novel weighting model named 'KWHFSM' (Keyword Hybrid Features Selection Model). Precisely, this model calculates the weight of single keywords that are used later to rank the generated candidate keyphrases from the first step. It considers three kinds of features: the statistic feature, namely the CHIR measure (an improved version of the Chi-Square Statistic), the structural feature: the TPW (Term Position Weight), and semantic feature: SIM measure (a Mutual Information based method). Finally, another lightweight model called 'KPHFSM' (Keyphrase Hybrid Features Selection Weighting Model) is built to rank and select the top keyphrases' final list.

Experimental results show that our system outperforms other baseline systems (supervised and unsupervised) on five referenced datasets.

Typically, the automatic keyphrase extraction task is performed in three main steps: 1) identifying a set of appropriate phrases as candidate keyphrases, 2) using features to discriminate keyphrases from other terms, and 3) extracting final keyphrases from the set of candidates by means of a supervised or unsupervised approach. In what follows, the related works of each step are discussed.

3.2.1. Candidate Keyphrase Identification Methods

The first step, 'candidate phrase identification' or 'candidate phrase generation', aims to identify a set of words serving as candidate keyphrases. This step reduces the number of false-positive candidates while maintaining the true-positive ones. The majority of methods usually fall into two categories: (i) N-Gram based, (ii) Part-Of-Speech (POS) sequence-based, or both. The first category was adopted in pioneering systems (Frank et al., 1999)

(Turney, 2000). The input text is separated according to phrase boundaries and to a limited length (bigram, trigram, . . .) (Witten et al., 1999) (Turney, 2000). All subsequences of these initial phrases up to a limit length are taken as candidate phrases; then, some rules are applied on these candidates to filter meaningless subsequences (Huang et al., 2006) (Liu et al., 2 009) (Campos et al., 2020). However, n-grams are most likely to be grammatically incorrect and do not always capture complete information. The alternative solution to this problem is the use of specific linguistic patterns based on part-of-talk (POS) tags or noun phrases (Hulth, 2003) (Haddoud and Abdeddaim, 2014). The second category is also widely used. It allows words with finite part-of-speech tags (e.g., nouns, adjectives, verbs) to be candidate keyphrases (Barker and Cornacchia, 2000) (Nguyen and Kan, 2007) (Haddoud and Abdeddaim, 2014). Hulth (Hulth, 2003) compares three different phrase identification methods and concludes that extracting noun phrase chunks and phrases with a finite POS tag sequence gives better results than using n-grams. Grineva et al. (Grineva et al., 2009) use n-grams that appear in Wikipedia article titles as candidates. Haddoud and Abdeddaım (Haddoud and Abdeddaim, 2014) identified five existing POS tag sequence definitions of a noun phrase and proposed a new one that reveals a noteworthy improvement over the other filters. Recently, Bennani-Smires et al. developed a technique based on the noun phrase approach using linguistic patterns of POS tags (Bennani-Smires et al., 2018). It selects phrases that consist of zero or more adjectives and followed by one or more nouns. Various heuristics rules obtained from training sets or thesaurus have also been used to identify and extract candidate keyphrases for long textual entities (Huang et al., 2006) (El-Beltagy and Rafea, 2009) (Newman et al., 2012).

3.2.2. Candidate Keyphrases Scoring and Ranking Approaches

This step of automatic keyphrase extractor aims to score the candidate keyphrases. We highlight both supervised and unsupervised approaches for this purpose (see Table 3.1).

Table 3.1 Some prominent systems for keyphrase extraction

Technique/ Contributor	**Approach**	*Features*
KEA System (Witten et al., 1999)	Supervised	*Term First occurrence position, tf -idf*
GenEx Algorithm (Turney,2000)	Supervised	*Term First occurrence position, tf -idf, phrase length*
TextRank Algorithm (Mihalcea and Tarau 2004) (Mihalcea and Tarau, 2004)	Unsupervised	*POS, distance between word occurences*
CorePhrase Algorithm (Hammouda et al,2005)	Unsupervised	DF-average weight, Phrase frequency, Average phrase depth
SingleRank Algorithm (Wan and Xiao,2008) (Wan and Xiao, 2008)	Unsupervised	POS tagging, neighborhood, Level saliency score
Maui Algorithm (Medelyan et al. 2009)	Supervised	tf–idf, position of first occurrence, Keyphraseness, Phrase length, spread, node degree
KP-Miner System (El-Beltagy & Rafea, 2009)	Unsupervised *(Weighting model)*	*tf-idf, first occurrence position, Boosting Factor for weight bias*
Core Word Expansion Algorithm, (Barthès et al., 2009)	Supervised; Unsupervised	Inverse document frequency (idf), N-gram
CommunityCluster Algorithm (Grineva et al.,2009)	Unsupervised	N-Gram
Keycluster Algorithm (Liu et al.,2009a)	Unsupervised	Co-occurrence-based term relatedness (idf), N-gram.
RAKE (Rose et al., 2010)	Unsupervised	term frequency; term degree (the number of different terms that it co-occurs with), and ratio of degree to frequency.
(Sarkar et al., 2013) Algorithm	Supervised	Noun Phrase, Phrase frequency, phrase links to other phrases, idf, phrase position, phrase length and word length
CeKE Algorithm (Bulgarov and Caragea, 2015)	Supervised	*tf – idf; relativePos, POS; inCited, inCiting; citation tfidf, first position, tf-idf-Over, firstPosUnder, keyphraseness*
KeyphraseDS System (Yang et al.,2016)	Unsupervised	Syntactic features (POS and Relation), correlation features (MutInfoPre, MutInfoNext), semantic relatedness
KeyEx Method *(Xie et al. 2017)*	Supervised	*tf -idf ; POS; first occurrence position, statistical pattern features (pattern support, pattern length,number of closed patterns, P F IDF value)*
TSAKE (Rafiei-Asl and Nickabadi, 2017)	Unsupervised	*N-gram, co-occurrence graph*
YAKE! (Campos et al., 2020)	Unsupervised *(Weighting model)*	*word position, word frequency, word relatedness to context, and word in the different sentences.*

- *In unsupervised approaches*, the AKPE task is considered as a ranking problem and performs without prior knowledge. There is no need to build any manually labeled data, which requires considerable time and human resources. Most unsupervised keyphrase extraction techniques could be broadly divided into two groups, namely statistical and graph-based methods. These approaches usually assign a salience score to each candidate phrase by considering various features (Wan, 2008). In the statistical method, Term Frequency - Inverse Document Frequency (TF-IDF) is the most prominent and state-of-the-art statistical technique (Jones, 1972), which compares the frequency of a term in a document to its frequency in a large collection. Rabby & al. (Rabby et al., 2018) used nominal statistical knowledge to select keyphrases.

  Matsuo & Ishizuka (Matsuo and Ishizuka, 2004) considers the relevance of words if they co-occur in the document more frequently than randomly distributed in the document. They used the $\chi 2$ values to determine the biases between expected and observed frequencies.

  Similarly, word co-occurrence models manage statistical information about the number of times a word has to occur and the number of times it has occurred with another word. This statistical information is then used to compute the support of words and confidence. Another feature that has shown significant improvements in different keyword extraction systems is the CHIR Statistic (Li et al., 2008) (Frikh et al., 2011). It is basically based on the well-known chi-square $\chi 2$ statistic measure. It determines whether a term and a category are statistically independent by comparing their observed frequency and the expected one (Matsuo and Ishizuka, 2004). Despite its simplicity, the ranking based on tf-idf and co-occurrence measures has been shown to work very well in practice (Hasan et al., 2014) (Merrouni et al., 2019). Chen et al. (Chen et al., 2005) use three new features (phrase frequency, average term frequency, and Okapi feature that involves tf part, idf part and a document length part) considering web pages' structure. Wang and Peng (Wang and Peng, 2005) present mark-up features to extract keyphrases from web pages. Nguyen and Kan (Nguyen and Kan, 2007) consider that keyphrases are distributed non-uniformly in different sections of a scientific paper. They propose section occurrence vectors ('in title' and 'in a section occurrence') favoring sections like introduction and related work. Kumar and Srinathan (Kumar et al., 2008) use the position of phrases in a sentence and position of a sentence in the document. Berend and Farkas (Berend and Farkas, 2010) present features that consider the document, corpus, and knowledge-based levels. They also use external knowledge sources (such as Wikipedia) to achieve more improvements in performance. Medelyan and Witten (Medelyan and Witten, 2006) propose the KEA++ system that used semantic information on terms and phrases generated from a domain-specific thesaurus. Adar and Datta (Adar and Datta, 2015) extract keyphrases by mining abbreviations from scientific literature and composing those keyphrases into semantically-meaningful hierarchies to build a domain-specific keyphrase database. El-Beltagy and Rafea (El-Beltagy and Rafea, 2009) proposed the KP-Miner system, an unsupervised non-learning ranking-based system for keyphrase extraction, which relies on a three-step process: candidate keyphrases selection, candidates weight calculation, and finally keyphrase refinement.

To determine the importance of a keyphrase, this system assigns scores to phrases based on a modified version of tf-idf, and two boosting factors (position of the first occurrence, and word length that increases the chance of longer phrases being selected). Recently, a closest prior research to our proposal, especially in the second AKPE step, Florescu and Caragea (Florescu and Caragea, 2017) proposed a new scheme for scoring phrases that calculates the final score using the scores' average of individual words weighted by the frequency of the phrase in the document. Tomokiyo and Tomokiyo and Hurst (2003) propose an approach that extracts keyphrases based on statistical language models using the point-wise KL-divergence between multiple language models. Rose et al. (Rose et al., 2010) presented RAKE, an independent, unsupervised, single-document domain, and language approach. It considers term sequences (delimited by stopword or phrase delimiters) as candidate keywords and builds a matrix of term co-occurrences on this basis. Term scores are then determined for each candidate keyword as the sum of its member word scores, based on the degree and frequency of the words in the matrix: (1) the frequency of the term; (2) the degree of the term (the number of different words in which it co-occurs) and (3) the degree-to-frequency ratio.

Most recently, Campos et al., proposed YAKE! (Campos et al., 2020), an unsupervised system which is based on a lightweight technique. It takes five features into consideration, namely casing, word position, word frequency, word relatedness to context, and word in the different sentences to calculate a keyphrase's weight. However, due to generating candidate keyphrases employing the N-grams technique, its computational complexity increases linearly with respect to N-grams. Consequently, a large number of keyphrases are generated, which entices the ranking procedure. Rabby et al., (2020) proposed 'TeKET', a new unsupervised tree-based Keyphrase Extraction Technique, which is domain and language independent. It employs limited statistical knowledge, but no train data are required. CorePhrase (Hammouda et al., 2005) system constructs a list of keyphrase candidates by comparing every pair of documents. It scores each candidate according to some criteria (df: document frequency, average weight, average phrase frequency, and average phrase depth), and ranks keyphrases according to their score. Liu et al. (Liu et al., 2009a) involve an unsupervised clustering technique (henceforth Keycluster). They assign an importance score to a word, and then pick the top-ranked words as keywords by clustering semantically similar candidates using Wikipedia and cooccurrence-based statistics. In the KeyphraseDS system (Yang et al., 2017), keyphrases are extracted through a conditional random field (CRF) based model exploiting various features such as syntactic features and correlation features. Spectral clustering is then performed on a keyphrase graph to generate different aspects, where the semantic relatedness between keyphrases is computed through knowledge-based similarity and topic-based similarity. Boudin (Boudin, 2015) proposes a weighting model applied to both supervised and unsupervised word weighting function. In their inference model, the sets of candidates are weighted according to their component words. He used different weighting functions such as tf-idf, TextRank, and a logistic regression model to assign importance weights to words in the document.

Graph-based methods offer an alternative solution to purely statistical methods by building a graph from an input document and rank its nodes according to their importance.
TextRank algorithm (Mihalcea and Tarau, 2004) is one of the most well-represented graph-based models. It represents a document as a graph. Each vertex in the graph corresponds to a word (that respect a certain PoS), and there is an edge between any two words co-occurring together within a window of n terms. A ranking algorithm is then performed on top of this graph, and keywords are sorted by decreasing order. Later, several variations of TextRank are proposed, such as SingleRank (Wan and Xiao, 2008), and ExpandRank (Wan and Xiao, 2008), the latter being enriched by terms of the k-nearest neighboring documents. An expanded version of this neighborhood approach was proposed in the CiteTextRank algorithm (Gollapalli and Caragea, 2014), which extracts keyphrases from research articles. This graph-based algorithm incorporates evidence from both (1) document content, and (2) contexts in which the document is referenced within a citation network. This evidence is used flexibly to score keywords that are later used to score keyphrases. CiteTextRank builds a graph to compute scores (of nodes or vertexes) using information from multiple contexts. Another enhancement of TextRank is TopicRank (Bougouin et al., 2013). TopicRank extracts noun phrases that represent the main topics of a document and clusters them into topics. Consequently, documents are represented as a graph, where nodes instead of terms are topic clusters of single and composed expressions.

Later, Sterckx et al. propose TopicalPageRank (Sterckx et al., 2015) that runs TextRank multiple times for a document, once for each of its topics generated by a Latent Dirichlet Allocation LDA. Consequently, TPR ensures that the extracted keyphrases cover the main topics of the document. The final score of a candidate is computed as the sum of its scores for each topic, weighted by the probability that a given topic is in the document.

More recently, Florescu and Caragea (Florescu and Caragea, 2017) proposed an unsupervised graph-based model called PositionRank, which incorporates information from all positions of a word's occurrence and its frequency in a document score a biased PageRank score for each word. Boudin proposed the MultipartiteRank technique Boudin (2018) that utilizes a multipartite graph. Here, this model encodes topical information within a directed multipartite graph structure, in which nodes are keyword candidates will be connected if only they belong to different topics.

- *In the supervised approaches*, keyphrase extraction is considered as a binary classification task, where each candidate keyphrase in a document is determined as a keyphrase or not (Frank et al., 1999) (Turney, 2000). Supervised methods have mainly focused on (1) selecting features that measure candidate keyphrases, and (2) running learning models to determine keyphrases from non-keyphrases. These models should be trained on manually annotated keyphrases (which may sometimes not be practical, and time consuming). The used features can be classified as: statistical (Liu et al., 2009a) (You et al., 2009) (Nguyen and Luong, 2010) (You et al., 2013) (Ong and Chen, 1999) (Cohen, 1995) (Ramos et al., 2003), linguistic (Barzilay and Elhadad, 1999) (Krapivin et al., 2008) (Krapivin et al., 2010) (Adar and Datta, 2015) (Hulth,

2003) (Salton et al.,1997) (Haddoud and Abdeddaim, 2014) (Le et al., 2016), structural (Nguyen and Kan, 2007) (Kumar et al., 2008) (Lopez and Romary, 2010)., semantic (Chen et al., 2019) (Dagan et al., 1993) (Kelleher and Luz, 2005) (Li and Park, 2009) (Song et al., 2014), or hybrid ones (Sarkar, 2013) (Yang et al., 2017) (Frantzi et al., 1998) (Haddoud and Abdeddaim, 2014). Different learning models are used in the supervised systems such as SVM (Jiang et al., 2009), Maximum Entropy (Yih et al., 2006), Bagged C4.5 (Turney, 2000), Naïve Bayes (Witten et al., 1999) (Frank et al., 1999) (Xie et al., 2017), Bagged Decision Tree (Lopez and Romary, 2010), Logistic Regression (Haddoud and Abdeddaim, 2014), Neural Networks (Sarkar et al., 2010).

As mentioned in the first part of chapter 2, using different features turns out to be very useful for the AKPE task. Indeed, the best-performing systems (El-Beltagy and Rafea, 2009) (Lopez and Romary, 2010) (Haddoud and Abdeddaim, 2014), and YAKE! (Campos et al., 2020) have combined at least two types of features to extract and rank relevant keyphrases. Hence, any AKPE system's success depends mainly on the used candidate keyphrases selecting technique, and the pertinence, number, and the variety of the used features. In the following we present our proposed AKPE method.

3.3. A New AKPE System

In this section, we describe HAKE, our unsupervised Hybrid Automatic Keyphrase Extraction method, where our system's input is a collection of text documents, and the output is a set of relevant keyphrases of every single document.

The System Architecture

Our proposed method is intuitively based on the following assumptions: (1) a selected candidate keyphrase should be comprehensive and syntactically correct, (2) a keyphrase should be relevant if it contains one or many relevant keywords, and (3) a keyword should be relevant if it is selected and ranked with a high weight. Thus, we adjust a keyphrase's weight by its keyword's salience scores (see Algorithm 1 and Figure 3.1).

HAKE system consists of four main steps:

- Text pre-processing and candidate keyphrase identification: The first step, pre-process the input documents, then identify and form comprehensive and syntactically correct candidate keyphrases using a novel method that relies on a parsing-based technique and part of speech tagging.

- Feature extraction: Several features selection criteria (statistic, structural, semantic) are calculated to select relevant individual keywords in the second step.

- Computing keyword score: in the third step, these selected features values are combined using a novel lightweight model 'KWHFSM' (KeyWord Hybrid Features Selection Model) that gives a synthetic score to each keyword (see Subsection 3.3.2) to reflect the importance of the term.

- Post-processing: the goal of this step is to compute candidate keyphrases score, and rank them. The keyword's scores or weights are gathered using a new scheme to obtain scores for the candidate keyphrases that have been identified from the first step. Then, the weighting model namely 'KPHFSM' (KeyPhrase Hybrid Features Selection model) is performed to rank the keyphrases. The candidate keyphrases with the highest KPHFSM score are ranked as more relevant keyphrases. Algorithm 3.1 and Figure 3.1 describe the steps of the HAKE system.

Algorithm 3.1 HAKE general Algorithm

Input: $D = \{d_1, d_2, \ldots, d_n\}$: a set of n documents

Output: Set of keyphrases of each document

-Let's **KP**(s) be the candidate keyphrases of D and $KP_{i,d} = w_1, w_2, \ldots, w_n$ is the *i-th* keyphrase of a document d containing n words.

-Let's ***kwhfsm*()** be the 'keyword hybrid feature selection model', an algorithm that compute keyword score, and ***kphfsm*()** be the keyphrase hybrid feature selection model, an algorithm that calculate the score of candidate keyphrases.

1: **# (Step 1) Text pre-processing and candidate term identification**
2: **for each** $d \in D$ **do**
3: Pre-process d
4: Identify and extract all candidate keyphrases
5: Build candidate keyphrases List
6: **end for**
7: **# (Step 2) Feature extraction & (Step 3) computing keyword score**
8: **for each** *word* **in** *sentences* **do**
9: Feature extraction
10: Perform the *kwhfsm*()
11: Rank the relevant keywords w_i in a descending order of their criterion function
12: *List* (w_i, *score*)
13: **end for**
14: **# (Step 4) Computing candidate keyphrases score and ranking**
15: **for each** *candidate* in *candidate keyphrases List* **do**
16: Compute candidate keyphrases weight by performing the *kphfsm*()
17: List ($KP_{i,d}$,*d*, *score*)
18: **end for**
19: **# (Ranking candidates)**

20: **for each** $KP_{i,d}$,*d in List* ($KP_{i,d}$,*d, score*) **do**

21: *Keyphrases* = *sort* (candidate keyphrases) *by descending score*

22: **end for**

23: Return the k-top ranked $KP_{i,d}$

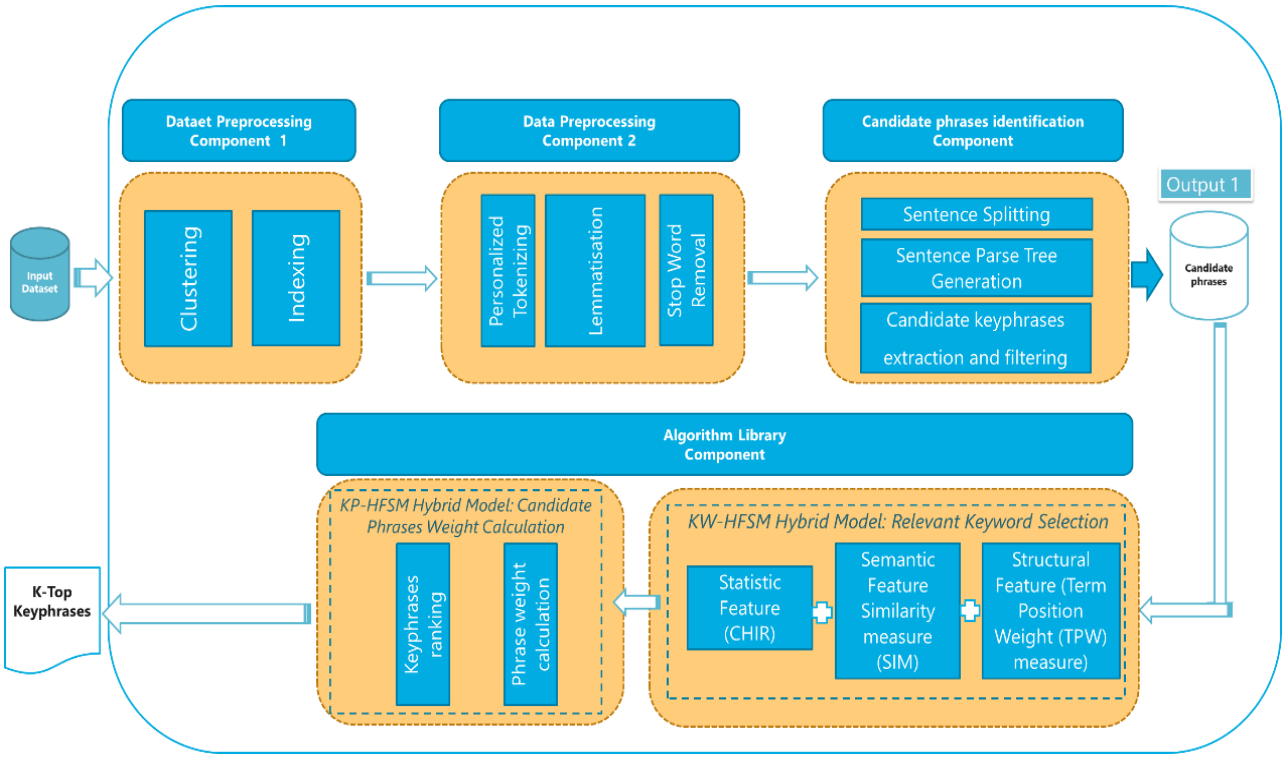


Figure. 3.1 Schematic diagram of HAKE system for extracting keyphrases

Further details about each step are presented in the following sub-sections.

3.3.1. Identification of Candidate Keyphrases

In this subsection, we present our method for identifying candidate keyphrases. The phrases have a stronger ability for representing text topics compared with simple words. However, phrases have a complex grammatical structure. Hence, the first problem, we have to deal with, is phrase identification, because it will be hard to let the AKPE algorithm know what a phrase is if phrases do not have a clear boundaries and structure.

Some previous works usually include all possible n-grams, but the number of remaining candidates can be very large. Some other systems use syntactically motivated boundaries such as NP-chunks and POS tags sequence, which decrease the number of candidate phrases. Our method aims at obtaining a refined candidate set benefiting from the best identification techniques. To achieve this goal, we

propose a novel technique (based on a parse tree approach, part of speech tagging, and filtering) to generate a comprehensive and meaningful list of candidate keyphrases.

3.3.1.1. Documents Pre-processing

Identifying the candidate terms of each document needs to pre-process the text. Text Pre-processing (TP) aims to regularize the input files and set phrase boundaries for word sequences. The way, this classical step is done can significantly affect the accuracy of the keyphrase extraction method. Thus, it is important to give some details on how we have implemented it. Each document goes through the following pre-processing stages:

- *Personalized Tokenizing*: for each sentence, we tokenize it by eliminating all marks, punctuation, brackets, numbers and special characters along with lower-casing all words. For example, given the sentence: (*text miningInstA˜ance*; , ; :, *Ãare*; *ƒea = tureÃs*), the result will be: (text mining instance are features).

- *Stop word removal*: stop words helps in reducing time and feature space complexity. They include a large number of prepositions, pronouns, conjunctions in sentences. These words need to be removed before we analyze the text so that the frequently used words are mainly the words relevant to the context and not common words used in the text. In particular, we exclude from the list some words that may occur as part of keyphrases (e.g., “in” “of”, “for”, “and”). The reason for keeping such conjunctions and prepositions is that the lack of such words may impact the readability of the final result caused by grammatical mistakes.

- *Lemmatization*: the term variation can affect its frequency. Lemmatization reduces inflectional forms and derivationally related forms of a word to a common base form so they can be analyzed as a single item, and identified by the word’s Lemma. Unlike stemming, lemmatization depends on correctly identifying the envisioned part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document. We use the Stanford CoreNLP library for lemmatizing our sentences (De Marneffe et al., 2006).

To recognize terms of equivalent meaning, stemming is also applied during the pre-processing step in many AKPE systems. However, in our system, we decided to stem in the later step of evaluation (see Section 4). We aim to hold the non-stemmed original input texts from which the final keyphrase can be picked up to avoid stemming grammatical complications.

3.3.1.2. Candidate Keyphrases Identification Technique

Since the proposed method does not know what a phrase is, it has to use a set of conditions to identify and generate candidate keyphrases. We propose a novel technique for candidate keyphrases extraction based on sentence parsing and POS tagging in this phase. The generated sentences are then

analyzed to identify potential candidate keywords that are the topical terms to replace the document text in its entirety. In subsequent stages, linguistically relevant candidate keyphrases are selected from this set. Figure 3.2 shows the schematic diagram of our proposed technique.

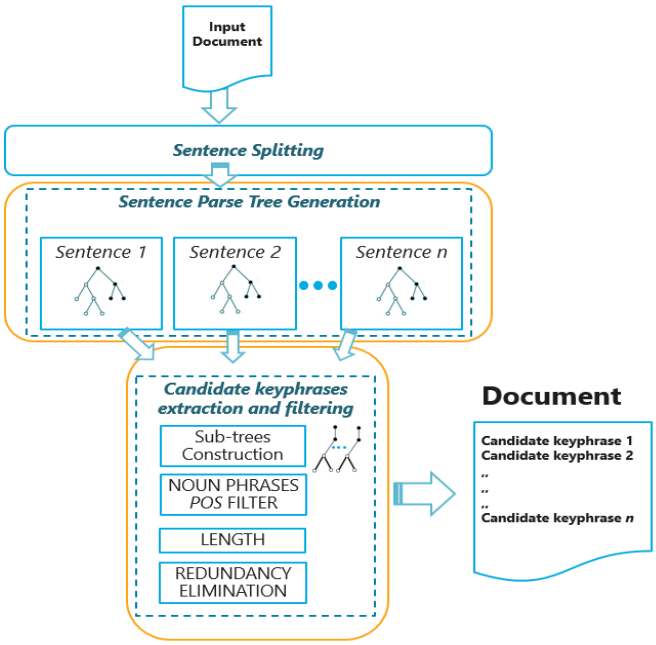


Figure. 3.2: Schematic diagram of the proposed technique for candidate keyphrases identification

The steps of the proposed technique are as follows:

- **Sentence parse tree generation:**

Sentence Splitting is a preliminary step that is commonly performed on texts before further processing is called sentence boundary detection or sentence segmentation. Namely, it is the process of dividing up the input text into sentences. We split the input document into a list of sentences. This list then passed through the sentence parser, which generates and returns parse trees for each sentence. We employ the Stanford Parser (De Marneffe et al., 2006) for this aim. By creating a parse tree for each sentence (see example in Figure 3.3), we can depict the sentence's full picture. It shows the overall structure of the sentence and its relationship to different parts. For instance, the parse tree of the lengthy sentence "grid computing has become a real alternative to traditional parallel computing, a grid provides much computational power, and thus offers the possibility to solve very large problems", is given in Figure 3.4. The tree structure list will then be passed to the next step of candidate keyphrases extraction and filtering for meaningful and comprehensive candidate keyphrases.

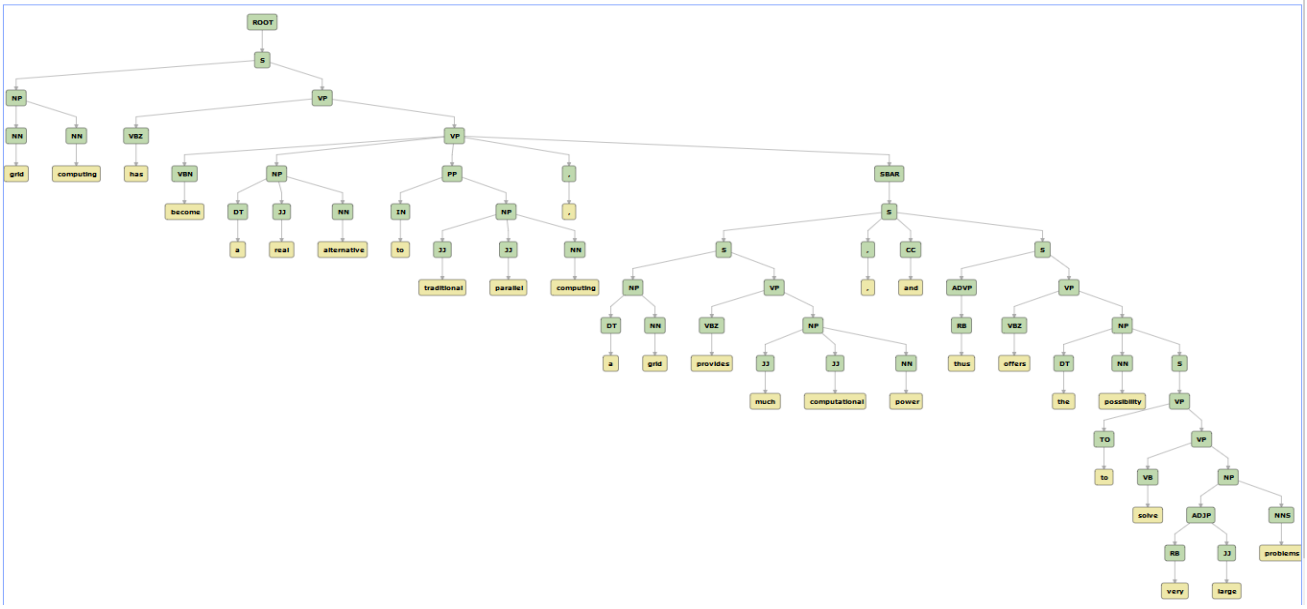


Figure. 3.3 Example of parse tree of a sentence

- **Candidate keyphrases extraction and filtering**

After parsing the document, this step aims to generate a syntactically correct and comprehensive candidate keyphrases. To do that, we extract the meaningful parts from the sentence tree structures. We cut the tree into subtree for each sentence. Then we branch out noun phrases via the filtering process. In the filtering process, the generated list of candidate keyphrases is filtered by a certain heuristic. As a result, a candidate can be included in the final list of candidates if it meets the following heuristic rules:

- Part-Of-Speech tags (POS tags) and noun phrases chunks are considered as performance and most used linguistic features that capture the linguistic properties of the keyphrases (Lopez and Romary, 2010) (Haddoud and Abdeddaim, 2014) (Merrouni et al., 2019). Indeed, selecting a well-formed candidate that should be linguistically informative, will limit and filter unlikely keyphrases. Consequently, we bound the computational complexity of the next feature selection step.

- To achieve that, we place a phrase type restriction to identify correct candidate keyphrases linguistically (Turney, 2000) (Hulth, 2003) (Nguyen and Kan, 2007) (Krapivin et al., 2008) (Wan, 2008) (Liu et al., 2009) (Haddoud and Abdeddaim, 2014). We only keep candidate keyphrases, which are considered as noun phrases. The definition of a noun phrase can differ from a method or a language to another. To select proper and accurate candidate keyphrases, we have proposed a new POS tag sequence definition of all possible noun phrases forms. We use Stanford Parser (De Marneffe et al., 2006) and Stanford POS Tagger (Toutanova et al., 2003) for these aims.

Our proposed POS tag definition of noun phrases satisfy the following regular expression rule: (NN | NNS | NNP | NNPS | VBN | JJ | JJS | RB) * (NN | NNS | NNP | NNPS | VBG) +. Unlike other definitions, our structure of noun phrases can comprise an adverbial noun (tag RB) such as “double experience” (RB NN), and a verb in present participle (tag VBG) such as: “virtual desktop conferencing” (JJ NN VBG), where the VBG tag can be at the beginning, the middle or at the end of the noun phrase. Adverbial nouns are sometimes referred to as adverbial objectives, they hold a position normally occupied by a verb’s direct object and modify the verb with an aspect of time, distance, weight, age, or monetary value. Adverbs, in general, can interact with the noun phrase.

They can change the context and the meaning of a candidate keyphrase, especially in scientific domains where authors are more precise and specific in explaining some situations.

- The candidate keyphrase start or end with a stop word.
- The Length of candidate keyphrase up to n extracted tokens (i.e., unigrams, bigrams, trigrams, four grams, etc.) without specified limit (usually with n<=5 at maximum). Nevertheless, the length parameter can be fin-tuned for one dataset to another, depending on the size of the documents and the dataset's knowledge domain.

After filtering candidate keyphrases, some of them may still contain redundant phrases, or one phrase may be part of another one. Therefore, we perform a redundancy removal that eliminates the overlapping between phrases.

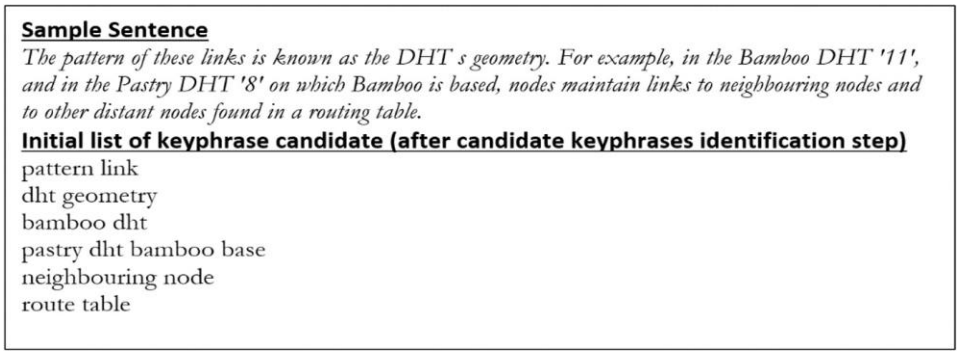


Figure. 3.4 Identified candidate keyphrases from a simple sentence

Figure 3.4 represents a sample sentence and the identified candidate keyphrases from this sentence. Sometimes, some generated candidate keyphrases using the technique mentioned above may be less meaningful to human readers. After the feature engineering step, such kind of candidate keyphrases is filtered out. For this purpose, a condition is applied. The condition is to choose a threshold on the phrase weight (phrase weighting schema presented in the next subsection).

3.3.2. Feature Selection

After the text pre-processing and candidate keyphrase identification stage, we obtain a list of

document's resulting candidate set. The success of an AKPE system depends on its capability to extract the best terms that describe the document content. This step aims at extracting a subset containing the most important single terms. Our method adopts the feature weight calculation. Feature weight is an index of phrase importance. The axiomatic idea is that keyphrases should comprise keywords. Here, keywords mean the most relevant single words in the document. However, by looking at the majority of documents, it can be observed that phrases are much less frequent than the occurrence of single terms within the same document (El-Beltagy and Rafea, 2009). To overcome this issue, our method extracts and selects single relevant keywords used later to rank the candidate keyphrases. More precisely, our keyword set is based on a keyword weighting scheme. The synthetic weight of a relevant keyword is calculated using the hybrid keyword feature selection model "KWHFSM". Then, the remaining weights of keywords are gathered to obtain scores for the candidate keyphrases. We employ a hybrid analysis based on, co-occurrence, structure, and semantic features. Consequently, the used features are divided into three groups: statistical features, semantic features, and structural features. Detailed explanations of each type of feature are given below.

- *Statistical features Measure*

In information retrieval and text mining, the χ^2 statistic is regularly used to measure the degree of dependency between a term w and a specific category c. The χ^2 term-category independence test has been widely used as a feature selection (Li et al., 2008) (Frikh et al., 2011) and defined as:

$$\chi^2_{w,c} = \sum_{i\in\{w,\overline{w}\}}\sum_{j\in\{c,\overline{c}\}}\frac{(O(i,j)-E(i,j))^2}{E(i,j)} \qquad (3.1)$$

where $O(i,j)$ is the observed frequency and $E(i,j)$ is the number of documents which are in the category c and contain the term w. However, using only the χ^2 statistic method can eliminate the terms that are fairly relevant to a category and keep the irrelevant and redundant terms, and do not provide enough information about the relationship between the selected terms and the corresponding categories. To address this problem, Li et al. (Li et al., 2008) define a new criterion for the relevancy of a term w to a category c as: the term w should have a strong positive dependency on the category c. First, to evaluate whether the dependency between a term w and a category c is positive or negative, the $R_{w,c}$ measure is defined as:

$$R_{w,c} = \frac{O(w,c)}{E(w,c)} \qquad (3.2)$$

which can be interpreted with the probabilities as follows:

$$R_{w,c} = \frac{p(w,c)p(\neg w,\neg c)-p(w,\neg c)p(\neg w,c)}{p(w)p(c)} + 1 \qquad (3.3)$$

where $R_{w,c}$ is the ratio between $O(w,c)$ and $E(w,c)$. $R_{w,c}$ should be close to 1 if there is no dependency between the term w and the category c (i.e., $\chi^2_{w,c}$ is not statistically significant), $R_{w,c}$ should be larger than 1 if there is a positive dependency, which means that the observed

frequency should be larger than the expected frequency.

$R_{w,c}$ should be smaller than 1 if there is negative dependency.

In order to calculate the feature goodness of term w in a corpus with k categories, Li et al. (Li et al., 2008) combine equations $\chi^2_{w,c}$ statistic (1) and $R_{w,c}$ measure (2) and propose a new measure called CHIR, defined as follows:

$$r\chi^2(w) = \sum_{j=1}^{k} p\left(R_{w,c_j}\right)\chi^2_{w,c_j} \quad with \; R_{w,c_j} > 1 \qquad (3.4)$$

where $p(R_{w,c_j})$ is the weight of chi-square statistic χ^2_{w,c_j} in the corpus in terms of R_{w,c_j}. It is defined as:

$$p\left(R_{w,c_j}\right) = \frac{R_{w,c_j}}{\sum_{j=1}^{k} R_{w,c_j}} \quad with \; R_{w,c_j} > 1 \qquad (3.5)$$

This new term-goodness measure, $r\chi^2(w)$, is the weighted sum of χ^2_{w,c_j} statistics when there is a positive dependency between the term χ^2_{w,c_j} and the category c_j, a bigger $r\chi^2(w)$ or $CHIR$ measure value indicates that the term is more relevant.

- *Semantic feature Measure*

To avoid any problems like the synonymy and polysemy and increase the semantic aspect of selected terms in a specified context, semantic features selection methods measure the importance of the term based on their semantic contents such as Pointwise Mutual Information, Information Gain or Log-likelihood ratio. One of the most used semantic measures is mutual information. We have the measure based on the Mutual Information (SIM) which is widely used in information theory. It was proposed as a measure of word association and reflects the strength of the relationship between words by comparing their actual co-occurrence probability with the probability that would be expected by chance.

Generally, Mutual Information represents the relative change in the probability of observing x when y is present (the amount of information that y provides about x) (Dagan et al., 1993). It is based on the fact that two words are considered similar if their mutual information with all the words in the vocabulary V is nearly the same (Dagan et al., 1993).

Dagan et al. (Dagan et al., 1993) have defined a semantic similarity measure between two terms w_1 and w_2 as follows:

$$sim(w_1, w_2) = \frac{1}{2|V|}\sum_{i=1}^{|V|}\left(\frac{min\left(I(z_i,w_1),I(z_i,w_2)\right)}{max\left(I(z_i,w_1),I(z_i,w_2)\right)} + \frac{min\left(I(w_1,z_i),I(w_2,z_i)\right)}{max\left(I(w_1,z_i),I(w_2,z_i)\right)}\right) \qquad (3.6)$$

where V is the vocabulary and $I(z_i, w_1)$ is the mutual information between the term z_i and w_1. $I(z_i, w_1)$ is evaluated using the following formula:

$$I(z_i, w_1) = P_d(z_i, w_1) log(\frac{P_d(z_i, w_1)}{P(z_i)P(w_1)}) \qquad (3.7)$$

where d represent the size of a sliding window, $P_d(z_i, w_1)$ is the probability of succession of z_i and w_1 in a window of $(d+1)$ words and $P(z_i)$ is the priori probability of the term z_i. This probability can be estimated by the ratio of the number of times that z_i is followed by w within the window and by the cardinal of the vocabulary.

The similarity between a term *w* and a document centroïd d is defined in (Sotoca and Pla, 2010) as the average of the similarities between the term w and the x terms of the document centroïd. This measure is given by:

$$SIM(w, d) = \frac{\sum_{j=1}^{x} sim(w, w_j)}{\sum_{j=1}^{x} \sum_{i=1}^{x} sim(w_j, w_i)} \qquad (3.8)$$

so as to determine the semantic relevance of a term *w* in a corpus of k clusters, for each cluster we calculate the weighted sum of its similarities with the document centroid $dcen_j$ of each cluster c_j using the following formula:

$$SIM(w) = \sum_{j=1}^{k} P\left(I\left(w, dcen_j\right)\right) SIM(w, dcen_j) \qquad (3.9)$$

where $P\left(I\left(w, cen_j\right)\right)$ is the weight of the similarity between the term *w* and the document centroïd $dcen_j$ and $I\left(w, dcen_j\right)$ is the mutual information between *w* and $dcen_j$. Considering the contingency table of a term *w* and a centroïd d where *A* is the number of times *w* and d co-occur i.e., *w* occur in documents that belong to the cluster whose centroïd is d, *B* is the number of times *w* occurs without *d*, *C* is the number of times *d* occurs without *w* and *N* is the total number of documents. The mutual information criterion between a term *w* and *a* document $dcen_j$ is defined by:

$$I(w, dcen_j) = P(w, dcen_j) \, log(\frac{P(w, dcen_j)}{P(w)P(dcen_j)}) \qquad (3.10)$$

If there is a strong association between w and $dcen_j$ then the joint probability $P(w, dcen_j)$ will be larger than $P(w)P(dcen_j)$; consequently $I\left(w, cen_j\right) > 0$. If w and $dcen_j$ are in complementary distribution, then $P(w, dcen_j)$ will be less than $P(w)P(dcen_j)$ hence $I(w, dcen_j) < 0$. In the case of poor association between *w* and $dcen_j$, then $P(w, dcen_j) \approx$ P(w) P ($dcen_j$), consequently $I(w, dcen_j) \approx 0$. The weight of $P\left(I\left(w, dcen_j\right)\right)$ defined as:

$$P\left(I\left(w, dcen_j\right)\right) = \frac{I(w, dcen_j)}{\sum_{i=1}^{k}\left(I(w, dcen_j)\right)} \; with \; I\left(w, dcen_j\right) > 0 \qquad (3.11)$$

The high weight of the $SIM(w)$ or SIM measure of a term indicates that this term is relevant semantically.

- *Structural feature measure*

Position features are related to the position of the given the word or phrase in the text. The word position in a document plays a significant role in linguistics and statistics approaches. Using a word's position as a feature has many advantages in selecting and increasing the precision and recall of keywords or keyphrase, especially in newspaper articles and Scientific papers (Witten et al., 1999) (Chen et al., 2005) (Rafiei-Asl and Nickabadi, 2017). For example, in a corpus of scientific research papers, each article comprises a consistent sequential structure (title, abstract, keywords, full text, etc.) that could not have the same importance level. Thus, the relevance of a given word varies from a part to another. In such valuable position parts of a document, words carry more important information than ordinary paragraphs. Statistics show that the title, the abstract, and the first paragraph usually highlight the topic and cover the whole text's core ideas. To calculate the importance and relevance of a term position along with its frequency in each part of a document, we propose a measure called the Term Position Weight 'TPW' (for single document and multi-documents). To do that, let's d be a document composed of n parts p_x and z terms t_y. Since we evaluate our method on a dataset of scientific articles, our case's considered parts are: the title, the abstract, and the full text.

- The document part weight:

We give for each part p_x of a document d a weight w_x, this weight reflects the relevancy of a part in the document with $\sum_{x=1}^{n} W_x = 1$. Notice that title and abstract have the larger weights.

- The term relevancy within a part of a document:

The relevancy of each term t_y from d in a part p_x is defined as the $tpw_{px}(t_y)$ and calculated as follows:

$$tpw_{p_x}(t_y) = tf_{p_x}(t_y) * w_x \qquad (3.12)$$

Where $tf_{p_x}(t_y)$ is the frequency of the term t_y in the part p_x.

- The term relevancy in a document:

Given a term t_y, its relevancy in the document d is defined as $tpw_d(t_y)$ calculated as follows:

$$tpw_d(t_y) = \sum_{i=1}^{n} tpw_{p_x}(t_y) \qquad (3.13)$$

- The term relevancy in the whole dataset:

The following equation is used to calculate the weight of a term t_y in the dataset:

$$tpw(t_y) = \frac{\sum_{y=1}^{n} tpw_d(t_y)}{n} \qquad (3.14)$$

Where, n is the number of occurrences of t_y in different documents of the dataset.
The high weight of the $tpw(t_y)$ measure of a term indicates that this term is relevant structurally.

3.3.3. Hybrid Keyword Scoring

After computing the score of each feature, we determine in this step the final keyword score. To compute this score, we gather all the feature weights into a unique $s(w)$ score. The gathering process goes through our novel hybrid term weighting model 'KW-HFSM' (The Keyword Hybrid Feature Selection Model), which can reduce the feature space by selecting terms that are statistically, semantically and structurally pertinent. We define the score of a term w as a weighted sum of its statistical measure $chir(w)$, semantic measure $sim(w)$, and statistic/structural measure $tpw(t)$.

The global measure of a term goodness $s(w)$ is defined as follows:

$$s(w) = \alpha * tpw(t_y) + \beta * chir(w) + (1 - \alpha - \beta) * sim(w) \qquad (3.15)$$

where α and β are weighting parameters between 0 and 1 and $\alpha + \beta \leq 1$. To select the most p pertinent terms, we conduct the three following steps: (1) calculate the hybrid measure $s(w)$ for each term in the dataset, (2) sort the terms in descending order of their scores, and (3) finally select the top p terms from the sorted list. A threshold δ is set to filter terms with a low $s(w)$ value. In our system, the value of δ is set to 0.25. As we can see in the next section, this is an important step in our process because single term scores are used to calculate scores of candidate keyphrases.

3.3.4. Hybrid Keyphrase Scoring

After identifying candidate keyphrases (in the candidate keyphrases identification stage) and calculating the features of single terms, in this subsection, we will describe how we use the feature values to rank all the candidate phrases. In this step, some authors choose a separate linear weighting coefficient perused feature (Kumar et al., 2008). Other authors use the supervised machine learning approach as a solution (Frank et al., 1999) (Hulth, 2003) (Haddoud and Abdeddaim, 2014). Although this method is useful, it involves additional steps of preparing the training set and the learning process. Our approach based on statistical features in conjunction

with the semantic and structural features is used to rank the candidate phrases. Similar schemes have been shown to yield performances comparable to the supervised approach (El-Beltagy and Rafea, 2009) (You et al., 2013). According to the type of the feature, a higher CHIR value indicates higher statistical importance. This also applies to the SIM feature and the TPW feature.

Keyphrases are built from keywords, so the higher the weight of the keywords that build a keyphrase is, the higher is the weight of a keyphrase. The keyphrase Hybrid Feature Selection weighting model (KPHFSM) define the weight of a keyphrase as:

$$s(kp) = s(w_1, w_2, \dots, w_n) = \frac{\sum_{i=1}^{n} s(w_i)}{n} \qquad (3.16)$$

where n is the number of words within a candidate keyphrase, consequently, the k-top keyphrases are ranked by their $s(kp)$ score. The final output of our system is a sorted list of non-overlapping K-top keyphrases.

3.4. Experimental Results and Evaluation

In this section, we discuss our experimental setup and evaluation results of the proposed keyphrase extraction method. The experiment aims first to explore how key parameters of features can influence HAKE's performance, then, to make the result more convincing, we compare the result with some existing representative methods.

3.4.1. Datasets

The primary corpus employed for evaluating the proposed technique and other similar techniques is the well-known "**SemEval-2010"** dataset published by the ACL SemEval workshop (Kim et al., 2010). This dataset contains 244 scientific conference and workshop articles collected from ACM Digital Library, including four research areas: (1) Distributed Systems, (2) Distributed Artificial Intelligence-Multiagent Systems, (3) Social and Behavioral Sciences-Economics, and (4) Information Search and Retrieval. The 244 articles are divided into three sets: 144 for the training set, 40 for the trial set (trial documents), and 100 for the test set. The average length of an article is between six and eight pages (see Table 3.2). For the collected papers, three sets of answers (gold standard keyphrases) were provided for each paper: author-assigned keyphrases, reader-assigned keyphrases, and finally, a set of a combination between the two previous sets. All reader-assigned keyphrases were extracted from the papers, whereas some of the original author-assigned keyphrases do not occur in the document's content (Kim et al., 2010). The evaluation runs on the combined set, which was the one whose final results were taken into consideration in the competition organizers. Since our system is unsupervised, experiments will only run the test set.

Table 3.2 Detailed statistics about evaluated datasets

Dataset	Number of Documents	#Gold Keys (per document)	Absent GoldKeys	Average document size (in word's number)
SemEval–2010 (Kim et al., 2010)	243	4002 (16.47)	11.3%	8332
Nguyen2007 (Nguyen and Kan, 2007)	209	2369 (11.33)	17.8%	5201
Theses100 (University of Waikato NZ, 2019)	100	767 (7.67)	47.6%	4728
Wiki20 (Medelyan et al., 2008)	20	730 (36.50)	51.8%	6177
Krapivin2009 (Medelyan et al., 2009)	2304	14599 (6.34)	15.3%	8040

The main benefit of using the SemEval-2010 corpus is that our results can be compared to those of the 25 AKPE systems, including those that participated in the challenge, and other recent works (such as TSAKE (Rafiei-Asl and Nickabadi, 2017), and KG-KE Model (Chen et al., 2019), TeKET (Rabby et al., 2020), YAKE! (Campos et al., 2020)). Notice that in the SemEval-2010 dataset, the maximum recall achieved over the combined keyphrase set was approximately 75 % (Kim et al., 2013).

The Nguyen2007 (Nguyen and Kan, 2007) is a dataset composed of 211 scientific conference papers with 5,201 tokens per document on average. The original author's assigned keyphrases were hidden to avoid bias. Standard gold keyphrases were manually assigned by student volunteers, which result in 11.33 gold keyphrases per document.

Theses100 (University of Waikato NZ, 2019) is a recent benchmark dataset. This dataset is composed of 100 master and Ph.D. theses from the University of Waikato, New Zeland. The dataset includes different domains of the theses, ranging from computer science, and economics to psychology, philosophy, history, chemistry, and others. All documents are in plain text, and the average length of these documents is approximately 7000 words. For comparison, standard gold keyphrases, along with the dataset, have been taken into account. Note that the theses100 dataset has 47.6% absent keys (as stated in the Table 3.2); for this reason, we did not expect to obtain high scores when testing on this particular dataset.

wiki20 (Medelyan et al., 2008) dataset contains 20 English technical research reports covering different subjects of computer science. Fifteen teams have assigned keyphrases to each report using Wikipedia article titles as the candidate vocabulary. Each team assigned 5.7 keywords on average, and each document has 35.5 gold keyphrases on average. Similarly, to 'Theses100' dataset, we did not expect to obtain high scores when testing on this particular dataset due to absent keys (51.8%).

Finally, the largest dataset '**The Krapivin2009**' (Krapivin et al., 2009) in terms of documents published by ACM in the period 2003–2005. It compromises 2304 full papers from the computer science domain, and 8040 tokens per paper, on average. The papers were downloaded from CiteSeerX Autonomous Digital Library, and each one has keyphrases assigned by the authors and verified by the reviewers, resulting in 6.34 gold keywords per document.

3.4.2. Baseline Methods

We compare our approach against some unsupervised and supervised state-of-the-art algorithms. An open-source Python Keyphrase Extraction (PKE) toolkit (Boudin, 2016) is utilized. The unsupervised systems compromise statistical methods such as the TF.IDF (Jones, 1972), KP-Miner (El-Beltagy and Rafea, 2009), RAKE

(Rose et al., 2010), YAKE! (Campos et al., 2020), and graph-based methods such as TextRank (Mihalcea and Tarau, 2004), SingleRank (Wan and Xiao, 2008), ExpandRank (Wan and Xiao, 2008), TopicalPageRank (Sterckx et al., 2015), PositionRank (Florescu and Caragea, 2017), TopicRank (Bougouin et al., 2013), and MultipartiteRank Boudin (2018). For the graph-based methods, we employed the default parameters as set in the corresponding papers. Namely, we set the co-occurrence window size for TextRank to 2 and ExpandRank to 10, as these values have yielded the best results for their evaluation datasets. For ExpandRank, we define the five nearest document neighbors for each processed document. A co-occurrence window size set to 10 for SingleRank, PositionRank, and TopicalPageRank, and a minimum similarity for clustering set to 0.74 with the linkage method for both TopicRank and MultipartiteRank. For statistical methods, in RAKE, no parameters were needed or defined. For YAKE! the maximum keyword length set to 3, the minimum keyword length set to 1, and the minimum number of keyword occurrences to 2. A full description of the parameters can be found in the corresponding papers. We compared our approach against the supervised KEA [www.nzdl.org/Kea/] (Witten et al., 1999) system, which is trained through 5-fold cross-validation using each corresponding dataset's documents. More comparisons were made against other supervised and unsupervised approaches, but only on the "SemEval- 2010" dataset since their implementation was not accessible. Though, the findings in their papers on that dataset were available.

3.4.3. Evaluation Metrics

Traditionally, automatic keyphrase extraction systems have been assessed using the proportion of *top-N* candidates that exactly match the gold standard keyphrases (Frank et al., 1999) (Turney, 2000). For example, in the SemEval-2010/Task-5 dataset (organized by SemEval workshop), there was a competition for extracting keyphrases from scientific articles. The participants were asked to extract the top 5-10-15 keyphrases for each article in the test document set. The evaluation was carried out by matching the extracted keyphrase sets in opposition to the answer sets (the combined set). We followed the same procedure given by the challenge SemEval-2010 (Kim et al., 2010) to evaluate our system against other systems, where an exact matching is considered if the stem of the automatically extracted keyphrases matches exactly the stem of the answer sets (the combined set). For the stemming process, all keyphrases are stemmed using the English Porter stemmer (Porter, 1980). The exact matching is strict and enables us to compare our results with previous systems.

Specifically, we used the following evaluation metrics:

The precision, which represents the number of the extracted keyphrases that match the manually assigned ones is given by equation (3.17):

$$Precision = \frac{TP}{TP+FP} \qquad (3.17)$$

The recall, which is the number of the keyphrases manually assigned that are extracted by the keyphrase extraction system, is given by equation (3.18):

$$Recall = \frac{TP}{TP+FN} \qquad (3.18)$$

where TP is the True Positive, FP is the False Positive, and FN is the False Negative.

The *F1-score*, is defined as:

$$F1 - Score = 2 * \frac{precision*recall}{precision+recall} \quad (3.19)$$

Given the precision and recall results of each document, we calculate then the micro-averaged of precision, of recall and of F-score of all documents.

In Micro-average method, we sum up the individual true positives, false positives, and false negatives of our system for different sets. By using this method, the precision and recall are defined as follows:

$$\text{Micro-average of precision} = \frac{\sum_{i=1}^{n} Tp_i}{\sum_{i=1}^{n} Tp_i + \sum_{i=1}^{n} Fp_i} \quad (3.20)$$

$$\text{Micro average of recall} = \frac{\sum_{i=1}^{n} Tp_i}{\sum_{i=1}^{n} Tp_i + \sum_{i=1}^{n} Fn_i} \quad (3.21)$$

The Micro-Average F1-Score will be simply the harmonic mean of (3.20) and (3.21).

3.5. Results and Discussion

In this section, we present the results of our experiments comparing HAKE against other methods. We divided our experiments into three different parts: (1) parameter value selection to determine the best parameter values; (2) linguistic filter comparison to show its effectiveness against other existing filters, and (3) overall effectiveness versus baseline and other methods, and finally discuss the overall findings.

3.5.1. Parameter Values Selection and Analysis

In this subsection, we include and analyze the obtained results from the experiments while selecting and changing the parameter values. We start by selecting suitable parameter values that directly influence the performance of the proposed method. The parameter selection is related directly to the importance of each feature. For this purpose, we conduct different analyses to understand the characteristics of the features CHIR, SIM and TPW, and how they behave and contribute to the final keywords weight S(w) when combined in the single keyword weighting model (KWHFSM). Moreover, we want to know their direct influence on the proposed method's performance over different evaluated datasets. Therefore, to estimate the importance of each feature regarding its contribution to the weights of the keywords S(w) in the KWHFSM weighting model, we conducted an ablation/relax study using a backward-like total exclusion/relaxation strategy. In this strategy, each feature is individually removed, strengthened, or relaxed in the weighting model. The best configuration was then selected and improved by varying parameter values.

First, we conduct experiments with different values of weighting parameters α and β on the SemEval-2010" dataset (evaluated on the gold combined set) and characterized by its large texts. Concretely, we set the weighting parameters $\alpha \; and \; \beta \in \{0,\ 0.2,\ 0.5,\ 1\}$. These values were applied to the KW-HFSM weighting model. A decrease in accuracy indicates how much the removed features impact the overall accuracy (see Figure 3.5). The number of keyphrases proposed by HAKE is set at @5, @10, and @15 for this first experiment (following the challenge evaluation procedure). The acquired results are demonstrated in Table 3.3. The highest performance shown for F1 value is 36,49% for top-15 keyphrases by (α *=0.2*) and (β *=0.2*) (that means including the *TPW*

feature with the *CHIR* and *SIM* measures), followed by 34.44% (α *=0.5*) and (β *=0.5*) (which means using just the *CHIR* and *SIM* measures), and 28.89% where (α *=0.5*) and (β *=0.5*) (which means disregarding the semantic measure). Whereas the lowest performance showed is 10.49% for the top 15 keyphrases where (α *=1*) and (β *=0*) (which means eliminating the statistic and the semantic measures). From the results, it is evident that with increasing the SIM measure value, F1 value increases.

To understand and get more insights into the usefulness and importance of each used feature of HAKE, we conduct the same experiments on the other datasets. Such as large texts dataset (Krapivin2009, and Wiki20), and on a dataset with different domains (Nguyen 2007) (See Figure 3.6). In Figure 3.6, we showed the effectiveness (F@10, P@10, R@10 scores) of HAKE while changing the weighting parameter values**.** As shown in this Figure, we observe that the measures CHIR, and SIM present competitive results compared to their elimination as the rates of F1, Precision, and Recall decrease.

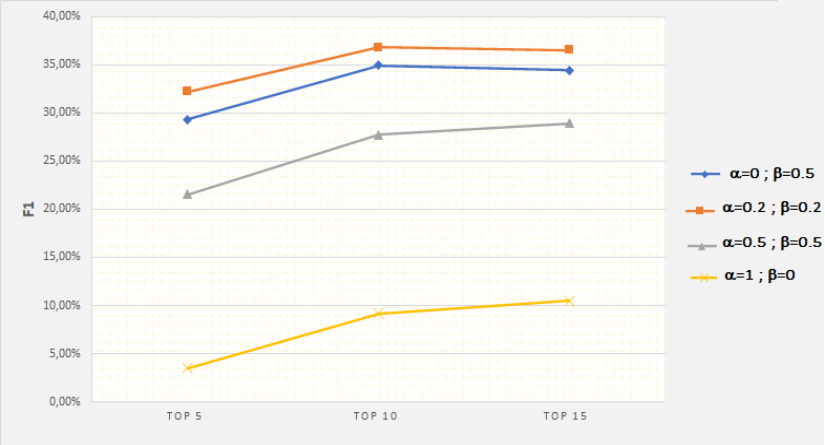


Figure. 3.5 The F1 scores of the top 5,10 and 15 keyphrases while varying the weighting parameters on the SemEval2010 Dataset.

Table 3.3 The performance of HAKE system while changing the weighting parameters

Parameters Values	Top 5 Keyphrases			Top 10 Keyphrases			Top 15 Keyphrases		
	P	R	FI	P	R	FI	P	R	FI
α =0; β =0.5; I- α - β =0.5	48,10%	21,10%	29,33%	40,10%	30,90%	34.90%	34,20%	34,70%	34,44%
α =0.2; β =0.2; I- α - β =0.6	**50,80%**	**23,60%**	**32,23%**	**43,90%**	**32,90%**	**36,78%**	**36,2%**	**36,8%**	**36,49%**
α =0.5; β =0.5; I- α - β =0	42,90%	14,40%	21,56%	35,80%	22,60%	27.70%	29,10%	28,70%	28,89%
α =I; β =0; I- α - β =0	24,60%	1,90%	3,52%	18,30%	6,10%	9,15%	10,40%	10,60%	10,49%

Indeed, it is apparent that the removal of the semantic feature ((α =0.5) and (β =0.5)) negatively impact the results in all the bassline datasets. This provides strong proof of how the semantic measure is an efficient way to boost the accuracy of the automated extraction of keywords and sustain a higher accuracy, precision, and recall rates, as keywords are more related to the document's topic despite its disparity in content domains.

Nevertheless, a remarkable performance drop can be seen when discarding the CHIR feature, and the SIM feature and using just the TPW ((α =1; β =0)). By disregarding the CHIR, we observe that this statistical feature tends to eliminate keywords with high relevance in the text. Thus, removing these keywords spoils the similarity computation and consequently, the quality of the extracted keywords. Another feature that enhanced the relevancy of a term is its relative position TPW. In this selected feature, we favorized words that appear at the beginning of the document (title, abstract) while considering other positions in the document (introduction, middle text, conclusion, etc.).

This is particularly evident for scientific texts that tend to concentrate a high degree of important keywords at the top of the text. It is apparent from Figure 3.6 that the F1 score shows a noticeable increase trend when taking into account this feature. For example, in the SemEval2010 dataset, an increase in the F1 score of 5.95 % is reported in the top 15 keyphrases (compared to the second-best score of the top 15 keyphrases where TPW is excluded). Likewise, in the Nguyen2007, wiki20, krapivin2009 datasets, an increase of F1 score 10.46%, 31.13%, and 17.58% is reported successively. Whereas the best score of the top 15 keyphrases without using the TPW. The Feature on the same dataset is 31.4%.

Analogously, the same point explains that the combination of features ((α =0,2; β =0,2)) while favouriting the semantic feature generates good keyword sets and improves the accuracy rate especially for a reduced number of keywords. Eccentricity, the combination of feature sets, presents a similar behavior for most of the datasets. Overall, it can be concluded that CHIR and SIM are the most important features for extracting the relevant keywords, and certainly without ablating the TPW feature. Given the analysis results, the hybrid model improves the accuracy rates across all the evaluated datasets.

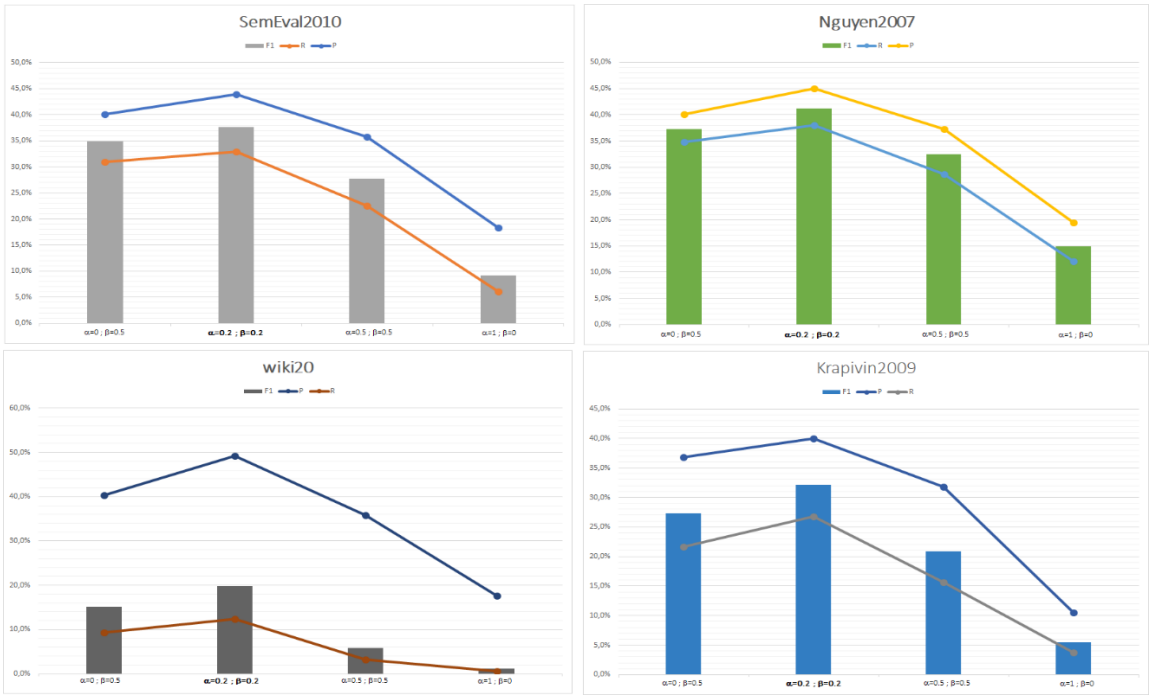


Figure. 3.6: The F1 scores of the top 5,10 and 15 keyphrases while varying the weighting parameters on the evaluated datasets

3.5.2. Linguistic Filter Comparison

To evaluate our system using different noun phrase filters, and to measure the contribution of our candidate keyphrases identification method to the overall system performance, we represent in Table 3.4 the results obtained by our method compared with six other most adequate filters evaluated on the SemEval2010 Dataset. The considered filters are the following:

- Filter 1: Our proposed POS tag definition of noun phrases corresponding to our POS tag sequence: (NN | NNS | NNP | NNPS | VBN | JJ | JJS | RB) * (NN | NNS | NNP | NNPS | VBG) +.
- Filter 2: (NN | NNS | NNP | NNPS | JJ | VBN | NN IN | NNS IN) * (NN | NNS | NNP | NNPS |
- VBG)
- Filter 3: (NN | NNS | NNP | NNPS | JJ) * (NN | NNS | NNP | NNPS | VBG) [71, 60]
- Filter 4: NBAR IN NBAR where NBAR = (NN | NNS | NNP | NNPS | JJ | JJR | JJS)* (NN | NNS
- | NNP | NNPS) [40, 39].
- Filter 5: (JJ)*(NN | NNS | NNP) + (Liu et al., 2009a).
- Filter 6: (NN | NNS | NNP | NNPS | JJ | JJR | JJS) * (NN | NNS | NNP | NNPS) (Nguyen and Kan, 2007).

Table 3.4 Performances of HAKE system with different noun phrase filters according to precision (P), recall (R) and F1-Score (F)

	Top 5 Keyphrases			Top 10 Keyphrases			Top 15 Keyphrases		
Filters	P	R	F	P	R	F	P	R	F
Filter 1	50,80%	23,60%	32,23%	42,20%	32,60%	36,78%	36,2%	36,8%	36,49%
Filter 2	44,80%	15,30%	22,80%	36,20%	24,70%	29,40%	28,30%	28,90%	28,60%
Filter 3	45,00%	15,40%	22,90%	33,50%	22,90%	27,20%	26,70%	27,30%	27,00%
Filter 4	42,20%	14.40%	21,50%	33,60%	22,90%	27,30%	26,30%	26,90%	26,60%
Filter 5	43,20%	14,70%	22,00%	32,70%	22,30%	26,50%	25,80%	26,40%	26,10%
Filter 6	37,80%	12,90%	19,2%	29,60%	20,20%	24,00%	23,80%	24,40%	24,10%

According to these experiments, we can see that our proposed filter (filter1) yield to the best results. For the top 5,10, and 15 on the SemEval2010 Dataset.

3.5.3. Overall Performance (Against Baselines)

In order to evaluate the efficiency of HAKE, we carried out experiments over five different datasets and compared HAKE's performance against state-of-the-art baselines (13 unsupervised and one supervised) and other methods (carried out over SemEval2010 Dataset Only). Table 3.5 lists the results of the top ten keyphrases (F1@10) (which is regularly used for this kind of evaluation (Campos et al., 2020)) for α =0.2; β =0.2 as a weighting parameter and n<=5 as a maximum number of words contained in a keyphrase. These parameters have achieved the best results in the parameter weighting selection experiments.

By looking at the baseline competing systems in Table 3.5, we notice that our method outperforms all the baseline systems (statistical and state-of-the-art graph-based) and ranks first over the top 10 keyphrases different datasets, text sizes, and domains. The best F1@10 score achieved by the HAKE system occurs with the Nguyen2007 dataset with 41,2%. In contrast, the best score of the best-performing system on the same dataset is 31.4%. Another remarkable point is that HAKE outperforms the supervised method KEA without any training on particular data. The low results occur with Theses100 and wiki20 datasets, which is expected because they have the most absent keywords. As can be seen from the Table 3.5 and Figure 3.7, the second-best result was obtained with KP-Miner in Nguyen2007, SemEval2010, Krapivin2009, and theses100 datasets, and YAKE! in wiki20 dataset which is statistical-based approaches. However, RAKE, which is considered one of the best-known state-of-the-art statistical approaches, appeared with deficient performance compared to all other methods. We conducted a more detailed analysis that allows us to conclude that HAKE outperforms the state-of-the-art methods when it is performed on large texts (as is the case of Krapivin2009, SemEval2010, and Wiki20 datasets), and on different domains (as is the case of Nguyen2007 dataset) (See Figure 3.8).

In Table 3.6, we tested our method following the challenge evaluation procedure of the SemEval201/Task 5 workshop. Each system's performance is given on the number of keyphrases: top 5, 10, and 15. The results (including their references) of all the participants are reported in (Kim et al., 2013) and ranked by F1-score. Hence, we compared our results with top-ranked methods (supervised and unsupervised) (Kim et al., 2013), including more recent ones who tested on the same dataset and followed the same evaluation procedure. By looking at the competing systems in Table 3.6, we notice that several ones adopt or partly adopt supervised learning approaches.

Few systems among the top-ranked ones, such as Core Word Expansion Algorithm (You et al., 2013), and KX FBK (Pianta et al., 2010) are based on the unsupervised approaches. As pointed out in the results, we notice that our method outperforms all the existing systems (supervised and unsupervised) and ranks first over the top 5-10-15 keyphrases inaccuracy. The F-score achieved by the HAKE on the assigned keyphrases (for the top 15 keyphrases) is 36.49%, whereas the F1-score of the best-performing system (which is also an unsupervised approach) on the same dataset is 35.1%. Hence For 15 keyphrases, HAKE improves at rate of 3.96% the F1-score compared to TSAKE (Rafiei-Asl and Nickabadi, 2017), and for 10 keyphrases, it increases by a rate of 25.1% compared to the F1-score of DPM-index (Haddoud and Abdeddaim, 2014) (which is a supervised approach). Notice that at the opposite of our system, some systems such as the HUMB system use different external knowledge features to improve their performances. These knowledge bases (HAL, GRISP, and GROBID/TEI) are specific to scientific papers.

In order to evaluate the efficiency of HAKE, we carried out experiments over five different datasets and compared HAKE's performance against state-of-the-art baselines (13 unsupervised and one supervised) and other methods (carried out over SemEval2010 Dataset Only). Table 3.5 lists the results of the top ten keyphrases (F1@10) (which is regularly used for this kind of evaluation (Campos et al., 2020)) for $\alpha = 0.2$; $\beta = 0.2$ as a weighting parameter and $n <= 5$ as a maximum number of words contained in a keyphrase.

These parameters have achieved the best results in the parameter weighting selection experiments. By looking at the baseline competing systems in Table 3.5, we notice that our method outperforms all the baseline systems (statistical and state-of-the-art graph-based) and ranks first over the top 10 keyphrases different datasets, text sizes, and domains. The best F1@10 score achieved by the HAKE system occurs with the Nguyen2007 dataset with 41,2%. In contrast, the best score of the best-performing system on the same dataset is 31.4%. Another remarkable point is that HAKE outperforms the supervised method KEA without any training on particular data.

The low results occur with Theses100 and wiki20 datasets, which is expected because they have the most absent keywords. As can be seen from the Table 3.5 and Figure 3.7, the second-best result was obtained with KP-Miner in Nguyen2007, SemEval2010, Krapivin2009, and theses100 datasets, and YAKE! in wiki20 dataset which is statistical-based approaches. However, RAKE, which is considered one of the best-known state-of-the-art statistical approaches, appeared with deficient performance compared to all other methods. We conducted a more detailed analysis that allows us to conclude that HAKE outperforms the state-of-the-art methods when it is performed on large texts (as is the case of Krapivin2009, SemEval2010, and Wiki20 datasets), and on different domains (as is the case of Nguyen2007 dataset) (see Figure 3.8).

In Table 3.6, we tested our method following the challenge evaluation procedure of the SemEval201/Task 5 workshop. Each system's performance is given on the number of keyphrases: top 5, 10, and 15. The results (including their references) of all the participants are reported in (Kim et al., 2013) and ranked by F1-score. Hence, we compared our results with top-ranked methods (supervised and unsupervised) (Kim et al., 2013), including more recent ones who tested on the same dataset and followed the same evaluation procedure. By looking at the competing systems in Table 3.6, we notice that several ones adopt or partly adopt supervised learning approaches. Few systems among the top-ranked ones, such as Core Word Expansion Algorithm (You et al., 2013), and KX FBK (Pianta et al., 2010) are based on the unsupervised approaches. As pointed out in the results, we notice that our method outperforms all the existing systems (supervised and unsupervised) and ranks first over the top 5-10-15 keyphrases inaccuracy.

The F-score achieved by the HAKE on the assigned keyphrases (for the top 15 keyphrases) is 36.49%, whereas the F1-score of the best-performing system (which is also an unsupervised approach) on the same dataset is 35.1%. Hence For 15 keyphrases, HAKE improves at rate of 3.96% the F1-score compared to TSAKE (Rafiei-Asl and Nickabadi, 2017), and for 10 keyphrases, it increases by a rate of 25.1% compared to the F1-score of DPM-index (Haddoud and Abdeddaim, 2014) (which is a supervised approach). Notice that at the opposite of our system, some systems such as the HUMB system use different external knowledge features to improve their performances. These knowledge bases (HAL, GRISP, and GROBID/TEI) are specific to scientific papers.

Table 3.5: The performance of HAKE using (α =0,2), and (β =0,2) and n<=5 compared with Baselines. F1@10

DATASET		Unsupervised Approaches											Supervised Approaches
		Statistical Based				Graph Based							Binary
	HAKE	TF.IDF	KP-MINER	RAKE	YAKE	Text Rank	Single Rank	Topic Rank	Topical PageRank	Position Rank	MultiPartite Rank	Expand Rank	KEA
Nguyen2007	**41,2%**	22,5%	31,4%	0,2%	25,6%	16,7%	15,8%	17,3%	14,8%	14,3%	19,0%	5,1%	22,1%
SemEval2010	**37,6%**	17,7%	26,1%	0,3%	21,1%	14,9%	12,9%	19,5%	12,5%	13,3%	19,9%	3,0%	21,5%
wiki20	**19,8%**	12,6%	15,6%	0,9%	16,2%	7,4%	3,8%	10,6%	5,9%	5,9%	9,1%	1,3%	13,4%
Krapivin2009	**32,0%**	15,0%	22,7%	0,2%	17,0%	12,1%	9,7%	13,8%	10,4%	10,4%	14,6%	2,7%	17,1%
theses100	**19,2%**	10,3%	15,8%	0,2%	11,1%	5,8%	6,0%	11,4%	8,3%	6,9%	11,7%	2,5%	10,4%

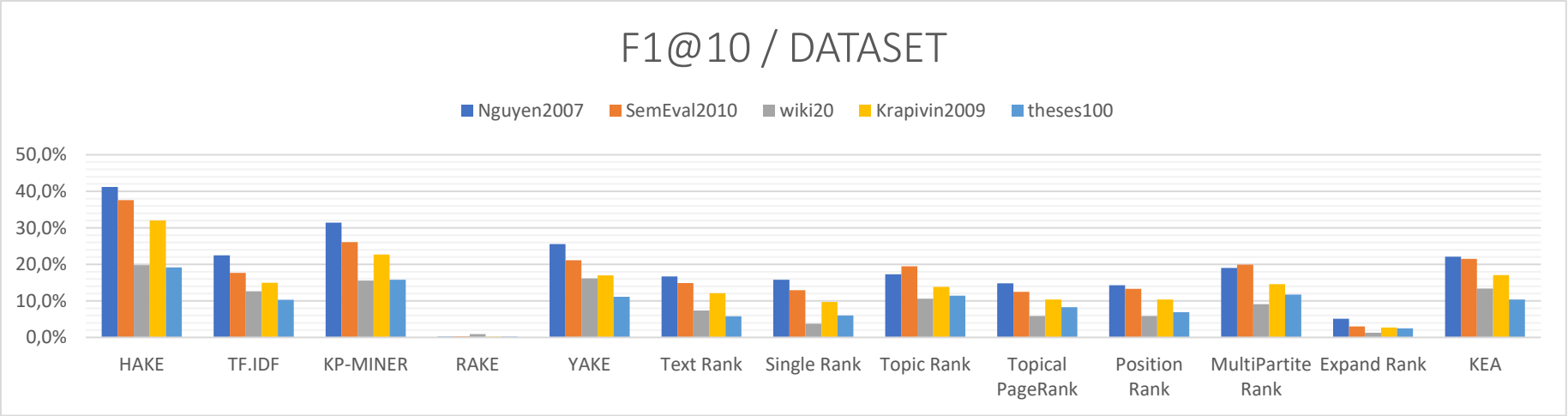


Figure. 3.7: Effectiveness of HAKE against Baselines. F1@10 SCORE

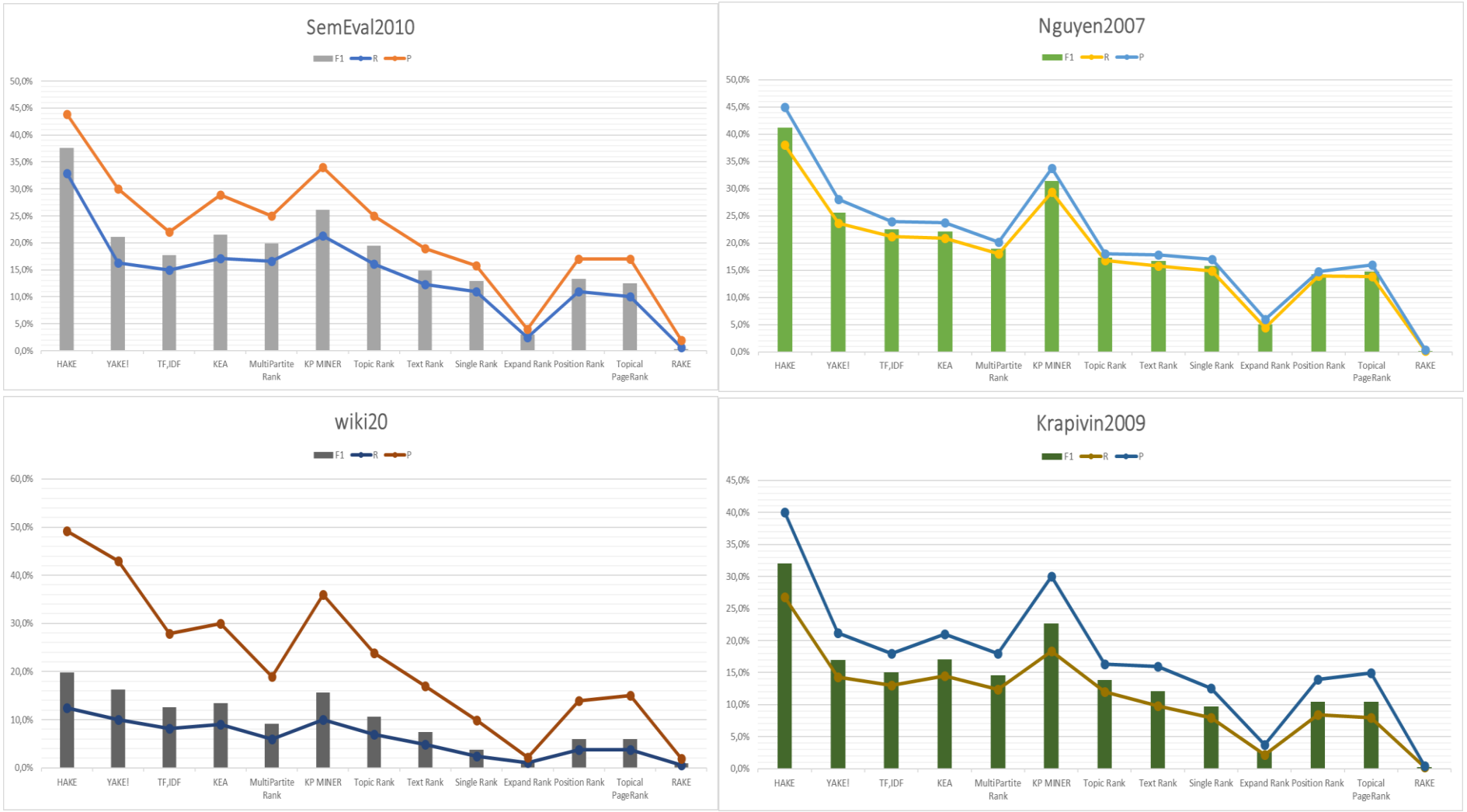


Figure. 3.8: Effectiveness of HAKE on top of large texts (Krapivin2009, SemEval2010, and Wiki20 datasets), and different domains (Nguyen2007 dataset). F@10, P@10, R@10

Table 3.6 The performance of HAKE using (α =0,2), and (β =0,2) and n<=5 compared with 23 other methods according to precision (P), recall (R) and F1-score (F) on the SemEval2010 Dataset

System	Top 5 Keyphrases			Top 10 Keyphrases			Top 15 Keyphrases		
	P	R	F	P	R	F	P	R	F
HAKE	**50,8%**	**23,6%**	**32,23%**	**43,90%**	**32,90%**	**37,60%**	**36,2%**	**36,8%**	**36,49%**
TSAKE (Rafiei-Asl and Nickabadi, 2017)	—	—	—	—	—	—	**35,1%**	**35,2%**	**35,1%**
DPM-index (Haddoud and Abdeddaim, 2014)	44,80%	15,30%	22,80%	36,20%	24,70%	29,40%	28,30%	28,90%	28,6%
HUMB (Lopez and Romary, 2010)	39,00%	13,30%	19,80%	32,00%	21,80%	26,00%	27,20%	27,8%	27,5%
Core Word Expansion Algorithm (You et al., 2013)	—	—	—	—	—	—	26,20%	26,8%	26,0%
WINGNUS	40,20%	13,70%	20,50%	30,50%	20,80%	24,70%	24,90%	25,5%	25,2%
KP-MINER	36,00%	12,30%	18,30%	28,60%	19,50%	23,20%	24,90%	25,5%	25,2%
SZTERGAK	34,20%	11,70%	17,40%	28,50%	19,40%	23,10%	24,80%	25,4%	25,1%
ICL	34,40%	11,70%	17,50%	29,20%	19,90%	23,70%	24,60%	25,2%	24,9%
SEERLAB	39,00%	13,30%	19,80%	29,70%	20,30%	24,10%	24,10%	24,6%	24,3%
KX-FBK	34,20%	11,70%	17,40%	27,00%	18,40%	21,90%	23,60%	24,2%	23,9%
DERIUNLP	27,40%	9,40%	13,90%	23,00%	15,70%	18,70%	22,00%	22,5%	22,3%
Maui	35,00%	11,90%	17,80%	25,20%	17,20%	20,40%	20,30%	20,8%	20,6%
DFKI	29,20%	10,00%	14,90%	23,30%	15,90%	15,90%	20,30%	20,7%	20,5%
DP-Seg (Jones, 1972)	33,30%	10,50%	16,00%	23,70%	16,90%	19,70%	19,20%	21,3%	20,2%
BUAP	13,60%	4,60%	6,90%	17,60%	12,00%	14,30%	19,00%	19,4%	19,2%
SJTULTLAB	30,20%	10,30%	15,40%	22,70%	15,50%	18,40%	18,40%	18,8%	18,6%
UNICE	27,40%	9,40%	13,90%	22,40%	15,30%	18,20%	18,30%	18,8%	18,5%
UNPMC	18,00%	6,10%	9,20%	19,00%	13,00%	15,40%	18,10%	18,6%	18,3%
U CSE	28,40%	9,70%	14,50%	21,50%	14,70%	17,40%	17,80%	18,2%	18,0%
LIKEY	29,20%	10,00%	14,90%	21,10%	14,40%	17,10%	16,30%	16,7%	16,5%
UvT	24,80%	8,50%	12,60%	18,60%	12,70%	15,10%	14,60%	14,9%	14,8%
POLYU	15,60%	5,30%	7,90%	14,60%	10,00%	11,80%	13,90%	14,2%	14,0%
UKP	9,40%	3,20%	4,80%	5,90%	4,00%	4,80%	5,30%	5,4%	5,3%

The main findings summarize that the generality of the parameters' features and robustness improve the keyphrases relevancy while calculating the weight of candidate phrases. We can consider these parameters as fixed parameters that do not require any training data or learning phase. On the one hand, the candidate selection method and the hybrid model that shape a well-chosen feature set (statistic, semantic, and structural) can perform in domain-independent and dependent corpus. On the other hand, the two parameters α and β are used to control the effect of the used features in the final result concerning parameters. Hence, we conclude that they are not sensitive to the test documents' domain, but they may be sensitive to their types. For example, when applied for web pages or unstructured documents (such as free text documents), We can expect that an appropriate increase of the value of α may reduce the effect of the structural feature (TPW), consequently decreasing the system accuracy.

To overcome this problem, we are thinking of suggesting models of parameter learning that applied experiments on different types of documents, leading to more corpus-adapted parameter settings and improving system performance. Our new algorithm for obtaining a refined candidate set improves applications that require high efficiency, for example, keyphrases for interactive query refinement. In such a real-time application, especially on a large searching scale (e.g., the whole web), a fast keyphrase extraction model is required, as time savings during this phase directly reduce the search response time. Finally, we prove that our unsupervised approach HAKE can achieve a good accuracy comparable to state-of-the-art and supervised methods without any human-labeled training sets or any external knowledge data.

3.6. Conclusion

In this chapter, we have presented our effective methodology of AKPE which is able to automatically extract relevant keyphrases for a given document. Our method, HAKE, is an unsupervised approach. It does not require any human-labeled training sets or any external knowledge data. This makes it more flexible, easy to use, and can be adapted across different domains. Moreover, our solution scales to any document length. And it can be integrated into any IR system or Text Mining Software. To be more efficient, we have introduced first a new method based on a parse tree approach, part of speech tagging, and filtering to cope with the difficulty of identifying potential candidate phrases. Then, during the step of feature selection, we have proposed a hybrid model that incorporates three types of features (structural, statistical and semantic) to select relevant individual keywords that are used later to rank relevant keyphrases. Our system deals with the lack of semantic information in the extraction process, by producing non-redundant, well-formed, and informative keyphrases. The obtained results are encouraging, they showed that our approach is effective in both: candidate keyphrases selection and relevant keyphrases extraction.

Additionally, we have provided evidence that scoring keyphrases according to their keyword's scores improves the keyphrase extraction performance, and considering the semantic relationships between features improve its accuracy. Since our AKPE method turned out to be an interesting side product of the work with extracting relevant information, condensing this extracted relevant information in the form of a summary would be very beneficial for several readers. Certainly, without summaries, it would be practically hard for human beings to get access to the ever-growing mass of information available online.

Hence, in the next chapter, we will present a hybrid Automatic Text Summarization (ATS) approach that is capable of generating the most common summaries tapes (extractive and abstractive), and it uses some of the proposed techniques in this AKPE method. In this way, keyphrase extraction can enrich the target summarized information.

4

EXABSUM: A hybrid Text Summarization System for Generating Extractive and Abstractive Summaries

4.1. Introduction

As seen previously in chapter 3, we have presented a novel AKPE, that helps users to extract relevant keyphrases from a specific document. The proposed method will enable fast and accurate search in large text collections, and the produced keyphrases will serve as condensed document summaries and can be used for various other applications. However, condensing the extracted important information in the form of a summary would be very beneficial for several readers. Certainly, without summaries, it would be practically hard for human beings to get access to the ever-growing mass of information available online. These challenges are addressed by proposing a new ATS approach that is capable of generating the most common summaries types: extractive and abstractive. This chapter is focused on reviewing the ATS task and handling the extractive, abstractive and hybrid ATS challenges and their underlying issues in particular.

4.2. The ATS Task: A Review

Without summaries, it would be practically impossible for human beings to get access to the ever-growing mass of information available online. Sometimes users may skip reading relevant information due to the large size of texts. Henceforth, condensing this important information in the form of a summary would be very beneficial for several users. Since the advent of text summarization in the 1950s, researchers have been trying to improve techniques in order to generate summaries matching with the human-made ones. Text summarization is one of the most difficult, though promising Natural Language Processing (NLP) tasks.

There are two fundamental questions in text summarization; i) how to select the relevant content of a document, and ii) how to express the selected content in a condensed manner while keeping the repetition to minimum. Hovy & Marcu (Hovy and Marcu, 2005) define a summary as a text which is produced from one or more texts, which contains a significant portion of the original text(s) information, and which is no longer than half of the original text(s). Mani & Maybury (Mani and Maybury, 1999), define the text summarization task as the process of finding the important contents in the original text and presenting them as a concise text in a predefined template. Four main objectives of a summary are considered by Huang et al. (Huang et al., 2010): coverage of information, information significance, redundancy in information, and cohesion in text.

According to these definitions, we can assume that a good summary should retain the original document's (or a cluster of documents) most important contents while being non-redundant and grammatically readable. Summaries can also be classified into indicative or informative depending on their intended purpose to alert or to inform respectively.

ATS systems can be classified into single-document or multi-document summarization systems. The former produces the summary from a single document while the latter generates the summary from a collection of documents. Text summarization approaches can be classified into: extractive, abstractive or hybrid ones. In one hand, the extractive approach selects the most salient (relevant) fragments (sentences or paragraphs) from the source document and concatenate them without any modification to the summary. On the other hand, the abstractive approach produces a new summary from an original one but not necessarily restricted to the text present in the original input document. The hybrid approach combines both the extractive and abstractive approaches. However, the extractive summarization suffers from the lack of coherence resulting summaries exhibit, partly due to the redundancy problem, the wrong link between sentences, and non-resolved coreference relationships. Abstractive summarization approaches still a growing and challenging field to the NLP research community. They imply the generation of new information, using sentence compression or fusion, lexical paraphrasing, or even Natural Language Generation NLG. Thus, new pieces of information can be added to generate an abstractive summary. Currently, the attention has shifted more towards the abstractive or the hybrid summarization (the fusion of extractive and abstractive techniques) (Gupta and Gupta, 2019).

The hybrid ATS is a promising research direction. It is recommended to propose hybrid TS systems in order to benefit from the advantages of both extractive and abstractive approaches (Nenkova et al., 2012) (Mahajani et al., 2019).

Since the advent of text summarization in the 1950s, there is a growing interest among the research community for developing new approaches to summarize text automatically. Researchers have been trying to improve ATS techniques to produce summaries that match with the human-made summary.

4.2.1. ATS Systems Classifications

In this subsection, we highlight the different classifications of ATS systems. As illustrated in Figure. 4.1, there are several classifications for ATS systems. These specifications are presented below.

- *The Input Size*: Single-document or multi-document. Single Document Summarization uses a single text document to generate a summary. In Multi-Document Summarization, the summary is generated based on a set of input documents.

- *The Nature of the Output Summary*: it is about how much information to be summarized. Generic or Query-Based. A generic text summarizer extracts important information from one or more input documents to provide a general meaning of its contents. While a query-based summary presents the information that is most relevant to the initial search query. It is sometimes referred to as a user-focused summary, like query-focused, or topic-focused (Mahajani et al., 2019).

- *The Summarization Algorithm*: Supervised, or Unsupervised, Deep Learning (Skip Thought Vectors)

- *The Summary Content*: Indicative or Informative. An indicative summary contains only information about the source text or the general idea (Bhat & al., 2018). It informs users about the topic of the input text to help them decide whether or not to read the original text. An informative summary covers all topics and main ideas of the text without details (Gupta et al., 2011).

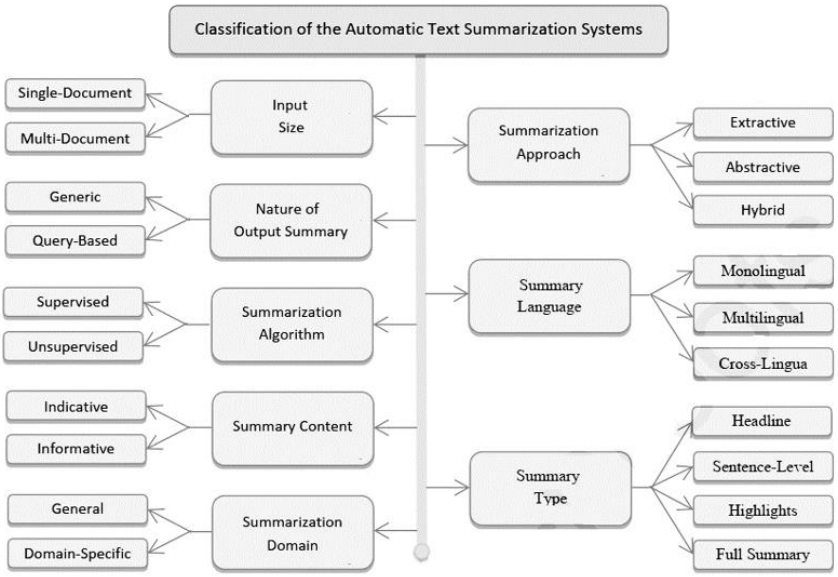


Figure. 4.1 Classifications of the ATS systems (El-Kassas et al.,2020)

- *The Summary Type*: Full Summary, Highlights, Headline, or Sentence-Level. The length of the generated summaries can differs based on the purpose of the ATS system. A full summary generation is usually guided by the required summary length or a compression ratio. The highlights summary provides the reader with a brief overview (highly compressed) about the main information in the input document(s). Headline generation produces a headline which is usually shorter than a sentence. And finally, a sentence-level summarization generates a single sentence from the input text which is usually an abstractive sentence (Gupta and Gupta, 2019).

- *The Summarization Domain*: Domain-Specific (to summarize documents from a certain domain), or General (to summarize documents that belong to different domains.

- *The Text Summarization Approach*: Extractive, Abstractive, or Hybrid. These approaches will be explained in detail in the next subsection.

4.2.2. ATS Systems Applications

ATS has shown that it is beneficial as a stand-alone application (Silveira and Branco, 2012) (Costa et al., 2013), as well as in combination with other systems, such as information retrieval (Cataldi et al., 2012), information extraction (Chen and Chundi, 2011), automatic rating prediction (Saggion et al., 2012), or text classification (Shen et al., 2007). TS has a wide range of applications. For instance, in the scientific context, ATS is often used to generate abstracts of research papers automatically, thus avoiding authors the difficult task of getting to synthesize in a paragraph the main subjects addressed of their

research. In particular, summaries are notably used by abstracting databases or abstracting services to inform the research community about advances in a specific kind of science. Another application of TS would be the automatic generation of newsletters, containing information of interest for a particular group of experts. In such domains, this would allow them to be updated on recent information.

Various ATS applications such as: News Summarization (Sahni and Palwe, 2017; Sethi et al., 2017); Opinion/Sentiment Summarization (Lovinger, et al., 2019; Roul & Arora, 2019); Microblog/Tweet Summarization (Chakraborty et al., 2019; Rudra et al., 2019; Kumar and Reddy, 2019); Books Summarization (Mihalcea & Ceylan, 2007); Biomedical Documents Summarization (Menéndez et al., 2014; Ghadiri and Davoodijam, 2018); Legal Documents Summarization (Anand and Wagh, 2019; Merchant and Pande, 2018); Scientific Papers Summarization (Lloret and Romá-Ferri, 2013; Jiang et al., 2019)

4.2.3. Automatic Summarization Approaches

ATS approaches are classified as extractive, abstractive, or hybrid. Each approach is implemented using a variety of method as shown in Figure 4.2. In this subsection, we provide a brief overview of each ATS approaches along with the methods of each approach in the literature.

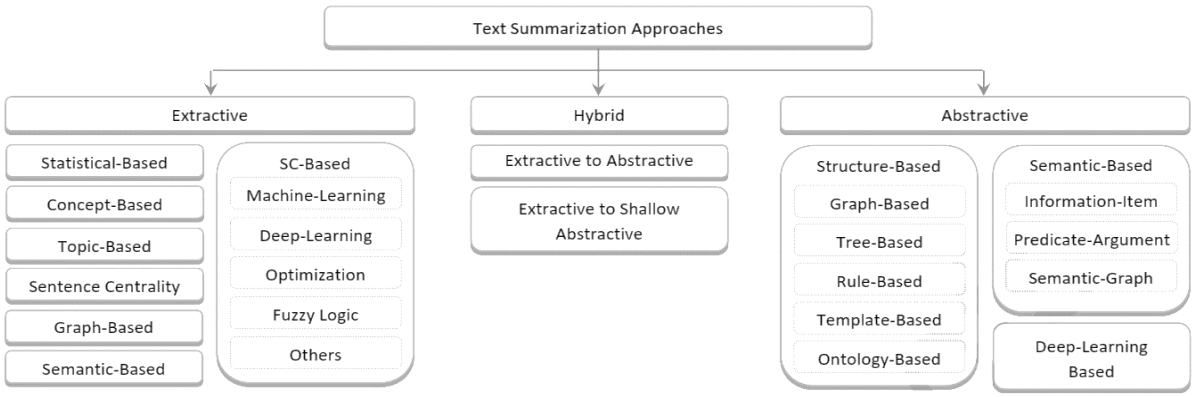


Figure. 4.2 ATS approaches and their associated methods (El-Kassas et al.,2020)

4.2.3.1. Extractive Summarization

The extractive TS system architecture consists of 1) pre-processing of the input text, 2) post-processing and 3) the processing tasks as follows:

- Creating a representation of the input text to ease the text analysis (e.g., graphs, N-gram, etc.) (Parveen and Strube, 2015). The Linguistic analysis is carried out to prepare the text for further processing.

The goal of Text pre-processing (TP) is to regularize the input files and set sentences boundaries for word sequences.

- Sentence scoring: ranking sentences based on the input text representation. It is a way to assign scores to the most relevant word(s) and then verifying and calculating sentence features such as their positions in the document, sentence length, title similarity. These features can be combined to score the remaining sentence relevancy using many aspects and select the highest scored ones for the summary. In most of the indicator representation methods, the score is computed by aggregating the evidence from different indicators. Machine learning techniques are often used to find indicator weights.

- Extraction and ranking of high-scored sentences: selecting the most important sentences from the input document(s) and concatenating them to create the summary (Nenkova et al., 2012; Ferreira et al. 2013). Then, an extractive summary is generated depending on a predefined length and preserving the same order of the generated sentences as the input text (Mahajani et al., 2019).

The extractive ATS methods can be categorized into seven categories (see Figure 4.2):

- Statistical-Based Methods: these methods extract important words and sentences from the input text based on the statistical analysis of a set of features. These methods are independent of any language, such that if the summarizer is developed using these methods, then it can summarize text in any language. Thus, these methods do not require any additional linguistic knowledge or complex linguistic processing (Li et al., 2008).

- Concept-Based Methods: these methods extract concepts from a piece of text from external knowledge bases like Wikipedia, HowNet, WordNet (*Miller, 1995; WordNet: An Electronic Lexical Database, 1998*), etc. The relevance of sentences is then calculated based on the concepts retrieved from these knowledge base instead of words (Moratanch & Chitrakala, 2017).

- Topic-Based Methods: these methods rely on identifying a document's topic which is its main subject (i.e., what the document is about). Common methods for topic representations include Term Frequency, Term Frequency-Inverse Document Frequency (TF-IDF), lexical chains, topic word approaches Moratanch & Chitrakala, 2017).

- Sentence Centrality or Clustering-Based Methods: in these methods, a multi-document extractive summarizer identifies the most central and important sentences in a cluster such that they cover the relevant information related to the cluster main subject.

- Graph-Based Methods: These methods use sentence-based graphs to represent a document or document cluster.

In a graph, text elements (words or sentences) are represented by nodes and edges connect the related text elements (semantically related) together. Sentence compression (Zajic et al., 2007), and sentence fusion (Barzilay & McKeown, 2005) are also quite used when in generating abstractive summaries. In particular, graph-based algorithms used for such purpose have been proven to be very successful (Filippova & Strube, 2008), (Filippova, 2010). Regarding sentence fusion, in (Filippova & Strube, 2008) related sentences are represented by means of dependency graphs, and then the nodes of such graphs are aligned taking into account their structure. Then, Integer Linear Pro- gramming (Schrijver, 1986) is used to generate a new sentence, where irrelevant edges of the graphs are removed, and an optimal sub-tree is found employing structural, syntactic and semantic constraints. For sentence compression, Filippova (2010) suggests a method based on word graphs, where the shortest path is computed to obtain a very short summary (only one sentence) from a set of related sentences. In (Liu & Liu, 2009), the authors attempt to transform an extractive summary into an abstractive one in the context of meeting summarization by performing sentence compression.

- Semantic-Based Methods: Latent Semantic Analysis (LSA) is a commonly used semantic-based extractive ATS method. LSA is an unsupervised technique that represents text semantics based on the observed co-occurrence of words (Merchant and Pande, 2018).

- Machine-Learning-Based Methods: machine learning based approaches learn from the data. They can be supervised, unsupervised or semi-supervised. The supervised systems convert the summarization problem to a supervised classification problem at the sentence level. The system learns by examples to classify each sentence of the test document either as “summary” or “non-summary” class using a training set of documents. In the other hand, unsupervised systems do not require any training data. They generate the summary by accessing only the target documents. They try to discover hidden structure in the unlabeled data.

- Deep-Learning-Based Methods: Deep Learning is a part of machine learning based methods and involves training and learning data. It involves multiple layers of non-linear processing to extract the features from the text. Learning can be both supervised or unsupervised (Gupta and Gupta, 2019).

- Optimization-Based Methods: these methods consider the summarization problem to an optimization problem. For example, a generic extractive multi-document ATS system is formulated as a multi-objective problem in (Sanchez-Gomez, et al., 2020).

- Fuzzy-Logic-Based Methods: in these methods, the fuzzy logic concept is used for ATS. Fuzzy logic resembles the human reasoning system and provides an efficient way to represent the feature values of sentences (Kumar & Sharma, 2019).

4.2.3.2. Abstractive Summarization

Abstractive text summarization produces an abstract summary that includes words and phrases different from the ones occurring in the source document. Therefore, it needs a deeper analysis of the input text and extensive natural language processing. Therefore, it is much more complex than extractive summarization. However, from the evaluation carried out in terms of coherence, the results showed that more than 78% of the abstracts were more coherent than extracts (Hasler, 2007; Gupta and Gupta, 2019).

The architecture of an abstractive text summarizer consists of the pre-processing, post-processing, and the processing tasks which include: 1) building an internal semantic representation, and 2) generating a summary using natural language generation techniques to create a summary that is closer to the human-generated summaries (Gupta and Gupta, 2019) (see Figure 4.3).

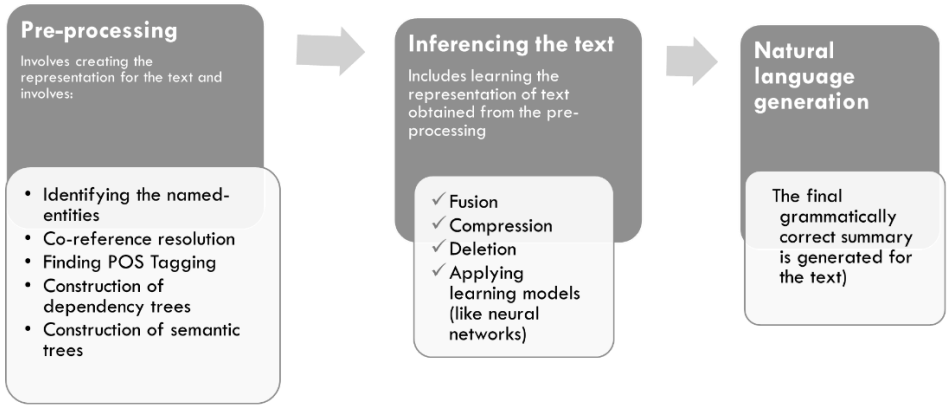


Figure. 4.3 The architecture of the abstractive text summarization system

The abstractive ATS methods can be categorized into three categories (Gupta and Gupta, 2019) (see Figure 4.2):

- Structure-based: the important information of the text is populated into a predefined structure to create the abstractive summary (find the most important information from the text and then use templates, rules, trees, ontology, etc. to create the abstractive summaries). Structure-based methods compromise several approaches such as: tree-based, template based, ontology based, Lead and Body phrase methods, graph-based methods, rule-based methods (Gupta and Gupta, 2019).

- Semantic-based: using the text semantic representation and the natural language generation systems (e.g., based on information items, predicate arguments, and semantic graphs). Semantic-based methods build the semantic representation of the input text by using the information-items, predicate arguments, or semantic-graphs then use a natural language generation system to produce the abstractive summaries. Semantic-based methods includes: Information Item Based (INIT) Methods, Predicate-Argument Based Approach, Multimodal Semantic Method, Multimodal Semantic Method (Gupta and Gupta, 2019).

- Deep-learning based methods provide another categorization to the abstractive methods as neural-based or classical which broadly refers to any method that is not neural-based. It combines Semantic and Structure methods, and involves multiple layers of non-linear processing to extract the features from the text (Gupta and Gupta, 2019).

- Discourse and Rhetorical Structure Based Approach: discourse structure helps capture the structure of text and thus help find the most important sentences of the text. The main aim is to capture the internal representation of the text and convert them into the relations of discourse (Gupta and Gupta, 2019).

➔ After using one or multiple of these methods, various techniques are employed to generate the final summary such as:

  - Sentence compression: reduce the size of sentence (remove irrelevant parts) while maintaining its meaning. Or/and
  - Concept fusion: via alignment of sentences (as input) to find common information, and then on the basis of alignment, fusion tree is created. Or
  - Paraphrasing: replace the phrases with shorter paraphrases help reduce the length of text. Or
  - calculation of path scores: representing the sentence in subject-object-verb, and find the pendency relations between sentences (dependency score to be calculated).
  - Summary Generation: revision for generating the final summary out of the selected sentences using: The Natural Language process NLG that increase fluency and decrease grammatical mistakes. It adds new vocabulary and languages structures to sentences.

In conclusion, the recent abstractive summarization efforts mainly focus to use the deep learning models especially in short text summarization (Kouris, Alexandridis, & Stafylopatis, 2019). The structure-based methods are usually used with the extractive methods for producing hybrid summaries and with the semantic-based or deep-learning-based methods to produce abstractive summaries (Gupta & Gupta, 2019).

4.2.3.3. Hybrid Summarization

The hybrid approach combines both the extractive and abstractive approaches. It makes benefit from the combination of the advantages of both extractive and abstractive techniques. The two approaches are complementary and the overall performance of summarization is improved (Gupta and Gupta, 2019, Merrouni et al., 2019).

A typical architecture of a hybrid text summarizer is shown in Figure 4.4. It commonly consists of the following phases (Bhat et al., 2018; Lloret et al., 2011):

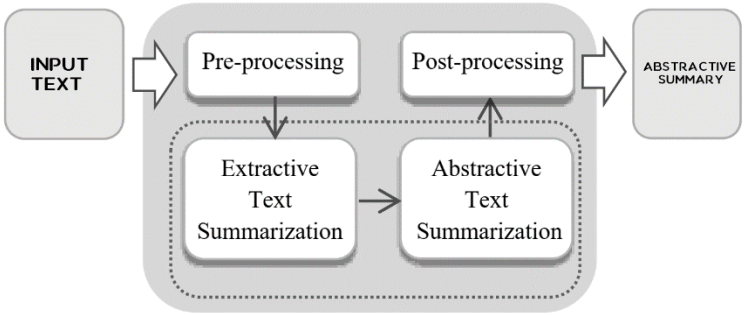


Figure. 4.4 An architecture example of a hybrid text summarization system

- Pre-Processing the input documents
- Relevant sentence extraction: extract the key sentences from the input text as an essential step of extractive summarization.
- Summary generation (abstractive ATS phase): apply abstractive methods and techniques on the extractive summary to generate the final abstractive summary.
- Post-Processing: to ensure that the generated sentences are valid, some general rules of filters need to be defined like such as: sentence length, sentence composition, etc.

There are mainly two methods that have been used in the hybrid text summarization (see Figure 4.3): extractive to shallow abstractive methods and extractive to abstractive.

- *Extractive to Abstractive Methods* use first the extractive ATS methods and then they use one of the abstractive text summarization methods which is applied to the extracted sentences. For example, a system can consists of two phases: 1) the extraction phase that uses features (statistical, semantic, etc.) features for scoring and extracting relevant sentences, and 2) the abstraction phase that use the deep learning techniques to generate the summaries.

- *Extractive to Shallow Abstractive Methods* start by employing one of the extractive ATS methods then they use a shallow abstractive text summarization method that applies one or

more of the following techniques to the extracted sentences: synonym replacement techniques (Patil et al., 2014), rephrasing (NAYEEM et al., 2018), information compression techniques, information fusion techniques (Lloret et al., 2013), etc.

The hybrid summarization approach is a very promising research direction. Mahajani et al. recommend researchers to propose hybrid ATS systems in order to benefit from the advantages of both extractive and abstractive approaches (Mahajani et al., 2019).

4.2.4. Available Corpora

One main problem for researchers is the availability of model summaries (or man-written summaries) while attempting to produce automatically summaries. These are used for measuring the quality of the content selected by automatic summarizers. Building such model summaries is not an easy task. Therefore, it leads to great benefits for the TS research community when specific corpora and datasets are available. For instance, this allows different approaches to be fairly compared. Here we present some available corpora for ATS evaluation.

Some of the best-known summarization corpora that are available and can be freely used for academic and research purposes are: the corpora used at DUC and TAC conferences. They deal mainly with newswire documents gathered from several press agencies. The model summaries provided are either extracts or abstracts written by humans. These model summaries vary in content, depending on the proposed task in each conference edition (e.g., single-document or multi-document summarization). In this case, instead of providing complete model summaries in TAC conferences, humans were asked to select fragments of information that were more relevant to the task, since summaries were evaluated using the Pyramid method (see the next Subsection).

The Multilingual summary evaluation data (Turchi et al., 2010) consist of a set of documents related to four topics (science-and-society, the-conflict-between- Israel-and-Palestina and genetic malaria). Each cluster contains the same 20 documents in seven languages (English, French, German, Arabic, Czech, Russian and Spanish). The relevant sentences of each document are manually annotated, and as a consequence, this dataset is very appropriate for evaluating single- or multi-document, as well as multi-lingual extractive summarization systems.

The CNN corpus (Lins et al., 2019) is a collection of news documents for single-document summarization based on the news articles (twelve subject categories) from the CNN website (http://www.cnn.com). It is the largest (3000 articles) test set available today for single document extractive summarization. An essential aspect of this dataset is the presence of a good quality abstractive summary for each document written by the original authors, called highlights, and extractive summaries (the gold standards) as well (available with approximately 90~100 words). It was also recently used in the DocEng'19 Competition on Extractive Text Summarization (Dueire Lins and Simske, 2020).

The CNN / Daily Mail dataset as processed by Nallapati et al. (2016) has been used for evaluating summarization. The dataset contains online news articles (781 tokens on average) paired with multi-sentence summaries (3.75 sentences or 56 tokens on average). The processed version contains 287,226 training pairs, 13,368 validation pairs and 11,490 test pairs.

4.2.5. Summary Evaluation

As is stated before, to generate a condensed and readable summary several objectives should be considered such as:

- Information Coverage: the summary should involve the relevant information of the input document(s).
- Information Significance: the summary should cover the various topics (main topics, or user-preferred topis) of the input document(s).
- Information Redundancy: minimize or eliminate the redundant information in the generated summary, including semantically redundant ones.
- Text Coherence: the summary should be readable and understandable text to the user.

Evaluation of summary is a very important task in the field of ATS. It enhances the development of reusable resources and infrastructure. It helps in comparing and replicating results and thus, adds competition to improve the results. However, it is practically impossible to manually evaluate multiple documents for obtaining an unbiased view. Information changes according to the purpose of the summary and to capture this information automatically, is a difficult task. Therefore, reliable automatic evaluation metrics are required for fast and consistent evaluation. Evaluation of summary is a challenging work too as it is not easy for humans to know what kind of information should be present in the summary.

There are two ways for measures the performance of text summarization (Gambhir and Gupta, 2017) (see Figure 4.5):

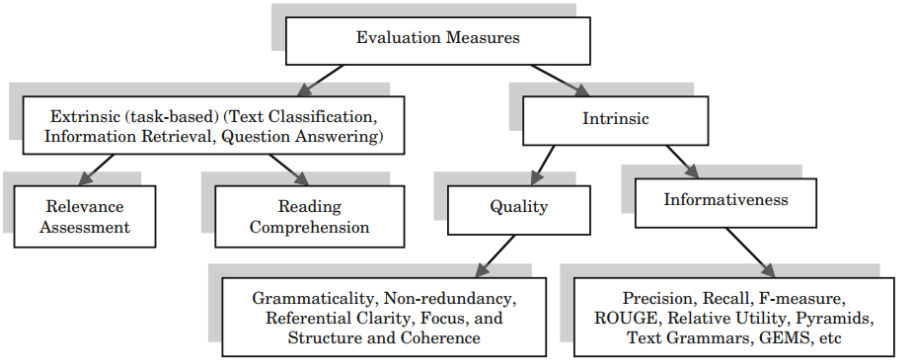


Figure. 4.5 Taxonomy of summary evaluation measures (Gambhir and Gupta, 2017)

1) Extrinsic methods: the extrinsic evaluation assesses the utility of summaries in a given application context (e.g., relevance assessment, reading comprehension, etc.) (Lloret et al., 2017). It measures summary quality through a task-based performance measure such as the information retrieval-oriented tasks (Text classification, Question answering, etc.).

   Various methods for extrinsic evaluation are (Gambhir and Gupta, 2017):

   – Reading comprehension: It determines whether it is able to answer multiple choice tests after reading the summary.
   – Relevance assessment: Here various methods are used for evaluating a topic's relevance present in the summary or the original document.

2) Intrinsic methods: it determines the summary quality on the basis of coverage between the human-made summary and the machine-made summary. Informativeness or quality or are the two important aspects on the basis of which a summary is evaluated. Typically, the informativeness of a summary is evaluated by comparing it with a human-made summary, i.e., reference summary. Another paradigm is usually used too, i.e., reliability to the source which checks whether the summary consists of the similar or the same content as present in the original text. There is a problem with this method, i.e., how to know which concepts in the document are relevant and which are not.

Some of the methods for ***informativeness evaluation*** are ROUGE (Lin 2004), Relative utility, Factoid score, Pyramid method, etc. ROUGE helps to automatically evaluate the summary. It counts the number of units common between a particular summary and a collection of reference summaries (Lin 2004).
Some of these methods are automatic, which do not require human annotations while others are semi-automatic, which require some sort of human annotations. Factoid score, Relative utility, Pyramid method and Text grammars are semiautomatic while the rest are automatic.

ROUGE includes five measures such as:

- ROUGE-N measures the *N-gram* units common between a particular summary and a collection of reference summaries. *N* determines the N-gram's length.
- ROUGE-L calculates Longest Common Subsequence (LCS) metric. LCS is the maximum size of common subsequence for two given sequences *X* and *Y*. ROUGE-L calculates ratio between size of two summaries' LCS and size of reference summary.
- ROUGE-W is the weighted longest common subsequence metric which is can be computed efficiently using dynamic programming. It is the improvement over the simple LCS approach. ROUGE-W prefers LCS with successive common units.
- ROUGE-S (Skip-Bigram co-occurrence statistics): in ROUGE-S, any word pair in the sentence order with random gaps is the skip bigrams. It evaluates the proportion of skip bigrams common between a particular summary and a collection of reference summaries.
- ROUGE-SU is the weighted average between ROUGE-1 and ROUGE-S. It extends ROUGE-S with counting unit as unigram. Actually, this is an improvement over ROUGE-S.

For intrinsically evaluating the summary, other popular metrics are precision, recall and f-measure. They are required to predict coverage between automatically generated machine-made summaries and human-made summary. With the benefit of above metrics, it is also feasible that two summaries generate different evaluation results even being identically good.

In the ***quality evaluation,*** the linguistic aspects of the summary are considered. In the conferences of DUC and TAC, five questions based on linguistic quality are employed for evaluating summaries like grammaticality, referential clarity, structure and Coherence, focus, and non-redundancy that do not need to be compared against the reference summary.

Expert human assessors evaluate the summary manually by assigning a score to the summary corresponding to five-point scale on the basis of its quality. Text quality of summary can also be evaluated by analyzing different factors for readability (Pitler and Nenkova, 2008).

Text quality is analyzed through different criterions like syntax, discourse, or vocabulary so that correlation can be estimated between these factors and already obtained human readability ratings.

4.3. Related Works

In this section, we highlight some related works and existing ATS systems that were applied to different application domains.

Two fundamental questions in text summarization are i) how to select the relevant content of a document, and ii) how to express the selected content in a condensed manner, and keep the repetition to a minimum. The extractive ATS techniques are usually performed in three steps (Nenkova et al., 2012): (1) Create an intermediate representation of the original text (usually, it preprocess and divides the text into paragraphs, sentences, and tokens); (2) Sentence scoring (the score should be a measure of how significant a sentence is to the "understanding" of the text as a whole), by assigning scores to the most relevant words, and then verifying sentences features such as their positions in the document, sentence length, title similarity, etc.; (3) Select high scored sentences (combines the score provided by the previous steps) and generates a summary. The first attempt to text summarization using sentence scoring dates back to 1958 (Luhn, 1958). Previous works on extractive summarization have focused on statistical, graph-based, and optimization-based methods. Statistical methods (SAGGION et al., 2013) use simple statistics such as TF-IDF scores at their core, word co-occurrence (Liu et al., 2009) (Tonelli and Pianta, 2011) (Gupta et al., 2011) to extract relevant sentences. Ko and Seo (Ko and Seo, 2008) proposed an effective method for summarizing text in which significant sentences are extracted by applying contextual information and statistical approaches. Graph-based approaches (BARRIOS et al., 2016) represent text as a network linking sentences and use graph-based ranking methods to generate a summary. Baralis et al. (Baralis et al., 2013) proposed GRAPHSUM, a novel and general-purpose summarizer based on a graph model, which represents correlations among multiple terms by discovering association rules. Parveen and Strube (Parveen and Strube, 2015) proposed an extractive, graph-based unsupervised technique in order to summarize single documents that consider three important properties of summarization, i.e., importance, non-redundancy, and local coherence. Optimization-based approaches (DURRETT et al., 2016) use techniques such as integer linear programming (Alguliev et al., 2011), constraint optimization (Lin and Bilmes, 2010), and sparse optimization (Yao et al., 2015).

Different ATS systems have been released such as MEAD system (Radev et al., 2004), a multilingual (English and Chinese) summarizer system produces extractive single and multi-document generic or query-focused summaries. For determining the salient sentences, it relies on calculating surface features, such as sentence length, the similarity with the first sentence, the similarity with a centroid, sentence position, and sentence length, which are then combined linearly. QCS system (DUNLAVY et al., 2007) integrates a summary approach into a broad process for retrieving and clustering information. A method coupling sentence "trimming" and a hidden Markov model, followed by a pivoted QR decomposition, is used to create a single extract summary for each cluster. The sentences with the highest probability in the model are chosen for the summary.

Other ATS systems, such as SummGraph (Plaza, 2011), used graph-based algorithms and knowledge databases to detect relevant documents' content. In particular, this system has been proven to work successfully in news, biomedical, and tourism domains.

Natural Language Generation (NLG) has also been used in summaries to incorporate new vocabulary and language structures. Belz (Belz, 2008) proposes a TS approach based on 'NLG' to automatically generate weather forecast reports. Mohammad et al. (Mohammad et al., 2009) presented a system that used citations to generate technical surveys automatically. Recently, Erera et al., 2019 (Shai et al., 2019) proposed IBM Science Summarizer, a novel system providing summaries for Computer Science publications. It produces summaries focused around an information need provided by the user - a natural language query, scientific tasks (e.g., "Machine Translation"), datasets, or academic venues.

While extractive methods may be well suited to identify the relevant information, these techniques may lack human-generated summaries' fluidity and accuracy. Consequently, *abstractive ATS* techniques try to improve the coherence among sentences by eliminating redundancies and clarifying the contest of sentences and may produce new ones to the summary. Most of the abstractive techniques include sentence compression, fusion, or revision to generate the final summary. The pioneering work of Barzilay and McKeown (Barzilay and McKeown, 2005), introduced a framework where dependency trees represent the input sentences, some words are aligned to merge the trees into a lattice. Then the lattice is linearized using tree traversal to produce fusion sentences. Filippova and Strube (Filippova and Strube, 2008) presented a novel sentence fusion method which formulates the fusion task as an optimization problem. This unsupervised method is based on dependency structure alignment, semantically, and syntactically informed phrase aggregation and pruning. Later, Filippova considered the task of summarizing a cluster of related sentences with a short sentence, which she calls it multi-sentence compression and presented a simple approach based on shortest paths in word graphs (FILIPPOVA, 2010). Her method was capable of generating grammatical and informative summaries, and it was adopted in several recent summarization systems (GUPTA, 2019). Boudin (Boudin and Morin, 2013) extended the approach of Filippova and considered the Multi-Sentence Compression (MSC) as the task of generating a short single sentence summary from a cluster of related sentences. He proposed an N-best reranking method based on the number and relevance of keyphrases that they contain. This method was able to create more informative summaries. Banerjee et al. (Siddhartha et al., 2015) used the word graphs and the Integer Linear Programming (ILP) to create the multi-document abstractive summaries. They created the clusters of similar sentences among the important documents. The shortest paths are obtained by generating word-graphs, and the ILP model is applied to find the maximum information and readability of the sentences. ILP helps reduce redundancy in summary.

Nayeem et al. (NAYEEM et al., 2018) developed an unsupervised abstractive summarization system. They designed a paraphrastic sentence fusion model that jointly performs sentence fusion and paraphrasing using a skip-gram word embedding model the sentence level. This model improves the information coverage and, at the same time, abstractiveness of the generated sentences. Shang et al.

(Shang et al., 2018) introduced a novel fully unsupervised graph-based framework for abstractive meeting speech summarization. Their unified framework combines the strengths of 6 recent approaches that have been applied to 3 different tasks (keyword extraction, multi-sentence compression, and summarization) while addressing their weaknesses. They processed the abstractive summarization in 4 steps: preprocessing, community detection, multi-sentence compression, and submodular maximization.

Most recently, the NLP research community attention has shifted more to Hybrid ATS approaches. Hybrid approaches exploit extractive methods to identify content words and sentences that should be part of the summary and guide the generation of abstracts (Chen et al., 2018). These approaches combine the strengths of extractive and abstractive ATS techniques.

Di Fabbrizio et al. (Giuseppe et al., 2014) presented a hybrid method to generate summaries of product and services reviews by combining natural language generation and salient sentence selection techniques. Their system 'STARLET-H' operates as a hybrid abstractive/extractive summarizer. To provide evidence for reveal, or justify positive or negative opinions, it uses extractive summarization techniques that select salient quotes from the input reviews and embeds them into an automatically generated abstractive summary. Notwithstanding, the system needs a large amount of training data to learn aspect order. Lloret and ROMÁ-FERRI (LLORET and ROMÁ-FERRI, 2013) suggested the COMPENDIUM ATS system for the generation of abstracts of research papers for the biomedical domain. In particular, this system produces two types of generic summaries: extractive and abstractive-oriented, with the corresponding variants of COMPENDIUM: COMPENDIUM-E and COMPENDIUME–A, respectively. The extractive approach only selects and extracts the most relevant sentences, whilst the abstractive-oriented one combine extractive and abstractive techniques, by incorporating an information compression and fusion stage.

In (Bhat et al., 2018), Bhat et al. propose a single-document hybrid TS system called "SumItUp". The hybrid system consists of two phases: (1) Extractive Sentence Selection: uses some statistical features (sentence length, sentence position, TF-IDF, the noun phrase and verb phrase, proper noun, aggregate cosine similarity, and cue phrases) and a semantic feature (emotion described by text) to generate the summary. Cosine similarity is used to remove the redundant sentences in the extractive summary, in the (2) Abstractive Summary Generation, the extracted sentences are fed to a language generator (i.e., a combination of WordNet, part-of-speech tagger, and Lesk algorithm) to convert the extractive summary to the abstractive summary.

The process of Text Summarization (TS) can be seen from several perspectives. From a cognitive point of view, it attempts to explain the mental mechanisms involved in the reading comprehension of texts (Van Dijk & Kintsch, 1983). Instead, from a computational point of view, the process does not attempt to go into such detail. On the contrary, it is more concerned with the techniques and approaches suitably for generating summaries automatically (Hovy, 2005). In the next section we present our ATS approach EXABSUM which produces different types of summaries (extractive and abstractive) and it is based on a cognitive perspective. In addition, it takes into account the computational perspective that is needed for its automation.

4.4. EXABSUM: A hybrid Text Summarization System for Generating Extractive and Abstractive Summaries

The types of summaries EXABSUM is able to produce can be classified with respect to different factors. In this way, concerning the input, EXABSUM can either take one or several texts and produce single or multi-document summaries. Regarding the purpose of the resulting summaries, these can be generic, with the aim is to provide information about the source document(s), thus being informative. As output, the final summaries can be extractive or abstractive, or abstractive oriented summaries (i.e., a combination of extractive and abstractive information).

4.4.1. The System's Architecture

In this Subsection, we explain the two suggested approaches that the EXABSUM ATS system performs to generate the two types of summaries. First, we describe $\text{EXABSUM}_{\text{Extractive}}$, a pure extractive TS approach, and then we explain our approach $\text{EXABSUM}_{\text{Abstractive}}$ (including its two variants) that integrate abstractive techniques, leading to an abstractive summary and an abstractive oriented summary.

4.4.2. $\text{EXABSUM}_{\text{Extractive}}$ Core Stages

The first part of our work is focused on extractive summarization. A generic approach for extractive summarization is used to consider the sentence as an entity, selects the most salient ones from the text by considering some characteristic features (to ascertain the quality of sentence for inclusion in summary), and finally arrange the top *N* extracted sentences to create the summary. Hence, to generate extractive summaries, $\text{EXABSUM}_{\text{Extractive}}$ relies on four main stages (See Figure 4.6): a) surface linguistic analysis; b) Redundancy removal; c) Sentence relevance; d) Summary generation. The following core stages are described as follows:

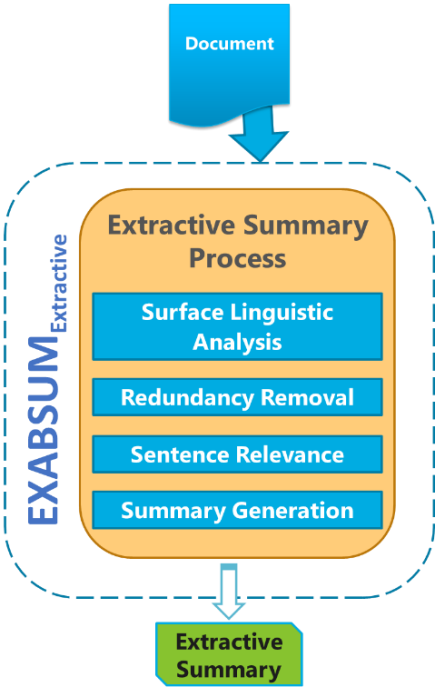


Figure. 4.6 EXABSUM$_{Extractive}$ stages for extractive summary

4.4.2.1 Surface Linguistic Analysis

First of all, basic linguistic analysis is carried out to prepare the text for further processing. The goal of Text pre-processing (TP) is to regularize the input files and set sentences boundaries for word sequences. TP can be divided into two broad categories, noise removal & normalization. Data components that are redundant to the core text analytics can be considered as noise. The way, this classical step is done can significantly affect the accuracy of the sentence selection method. Thus, it is important to give some details on how we have implemented it.

Depending on the dataset type, each document goes through the following pre-processing stages:

- ***Sentence Splitting or segmentation***: it is a preliminary step that is commonly performed on texts before further processing is the so-called sentence boundary detection or sentence segmentation, namely the process of dividing up the input text into sentences. We split the text into sentences in order to get meaningful information out.

- ***Personalized Tokenization***: for each sentence, we intelligently tokenize it by eliminating all marks, punctuation, brackets, numbers, and special characters along with lower-casing all words. For example, given the sentence: (text miningInstA˜ance; , ; :, A˜are; fea = tureA˜s), the result will be: (text mining instance are features).

- Thus, it will allow us to identify each word in the document since we will need to compute each word's co-occurrence or distinguish between stop words and nonstop words in later stages.

- **Part-of-speech tagging**. A part-of-speech tagger assigns each word with its corresponding morphological category (noun, verb, adjective, preposition, adverb, determiner, pronoun, and conjunction). This process is useful for distinguishing between types of words since some (e.g., nouns or verbs) can be more important than others (e.g., determiners). This tool will be used in the new stage of information compression and fusion. In particular, Stanford POS tagger was used as a part-of-speech tagger.

- ***Lemmatization***: The term variation can affect its frequency. Lemmatization reduces inflectional forms and derivationally related forms of a word to a common base form so they can be analyzed as a single item, and identified by the word's Lemma. Unlike stemming, lemmatization depends on correctly identifying the envisioned part of speech and meaning of a word in a sentence and within the larger context surrounding that sentence, such as neighboring sentences or even an entire document. We use the Stanford Core NLP library for lemmatizing our sentences (De Marneffe et al., 2006).

- ***Stop word identification***: Some stop words reduce the feature space, which helps in reducing time and space complexity. They include a large number of prepositions, pronouns, conjunctions in sentences. These words need to be removed before we analyze the text so that the frequently used words are mainly the words relevant to the context and not common words used in the text. In our case, we perform this step before computing single keywords relevance, since stop words will be not taken into consideration for the remaining stages.

4.4.2.2 Redundancy Detection and Removal

The redundancy is considered as a negative factor that affects the quality of summaries. In fact, the identified redundant sentences should be removed from documents, having, in the end, a single set of non-redundant sentences that is the input for the summarization approach.

Our aim in this stage is to identify semantically redundant information in the source documents, to not include it in the summary. For this purpose, Textual Entailment (TE) (Glickman, 2006) is employed. The objective of TE is to determines whether the meaning of a text snippet, also known as hypothesis (H), can be inferred from another one, called the text (T) (Glickman, 2006).

Typically, Textual Entailment (TE) is the task of predicting whether, for a pair of sentences, the facts in the first sentence necessarily imply the facts in the second. It addresses semantic inference as a direct mapping between language expressions and abstracts the common semantic inferences as needed for text based Natural Language Processing applications. TE was applied successfully to the summarization task in general (Tatar et al., 2008) (Parikh et al., 2016) (Pasunuru and Bansal, 2018), and for detecting

redundant information when addressing summarization in particular (Lloret and Manuel, 2009).

For instance, the following examples, taken from the PASCAL Recognizing Textual Entailment (RTE) challenge corpora[7], show a false and true entailment, respectively.

Pair id=18 (entailment = false)
T: Gastrointestinal bleeding can happen as an adverse effect of non-steroidal anti-inflammatory drugs such as aspirin or ibuprofen.
H: Aspirin prevents gastrointestinal bleeding.

Pair id=50 (entailment = true)
T: Edison decided to call "his" invention the Kinetoscope, combining the Greek root words "kineto" (movement), and "scopos" ("to view").
H: Edison invented the Kinetoscope.

Hence, Textual Entailment (TE) detects sentences whose context is already mentioned in other sentences, and as the information is already known, can be discarded. Therefore, by using TE, we can get a collection of sentences from the text with no entailment relation with others and then hold this set of sentences to further processing.

Algorithm 4.1 illustrate how TE is employed:

Algorithm 4.1 An example of TE Algorithm

Let's assume a document **D** consists of ***N*** sentences **(**S_1 S_2 S_3 S_4 S_5 S_6**)**
And ***N Rsent*** is the set of non-redundant sentences.
Then, in order to produce a set of non-redundant sentences, the TE is performed in this manner:
1: N Rsent = {S1}
2: N Rsent → entails → S2 ➔ NO
3: N Rsent = {S1, S2}
4: N Rsent → entails → S3 ➔ NO
5: N Rsent = {S1, S2, S3}
6: N Rsent → entails → S4 ➔ YES
7: N Rsent = {S1, S2, S3}
8: N Rsent → entails → S5 ➔ YES
9: N Rsent = {S1, S2, S3}
10: N Rsent → entails → S6 ➔ NO
11: N Rsent = {S1, S2, S3, S6}

[7] https://opendata.reseaux-energies.fr/pages/accueil/

In this example, S_4 and S_5 are removed from the text, and only the non-entailed sentences (i.e., S1, S2, S3 and S6) are reserved for further stages, thus forming the final set of non-redundant sentences (*N Rsent*). Such entailment relations are calculated using the TE approach presented in (Ferrandez, 2009). The TE tool relies on lexical (cosine similarity, Leveshtein distance), syntactic (dependency trees), and semantic measures based on WordNet (O. Ferrández et al., 2007).

Once the redundant sentences are removed from original documents, the remaining non-redundant ones will be inputted into the extractive summarization approach to detect relevant content using several scoring techniques (statistical and semantical).

4.4.2.3 Sentence Relevance

The relevancy of a sentence should be a measure of how significant is to the "understanding" of the text as a whole. This can be done by assigning scores to the most relevant words and then verifying and calculating sentence features such as their positions in the document, sentence length, title similarity. These features can be combined to score the remaining sentence and select the highest scored ones for the summary.

Mostly statistical features have been used to score sentences. Sentence salience scoring techniques (or their combinations) are applied to generate a score to each sentence depending on its importance. In our work, we propose a hybrid model, which is an extraction-based approach. This model combines statistical, structural, and semantic features. The techniques used in this step are briefly described in the next subsections:

a. Term Relevance – Inverse Sentence Frequency (TR-ISF):

We proposed a novel TR-ISF measure which is based on the classical Information Retrieval method Term Frequency - Inverse Document Frequency (TF-IDF).

This variant of the original TF-IDF is applied to text summarization at sentence level instead of the document level as TF-IDF. First, in this method, the relevancy *TR* of term *t* is calculated using its statistical and semantic relationship at the entire documents-dataset levels. Then, the *ISF* measures how much descriptive a word is, i.e., if a word is common or rare across all sentences. This method assumes that if a word is relevant, and appears in a few sentences, it will have a high probability of being included in the summary. In fact, relevant keywords can be used as a basis for identifying or measuring sentences relevance and the most relevant topic/topics of a document also.
First, we calculate the relevance of a term using '*TR'* measure using a Hybrid Feature Selection Model "HFSM". This model combines statistical and semantic features. Then, the final synthetic score of each term is calculated using TR-ISF (Eq. (4.12)) which is used later to calculate the salience score of the sentence (Eq. (4.13)). Of course, not all words should be considered, and Stopwords filtering and stemming are applied before computing the relevance of a term.

- **Term Relevance *TR*:**

The chi-square statistic allows to test whether a term and a category are statistically independent by comparing the observed frequency with the expected one, calculated under the hypothesis that they are independent. The χ^2 is defined as:

$$\chi^2_{w,c} = \sum_{i\in\{w,\bar{w}\}}\sum_{j\in\{c,\bar{c}\}}\frac{(O(i,j)-E(i,j))^2}{E(i,j)} \quad (4.1)$$

where $O(i,j)$ is the observed frequency and $E(i,j)$ is the number of documents which are in the category c and contain the term w. To determinate the nature of the dependency when it exists, Li et al. (Li et al., 2008) introduce a new term category dependency measure which is defined as:

$$R_{w,c} = \frac{O(w,c)}{E(w,c)} \quad (4.2)$$

where $R_{w,c}$ is the ratio between $O(w,c)$ and $E(w,c)$. $R_{w,c}$ should be close to 1 if there is no dependency between the term w and the category c (i.e., $\chi^2_{w,c}$ is not statistically significant), $R_{w,c}$ should be larger than *1* if there is a positive dependency, which means that the observed frequency should be larger than the expected frequency. $R_{w,c}$ should be smaller than *1* if there is negative dependency.

In order to calculate the feature goodness of word w in a corpus with k categories, Li et al. (Li et al., 2008) combine equations $\chi^2_{w,c}$ statistic (1) and $R_{w,c}$ measure (2) and propose a new measure called CHIR, defined as follows:

$$r\chi^2(w) = \sum_{j=1}^{k} p\left(R_{w,c_j}\right)\chi^2_{w,c_j} \quad with\ R_{w,c_j} > 1 \quad (4.3)$$

where $p(R_{w,c_j})$ is the weight of chi-square statistic χ^2_{w,c_j} in the corpus in terms of R_{w,c_j}. It is defined as:

$$p\left(R_{w,c_j}\right) = \frac{R_{w,c_j}}{\sum_{j=1}^{k} R_{w,c_j}} \quad with\ R_{w,c_j} > 1 \quad (4.4)$$

This new term-goodness measure, $r\chi^2(w)$, is the weighted sum of χ^2_{w,c_j} statistics when there is a positive dependency between the term χ^2_{w,c_j} and the category c_j, a bigger $r\chi^2(w)$ or CHIR measure value indicates that the term is more relevant.

To increase the semantic aspect of selected words in a specified context, and measure the its importance based on their semantic contents, we used the measure based on the Mutual Information (SIM) which is widely used in information theory. It was proposed as a measure of word association and reflects the strength of the relationship between words by comparing their actual co-occurrence probability with the probability that would be expected by chance.

Generally, Mutual Information represents the relative change in the probability of observing x when y is present (the amount of information that y provides about x) (Dagan et al., 1993).

It is based on the fact that two words are considered similar if their mutual information with all the words in the vocabulary V is nearly the same (Dagan et al., 1993). The semantic similarity measure between two terms w_1 and w_2 is defined as follows:

$$sim(w_1, w_2) = \frac{1}{2|V|} \sum_{i=1}^{|V|} \left(\frac{min(I(z_i,w_1),I(z_i,w_2))}{max(I(z_i,w_1),I(z_i,w_2))} + \frac{min\ (I(w_1,z_i),I(w_2,z_i))}{max\ (I(w_1,z_i),I(w_2,z_i))} \right) \quad (4.5)$$

where V is the vocabulary and $I(z_i\ , w_1)$ is the mutual information between the term z_i and w_1. $I(z_i\ , w_1)$ is evaluated using the following formula:

$$I(z_i\ , w_1) = P_d(z_i\ , w_1) log(\frac{P_d(z_i\ , w_1)}{P(z_i\ )P(w_1)}) \quad (4.6)$$

where d represent the size of a sliding window, $P_d(z_i\ , w_1)$ is the probability of succession of z_i and w_1 in a window of $(d + 1)$ words and $P(z_i\ )$ is the priori probability of the term z_i . This probability can be estimated by the ratio of the number of times that z_i is followed by w within the window and by the cardinal of the vocabulary.

The similarity between a term w and a document centroïd d is defined in (Benghabrit et al., 2013) as the average of the similarities between the term w and the *x* terms of the document centroïd. This measure is given by:

$$SIM(w, d) = \frac{\sum_{j=1}^{x} sim(w,w_j)}{\sum_{j=1}^{x} \sum_{i=1}^{x} sim(w_j,w_i)} \quad (4.7)$$

so as to determine the semantic relevance of a term w in a corpus of *k* clusters, for each cluster we calculate the weighted sum of its similarities with the document centroid $dcen_j$ of each cluster c_j using the following formula:

$$SIM(w) = \sum_{j=1}^{k} P\left(I\left(w, dcen_j\right)\right) SIM(w, dcen_j) \quad (4.8)$$

where $P\left(I(w, cen_j)\right)$ is the weight of the similarity between the term w and the document centroïd $dcen_j$ and $I(w, dcen_j)$ is the mutual information between w and $dcen_j$.

Considering the contingency table of a term w and a centroïd d where A is the number of times w and d co-occur i.e., w occur in documents that belong to the cluster whose centroïd is d, B is the number of times w occurs without d, C is the number of times d occurs without w and N is the total number of documents.

The mutual information criterion between a term w and a document $dcen_j$ is defined by:

$$I(w, dcen_j) = P(w, dcen_j)\, log(\frac{P(w, dcen_j)}{P(w)P(dcen_j)}) \qquad (4.9)$$

If there is a strong association between w and $dcen_j$ then the joint probability $P(w, dcen_j)$ will be larger than $P(w)P(dcen_j)$; consequently $I(w, cen_j) > 0$. If w and $dcen_j$ are in complementary distribution, then $P(w, dcen_j)$ will be less than $P(w)P(dcen_j)$ hence $I(w, dcen_j) < 0$.

In the case of poor association between w and $dcen_j$, then $P(w, dcen_j) \approx P(w)\ P\ (dcen_j)$, consequently $I(w, dcen_j) \approx 0$. The weight of $P\left(I(w, dcen_j)\right)$ defined as:

$$P\left(I(w, dcen_j)\right) = \frac{I(w, dcen_j)}{\sum_{i=1}^{k}\left(I(w, dcen_j)\right)} \ with\ I(w, dcen_j) > 0 \quad (4.10)$$

The high weight of the $SIM(w)$ measure of a term indicates that this term is relevant semantically.
The keyword Hybrid Feature Selection Model (KWHFSM).

We define the feature goodness of a term w as a weighted sum of its statistical measure $chir(w)$, and its semantic measure $sim(w)$. The global measure of a term relevance $TR(w)$ is defined as follows:

$$TR(w) = \alpha * chir(w) + (1 - \alpha) * sim(w) \quad (4.11)$$

where α is a weighting parameter between 0 and 1. To select the most p pertinent terms, three steps are followed: (1) calculate the hybrid measure $TR(w)$ for each term in the document and the dataset, (2) sort the term in descending order of their criterion function, and (3) finally select the top p terms from the sorted list. A threshold δ is set to 0.25 to filter terms with a low $TR(w)$ value. In other words, the higher the relevancy of a word, the more important it is to indicate the main topic of a document.

- $\boldsymbol{TR - ISF}$

The $\boldsymbol{TR - ISF}$ of a word is computed as shown in Eq. (4.12) and the salience score of a sentence is calculated as presented in Eq. (4.13).

$$TR - ISF(w_i) = TR(w_i) \times \log\left(\frac{S}{S_{w_i}}\right) \qquad (4.12)$$

$$TR - ISF(s_i) = \sum_{w_j \in T}^{S} TR - ISF(w_j) \qquad (4.13)$$

Where

- TF returns the relevancy of a term(word) w_j in the document(s),
- T is the total of terms (words) in s_i,
- S_{w_i} is the total of sentences in which a relevant word w_i is presented (calculated using Eq 4.11),
- S is the total of sentences in the document.

a. Sentence Resemblance to The Title:

The title usually reflects the main topics discussed in a document, especially in the news articles and scientific papers. The sentence resemblance to the title methodology measures the similarity between a document's sentences and its title. By using this technique, we assume that sentences with a higher degree of similarity with the title indicate the main topic discussed in a document (Edmundson, 1969) (Ferreira et al., 2013).

This feature is computed as presented in the following equation:

$$SenRT(S_i) = \frac{w_{s_i} \cap w_t}{|w_t|} \qquad (4.14)$$

Where,

- w_{s_i} is the set of the relevant words in s_i
- w_t is the set of word in the title
- $|w_t|$ is the total of words in the title.

b. Sentence Length:

Sentence length measure is used to avoid selecting concise sentences that may not represent the main topics of a document, and similarly, avoid selecting very long sentences that may be a waste of space. Considering that a sentence may contain important information in part and non-relevant information in another (Ferreira et al. 2013). Thus, the method uses length as the number of words in a sentence. First, we eliminate sentences smaller or larger than a given threshold (we remove sentences with less

than ten non-stop words and sentences containing more than 50 non-stop words before computing the sentence score). Then, we score the remaining sentences as presented in Eq. (4.15).

$$SentenceLen(s_i) = \frac{\#number\_of\_words\_in\_s_i}{\#\max\ \_number\_of\_words\_in\_a\_sentence} \quad (4.15)$$

Practically, the penalty score is calculated using a conditional:

$$\boldsymbol{S}core(S_i) = \begin{cases} L_i & \text{if}(\mathrm{L_i} > \mathrm{CL}) \\ L_i - CL & \text{othetwise} \end{cases} \quad (4.16)$$

Where,

- L_i is the length of sentence i and
- CL is certain length defined by user.

c. Sentence position:

The position of a sentence in a document is one of the most effective heuristics to select relevant sentences for automatic text summarization (Ouyang et al., 2010). The sentence position heuristic assumes that the first sentences in a document constitute the most relevant information. Their importance decreases as the sentence goes further in from the start of the text. In our case, we give more importance only to sentences located at the beginning of a document (Abuobieda et al., 2012).

This feature score is computed as the following:

$$SentPosition(S_i) = 1 - \frac{i}{S} \quad (4.17)$$

Where

- i is the i_{th} sentence in the document, with i starting by zero,
- S is the total of sentences in the document.

4.4.2.4 Summary Generation

Having computed each sentence's score, the objective of this stage is to generate a summary by ranking sentences according to their relevance, and the highest ones are selected and extracted in the same order as they appear in the original document, thus generating a relevant extractive summary. To calculate the overall sentence's relevance, we followed the averaged combination (which is considered as the best combination and leads to significant improvements in most of the cases (Ferreira et al. 2013) (Oliveira et al., 2016)).

In this combination, the salience score of a sentence is given by the average of *N* individual score techniques under consideration.

4.4.3. EXABSUM$_{Abstractive}$ Core Stages

This stage aims to generate an abstractive summary by producing new texts from the original text's fragments of information or concepts. This summary presents the primary information of the document concisely and coherently. To achieve this goal, we first generate a full abstractive summary by applying a graph-based method followed by a re-ranking stage based on keyphrases.

The main steps (See Figure 4.7) involved in this stage are as follows:

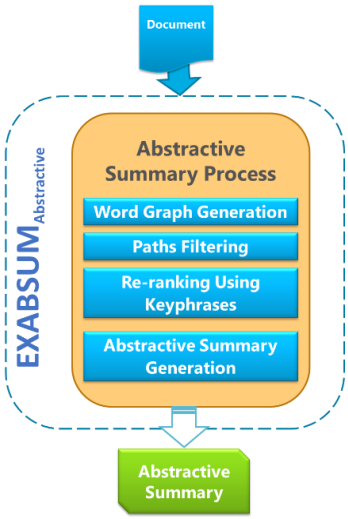


Figure. 4.7 EXABSUM$_{Abstractive}$ stages for generating abstractive summary

4.4.3.1 Word Graph Generation and Re-Ranking:

For generating a summary that includes new sentences, we apply this stage to compress, merge sentences, and then re-rank these sentences according to the number and relevance of keyphrases they contain. This process, has been proven that it is the most appropriate one to generate more informative summaries (Boudin and Morin, 2013). In this stage, we rely on word graphs adopting a similar approach to the ones described in Filippova (Filippova, 2010) and Boudin and Morin (Boudin & Morin, 2013).

Specifically, a weighted directed word graph is built, taking a document (represented as a directed weighted graph) as input. The words represent the nodes of the graph, and the edges are adjacency relationships between two words. The weight of each edge is calculated based on the inverse frequency of the co-occurrence of two words.

Once the document is represented as a word graph, a pool of new sentences is created by identifying the shortest path between nodes, beginning with each sentence's first word in the document extract, to cover its entire contents Filippova (Filippova, 2010). The method is described as follows:

Let $G = (V, E)$ be a directed graph with the set of vertices (nodes) V and a set of directed edges E, where E is a subset of $V * V$. Given a set of related sentences $S = (s_1, s_2, \dots, s_n)$, a word graph is constructed by iteratively adding sentences to it.

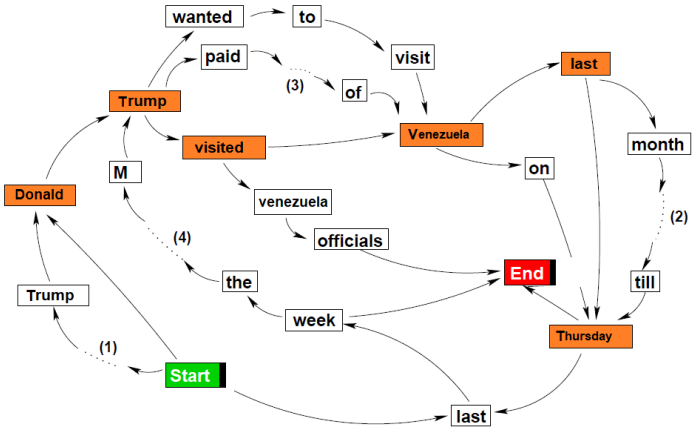


Figure. 4.8: Word graph built from sentences (1-4) and a possible compression path is given

Figure 4.8 illustrate the word graph built from the four sentences below. For clarity, edge weights are omitted and italicized fragments from the sentences are replaced by dots.

1. The president of U.S. Donald Trump visited Venezuela last Thursday.
2. Donald Trump did a visit to the People Republic of Venezuela on Thursday.
3. Last week the President of State M. Trump visited Venezuela officials.
4. Donald Trump wanted to visit Venezuela last month but suspended his arrangements till Thursday last week.

In the first step, the graph simply represents one sentence (a string of word nodes without punctuation) plus the start and end symbols (start and end symbols in Figure 4.8). For each word in the sentence, a node is added to G, and the words adjacent to the sentence are connected to the directed edges. A word from the following sentences is mapped to an existing node in the graph if they have the same lowercased word form, and part of speech 'POS' and no word from that sentence has been mapped to this node. Using POS information decreases the chances of merging verbs with nouns (e.g., visit) and generating ungrammatical sequences. A new node is created when there is no candidate in the graph.

In the second step, the word mapping/creation (adding words to the graph) is done in the three-following order:

1. non-stop words for which no candidate exists in the graph or for which an unambiguous mapping is possible;
2. non-stop words for which there are either several possible candidates in the graph, or which occur more than once in the sentence
3. stop words

For the last two groups of words where the mapping is ambiguous (i.e., there are two or more nodes in the graph that refer to the same word / POS tuple), the immediate context (the preceding and following words in the sentence and the adjacent nodes in the graph) is checked. Consequently, the candidate that has a larger overlap in the context is selected. Alternatively, the one with a higher frequency (i.e., the node which has words mapped onto it) is used to select this candidate node.

For example, in Figure 4.8, when sentence (3) is to be added, there are two candidate nodes for last. Stopwords are mapped only if there is some overlap in non-stop word neighbors. Otherwise, a new node is created. We use the stopword list included in NLTK8 extended with temporal nouns (e.g., Thursday, today). In Filippova's approach, punctuation marks are excluded. Boudin and Morin (Boudin and Morin, 2013) added a fourth step for adding punctuation marks in the graph to generate well-punctuated compressions. When the mapping is ambiguous, they select the candidate who has the same immediate context. Once the sentence's words are added to the graph, words adjacent in the sentence are connected with directed edges. To calculate edge weights and find the best path, that is, the best compression for the input sentences, the weighting function is defined in Equation 4.18:

$$w(i,j) = \frac{cohesion(i,j)}{freq(i) \times freq(j)} \qquad (4.18)$$

$$cohesion(i,j) = \frac{freq(i)+freq(j)}{\sum_{s \in S} d(s,i,j)^{-1}} \qquad (4.19)$$

where $freq(i)$ is the number of words mapped to the node i. The function $d(s,i,j)$ refers to the distance between the offset positions of words i and j in sentence s.

Two aims of this function are two-fold:

(1) to generate a grammatical compression, links between words which often appear in this order are favored (see Equation 4.19).
(2) to generate an informative compression, it promotes paths passing through salient nodes.

[8] http://nltk.org/

The K-shortest path algorithm is used to find the shortest paths in the graph from start to end using the weighting function in (20). Paths shorter than eight words, or not including a verb, are filtered. Eventually, the remaining paths are reranked by normalizing the total weight of the path over its length. Thus, the path with the lightest average edge weight is then considered to be the best compression. In our case, for obtaining new sentences, the starting node corresponds to the first word in each sentence. Thus, we ensure that we have at least one derived sentence for each sentence in the source document, so the entire content of the document is covered.

4.4.3.2 Paths Filtering

After obtaining the resulting sentences with shortest paths, some might be wholly incomprehensible and not correctly formed or suffer from incompleteness. Therefore, a filtering stage is needed to discard the invalid paths and guarantee the completeness and correctness of sentences. We define rules such that sentences that do not comply with all the rules are not taken into account.

These rules are defined as follows:

- Every sentence must contain a verb (FILIPPOVA, 2010).
- The minimal length for a sentence must be three words.
- The sentence should not end in an article (e.g., a, the), a preposition (e.g., of), an interrogative word (e.g., who), or conjunction (e.g., and).

Once the incorrect sentences have been removed, we can use the new sentences instead of the original ones. It is important to stress upon the fact that, in EXABSUM, this stage takes place after the relevance detection and before the summary generation stages. This is because we need to be aware of the most relevant sentences of the document first.

4.4.3.3 Re-Ranking Candidate Sentences Using Keyphrases

Although the Filippova's approach seems to work well, the relevant information is missing from 48% to 60% of the generated sentences (FILIPPOVA, 2010). It is due to the node salience which is determined only with the frequency measure. To resolve this problem, Boudin and Morin (Boudin and Morin, 2013) proposed a re-ranking approach that re-ranks the N-best list of compressions according to the number and relevance of keyphrases they contain. Effectively, an informative and relevant sentence should contain the most relevant keyphrases (Merrouni et al., 2019).

Hence, we added a re-reranking step that favors compressions that contain the most relevant keyphrases of the input sentence set. Consequently, the *N-best* multi-sentence compressions candidates generated by the word graph-based approach will be re-ranked according to the number and relevance of the keyphrases they contain.

The reason why we used the shortest path and adding later a re-ranking step is threefold.

- First, the shortest path allows sentences to be compressed while maintaining salient words from the input and put together words found next to each other in many sentences.

- Second, we can include more content in summary in the case of several sentences are fused.

- Third, the re-ranking step using keyphrases will produce more informative sentences by maximizing the range of topics they cover and generating sentences that are informative and grammatically correct. Indeed, keyphrases can be used to generate better sentences that convey the gist of the set of related sentences (Boudin and Morin, 2013).

The unsupervised method based on Wan and Xiao (Wan and Xiao, 2008) extracts keyphrases from each set of related sentences. This method is grounded on the assumption that a word recommends other co-occurring words. The strength of the recommendation is recursively determined based on the value of the word's importance, making the recommendation.

First, to extract keyphrases, a weighted graph is constructed from related sentences in which nodes represent words defined as word and POS tuples. Two nodes (words) are connected when their corresponding lexical units co-occur within a sentence. Edge weights are the number of times that two words co-occur. TextRank (Mihalcea and Tarau, 2004), a graph-based ranking algorithm that takes edge weights into account, is applied to compute each node's salience score. The score for node V_i is initialized with a default value and is computed iteratively until convergence using the following equation:

$$(V_i) = (1-d) + d \times \sum_{V_j \in adj(V_i)} \frac{w_{ji}}{\sum_{V_k \in adj(V_i)} w_{jk}} S(V_i) \qquad (4.20)$$

where $adj(V_i)$ represents the neighbors of V_i and d is the damping factor set to 0.85.

The second step consists of generating and scoring candidates keyphrases. Sequences of adjacent words satisfying a specific syntactic pattern are collapsed into multi-word phrases. In our case, we used our POS tag definition (proposed in Chapter 3) of noun phrases that satisfy the following regular expression rule: (NN | NNS | NNP | NNPS | VBN | JJ | JJS | RB) * (NN | NNS | NNP | NNPS | VBG) +. Unlike other definitions, our structure of noun phrases can comprise an adverbial noun (tag RB) such as "double experience" (RB NN), and verb in present participle (tag VBG) like "virtual desktop conferencing" (JJ NN VBG), where the VBG tag can be in the beginning, the middle or the end of the noun phrase. Adverbial nouns are sometimes referred to as adverbial objectives, they hold a position usually occupied by a verb's direct object and modify the verb with an aspect of time, distance, weight, age, or monetary value. Adverbs, in general, can interact with the noun phrase.

They can change the context and the meaning of a candidate keyphrase, especially in scientific domains where authors are more precise and specific in explaining some situations.

The score of a candidate keyphrase k is calculated by summing up the salience scores of the words that it comprises normalized by its length + 1 to favor longer n-grams (see equation 4.21).

$$\text{score}\,(k) = \frac{\sum_{w \in k} \text{Text Rank}\,(w)}{\text{length}\,(k)+1} \qquad (4.21)$$

The generated keyphrases are clustered using word overlap. For each cluster, the keyphrase with the highest score is selected. This filtering process enables the generation of a smaller subset of keyphrases while at the same time providing better coverage of the cluster content.

Re-ranking techniques may suffer from the *N-best* list's limited scope, which may exclude many potentially suitable candidates. This is why many paths than the one in Filippova approach (Filippova, 2010) are used. The paths are re-ranked by normalizing the path's total weight over its length, multiplied by the sum of the keyphrase scores it contains.

The score for sentence compression c is given by:

$$\text{score}\,(c) = \frac{\sum_{i,j \in \text{path}\,(c)} w_{(i,j)}}{\text{length}\,(c) \times \sum_{k \in c} \text{score}\,(k)} \qquad (4.23)$$

After applying the steps mentioned above, the remaining sentences will be used for building abstractive summaries. Thus, the resulting of steps above produces an abstractive summary with correct and complete formed sentences having the shortest paths, and the top *N* relevant ones are those with the highest keyphrases they contain.

4.4.4. EXABSUM$_{\text{Hybrid}}$ Hybrid Summary Generation

In this last step, we aim to generate the final abstractive oriented summary. To do that, we take full advantage of extractive and abstractive strategies by combining them adopting a hybrid approach (see Figure 4.9).

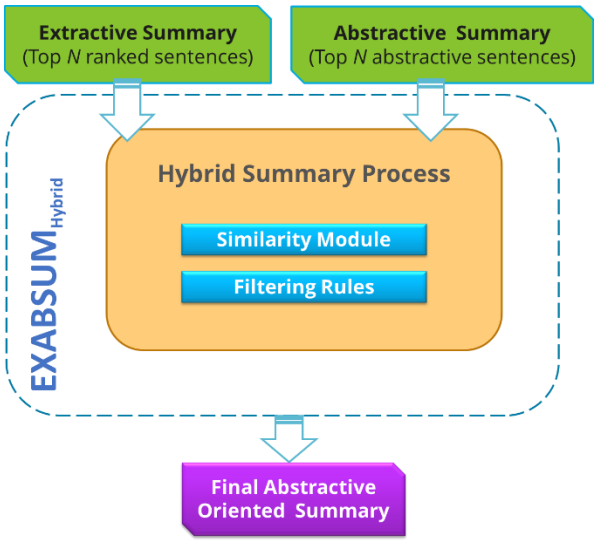


Figure. 4.9: Hybrid Summary Process

The final summary will be selected following a strategy that maximizes the similarity (using the cosine measure) between each of the new sentences generated in the information compression, fusion, and re-ranking stages, and the ones that have been selected as most pertinent in the sentence relevance stage. Given that the similarity between them is above a predefined threshold set (fixed to 0.5). In particular, we combined both types of summaries, respecting these rules:

- In the cases where a sentence in the extract (extractive summary) has an equivalent in the newly generated sentences (abstractive summary), the previous will be replaced for the latter.
- If the sentence in the extract does not correspond to any sentence in the abstractive summary, we keep the sentence in the extractive one.
- If the abstract similarity contains some sentences that are not present in the extract, we enrich the extract with these sentences.

In this manner, the newly produced summary contains both extractive and abstractive information. In all cases, the resulting summaries will be abstracts, since they do not reproduce verbatim the source document's sentences.

4.4.5. Experimental Setup

In order to evaluate the performance of EXABSUM, several experiments with various corpora from a wide range of domains are conducted. Thus, in this section, we describe: (i) the used datasets and the evaluation methods for the experiments; (ii) the experimentation for tuning some parameters. Finally, the results of our proposal are compared to other related works.

4.4.5.1. Datasets

It is a common practice to run an algorithm over a specific corpus of text summarization task that comprises the source documents with their summaries created by humans. As corpora, we used several data sets belonging to different domains. In this manner, EXABSUM can be assessed from a broader perspective by focusing on the newswire, tourist, Web 2.0, scientific, business, health, justice, living, opinion, politics, showbiz, sports, technology, and travel domains.

In particular, EXABSUM is evaluated on the following corpora:

- ***DUC 2001 and DUC 2002*** are considered as the most common used corpus in ATS task (Cao, 2017). They were published by the Document Understanding Collection (DUC) and Text Analytics Conferences (TAC). DUC 2001 contains 309 news articles in English, where each one has the golden summaries created by two different people. DUC 2002 contains 567 news articles in English of different lengths and different topics. Also, the two gold-standard summaries were created by two human experts. For both sets, the corresponding summaries have a length of around 100 words.

  It is worth noting that the DUC collections are already divided by sentences only to determine which sentences produce the best description. DUC datasets come from a wide range of categories. Their documents can be classified into 11 categories, i.e., Biography, Politics, Law, Society, Culture, Business, Health Natural Disaster, Science, Sports, and International. In particular, the Distribution Group for DUC 2002 'Natural Disaster,' 'Politics and History,' accounts for 60 percent of the documents in these categories. All DUC documents and clusters include human models available with approximately 100 words.

- ***CNN Corpus*** (Lins et al., 2019): the CNN corpus is a collection of news documents for single-document summarization based on the news articles from the CNN website (http://www.cnn.com). It is the largest test set available today for single document extractive summarization. The corpus consists of 3000 articles in English distributed into twelve subject categories, originally tagged by CNN: Business, Opinion, Politics, Showbiz, Health, Justice, Living, Sports, world news, Technology, United Stated, and Travel. An essential aspect of this dataset is the availability of good quality abstractive summaries for each document written by the original authors, called highlights, and extractive summaries (the gold standards) as well (available with approximately 90~100 words). They may serve as the reference for qualitative and quantitative assessment of automatic summarization techniques.

- The gold standard encompasses 10,754 sentences, around 10% of the total number of sentences for 3000 texts in the CNN-corpus. The experiments reported here were performed using the gold-standards of the CNN-corpus. The CNN-corpus is currently being used in many research initiatives ranging from: the analysis and resolution of dangling coreference, improving extractive summarization techniques, and automatically generating abstractive summaries from extractive ones. It was also recently used in the DocEng'19 Competition on Extractive Text Summarization (Dueire Lins and Simske, 2019).

Table 4.1: Statistics on the CNN and DUC datasets.

Dataset	Number of clusters	Domain's documents	Number of Documents	Sentences	Number of test documents	Avg. length (model sum)	Task
DUC01	30	Multi-Domain	309	269,990	309	100	Single and Multi
DUC02	59	Multi-Domain	567	348,012	567	100	Single and Multi
CNN	0	Multi-Domain	3000	2,628,336	2000	90	Single

Table 4.1 shows an overview of all the data sets we have worked with. Besides, some basic statistics of these corpora are also provided. In particular, for each corpus, the table shows the number of clusters, the document's domains, the number of total documents within each dataset, the total number of sentences, the available number of documents for the test, the average length of the model summaries (in the number of words, as well), and finally the target task.

4.4.5.2. Evaluation Method

We setup two different types of evaluation. Quantitative and qualitative (see Chapter 2, section 2.4.6 for more details). In the first one, we aimed to compare our results with the articles gold-standard models using a state-of-the-art evaluation tool. The second evaluation aims at determining to what extent our generated summaries contain the main topics of the research articles. Thus, we assess the summaries with regard to user satisfaction.

Specifically, for comparison, in quantitative evaluation, we measure the similarity between a set of candidate summaries with a collection of models (gold standard summaries). The goal of this evaluation is to assess the informativeness of the summaries with respect to their content. For this purpose, ROUGE-N provides different levels of N-gram co-occurrences between the candidate summaries and the summary models. The most well-known ROUGE metrics are ROUGE-1 and ROUGE-2, which compute the number of overlapping unigrams and bigrams. In particular, ROUGE-1 recall has the highest recall ability to identify a better summary in a pair (Lin, 2004). Formally, ROUGE-N is an n-gram recall between a candidate summary and a set of reference summaries. ROUGE-N is computed as follows:

$$ROUGE\text{-}N=\frac{\sum_{S\in\{ReferemceSummaries\}}\ \sum_{gram_n\in S}\ Count_{match}(gram_n)}{\sum_{S\in\{ReferenceSummaries}\ \sum_{gram_n\in S}\ Count(gram_n)} \quad (4.24)$$

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 a candidate summary and a set of reference summaries (Lin, 2004).

Furthermore, Lin (Lin, 2004) showed that ROUGE-1 recall achieves a high correlation with human judgments. ROUGE-SU4 is also used. It measures the overlap of skip-bigrams. An automatic summary comprises a model one, with a maximum distance of four words between them, and finally, ROUGE-L, which measures the longest common subsequence between two summaries.

In the last evaluation, we aim to assess the user satisfaction with respect to the generated summaries. For this purpose, we performed a qualitative evaluation and we asked 10 humans with English speaking knowledge to evaluate our summaries. We followed the same qualitative evaluation presented in (Lloret et al., 2012). For instance, a 3-level scale may comprise the values “low”, “medium”, and “high”, whereas in a 5-level Likert scale the degrees to measure the agreement with respect to a specific issue are established (“strongly agree”, “agree”, “neither agree nor disagree”, “disagree”, and “strongly disagree”).

Specifically, the asked questions are:

- Q1: The summary reflects the most important issues of the document.
- Q2: The summary allows the reader to know what the article is about.
- Q3: After reading the original summary provided with the document, the alternative summary is also valid.

Due to the numerous sizes of the documents, this evaluation method was processed using only 10 random generated documents of each tested dataset.

4.4.6. Experiments Results and Discussion

In this section, the experiments were conducted for evaluating the different types of summaries of EXABSUM. We perform a single document summary evaluation using the two variants of HYBRYSUM.

As previously explained, EXABSUM produces two kinds of summaries: i) extractive summaries using $EXABSUM_{Extractive}$, ii) abstractive summaries using $EXABSUM_{Abstractive}$, and abstractive oriented summaries using $EXABSUM_{Hybrid}$.

Thus, by evaluating both, we can examine, on the one hand, whether EXABSUM is capable of determining the most relevant information, and on the other hand, which summarization method will perform well regarding results and address the abstractive ATS challenge. Moreover, we can analyze if the proposed techniques within EXABSUM are appropriate for generating summaries across different well-known domains (Business, Opinion, Politics, Showbiz, Health, Justice, Living, Sports, Technology, Travel, newswire, etc.). Specifically, the experiments aim first to explore how the proposed techniques of sentence relevance detection can influence the performance of EXABSUM. Then, to make results more convincing, we compare them with some representative existing ATS systems.

4.4.6.1. Parameter Values Selection and Appropriateness of The Analysis of The Used Scoring Techniques

In this evaluation, we aim at evaluating the appropriateness of the proposed features for sentence relevance detection stage (see Section 3.4). The carried-out evaluation comprises the analysis of these features on their own, and on their combination, as it was explained in the corresponding subsections.

Regarding the Equation (4.13), which determines the single term's relevancy within a sentence, the weighting parameter α is necessary. It represents the weight associated to the statistical feature (CHIR) and the semantic feature (SIM) in the hybrid weighting model. To estimate the importance of each feature about its contribution to the relevancy of the keyword $TR(w)$, we run several experiments using different α values ($\alpha \in \{0,\ 0.2,\ 0.5,\ 0.6, 0.8,\ 1\}$). The results obtained using $\alpha = 0.6$ are better than those obtained by all other values.

Thus, while calculating the $TR - ISF$ measure, we fixed α to 0.6 in all the experiments. Then, regarding the other sentence relevancy scoring techniques, we conducted an ablation study using a backward-like total exclusion strategy. Each technique's score is individually removed or added in the weighting averaged model.

The evaluation using this strategy is threefold: (1) It will let us decide whether or not the scoring techniques are proper to improve the results (ROUGE scores); (2) Identify their role in covering the topics of the document; And (3) determine the relevance of a sentence.

To show the contribution of each summaries type (extract or abstract), we run the same experiments using EXABSUM variants: $\text{EXABSUM}_{\text{Extractive}}$, $\text{EXABSUM}_{\text{Abstractive}}$, and $\text{EXABSUM}_{\text{Hybrid.}}$ As mentioned before, the $\text{EXABSUM}_{\text{Abstractive}}$ takes a document as input and generates new sentences from the first words of each original sentence using the graph-based approach and the re-ranking stage based on keyphrases. $\text{EXABSUM}_{\text{Hybrid}}$ summary will be combined with the extractive one to produce the final abstractive oriented summary.

Table 4.2 shows an example of two summaries generated with $EXABSUM_{Extractive}$ and $EXABSUM_{Hybrid}$, respectively. As can be seen, both summaries share some of the sentences, whereas others have been shortened in the latter.

Table 4.2 Example of summaries generated with $EXABSUM_{Extractive}$ and $EXABSUM_{Hybrid}$, respectively for document WSJ891019-0021 (DUC 2002 corpus, cluster d062j), with 50% ratio of the original text.

$EXABSUM_{Extractive}$
The White House is making sure nobody will accuse it of taking this crisis lightly. Mr Bush and his aides were accused of responding too slowly after the Exxon Valdez oil tanker split open in Alaskan waters and Hurricane Hugo struck the Carolina coast, and they clearly do not want a repeat of those charges now. So, the White House announced that Mr Bush got his first earthquake briefing of the day at 6:30 am from chief of staff John Sununu. Mr. Bush had taken two phone calls from Vice President Dan Quayle, who was in California; made a televised statement of concern; signed a disaster proclamation; received a written report from the Federal Emergency Management Agency; and visited FEMA headquarters On his FEMA visit, Mr Bush said that he hoped there would be less carping about the emergency office performance this time, adding that the agency took a hit for its reaction to Hurricane Hugo. The White House already is talking of Mr Bush visiting the California earthquake site this weekend.
$EXABSUM_{Hybrid}$
the white house announced that mr bush got his first earthquake briefing of the day at 6:30 am already visiting the California earthquake site this weekend. from chief of staff John sununu making sure nobody will accuse it of taking this crisis lightly. mr bush and his aides were accused of responding too slowly after the Exxon Valdez oil tanker split open in alaskan waters and hurricane hugo struck the Carolina coast and they do not want a repeat of those charges now. on fema visit mr bush said that he hoped there would be less carping about the emergency office performance this time, adding that the agency took a hit for its reaction. mr bush taken two phone calls from vice president dan quayle, who was in california made a televised statement of concern signed a disaster proclamation; received a written report from the federal emergency management agency.

By applying the ablation strategy, we present in Table 4.3 the obtained results of ROUGE-1, ROUGE-2, for the DUC01 dataset. On the other hand, Table 4.4 shows the results of ROUGE-1, ROUGE-2, ROUGE-SU4, and ROUGE-L for the DUC02 dataset. In contrast, in Table 4.5, we show the results regarding ROUGE-1, ROUGE-2 scores for the CNN dataset.

The behavior of our proposed approach while varying the scoring techniques values is shown in Figure 4.10, Figure 4.11, and Figure 4.12 respectively.

As shown in Table 4.3 and Table 4.4, in the first experiment, we select as summary sentences those containing only the relevant keywords and topics identified by $TR - ISF$ scoring technique (which corresponds to the sentence relevance identification stage of EXABSUM). The TR is used to identify relevant keywords that identify important topics, and *ISF* measures how much descriptive a word is. In order to build a summary, we rank sentences according to the Equation 4.13, and the top-ranked ones are generated to produce the summary.

Table 4.3: ROUGE results for $EXABSUM_{Extractive}$ on DUC2001 dataset while analyzing its features, *Comb* denotes the combination of the selected scoring techniques

		DUC 2001	
Summary Type	Scoring Techniques	ROUGE-1	ROUGE-2
$EXABSUM_{Extractive}$	$TR - ISF$ ONLY	0.398	0.131
$EXABSUM_{Extractive}$	Comb1: $TR - ISF$ + Sentence position	0.402	0.143
$EXABSUM_{Extractive}$	Comb2: $TR - ISF$ + Sentence Position + Sentence Length	0.453	0.192
$EXABSUM_{Extractive}$	Comb3: $TR - ISF$ + Sentence Position + Sentence Length+ Sentence resemblance To the Title	0.480	0.208
$EXABSUM_{Abstractive}$	Graphs + Reranking based on keyphrases ONLY	0.319	0.151
$EXABSUM_{Hybrid}$	$EXABSUM_{Extractive}$ (with Comb3) +Graphs+ Reranking based on keyphrases	0.484	0.211

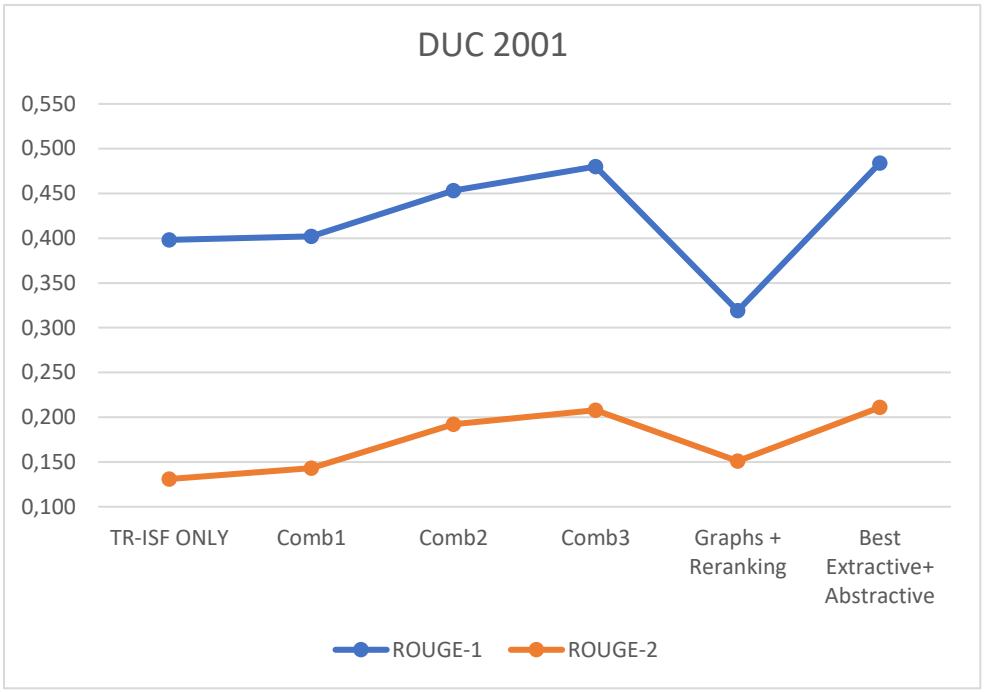


Figure. 4.10: ROUGE-1 and ROUGE-2 results of EXABSUM on the DUC 2001 collection while varying the scoring techniques and summary types.

In the second experiment, we select as summary sentences those whose relevance has been determined using only Sentence resemblance to the title, sentence length, or Sentence position, or all of them.

Table 4.4: ROUGE results for $EXABSUM_{Extractive}$ on DUC2002 dataset while analyzing its features, *Comb* denotes the combination of the selected scoring techniques

		DUC 2002			
Summary Type	Scoring Techniques	ROUGE-1	ROUGE-2	ROUGE-SU4	ROUGE-L
$EXABSUM_{Extractive}$	$TR - ISF$ ONLY	0.439	0.188	0.205	0.396
$EXABSUM_{Extractive}$	Comb1: $TR - ISF$ + Sentence position	0.441	0.189	0.207	0.401
$EXABSUM_{Extractive}$	Comb2: $TR - ISF$ + Sentence Position + Sentence Length	0.488	0.236	0.251	0.442
$EXABSUM_{Extractive}$	Comb3: $TR - ISF$ + Sentence Position + Sentence Length+ Sentence resemblance To the Title	0.493	0.257	0.288	0.472
$EXABSUM_{Abstractive}$	Graphs + Reranking based on keyphrases ONLY	0.341	0.104	0.134	0.322
$EXABSUM_{Hybrid}$	$EXABSUM_{Extractive}$ (with Comb3) +Graphs+ Reranking based on keyphrases	0.497	0.260	0.291	0.479

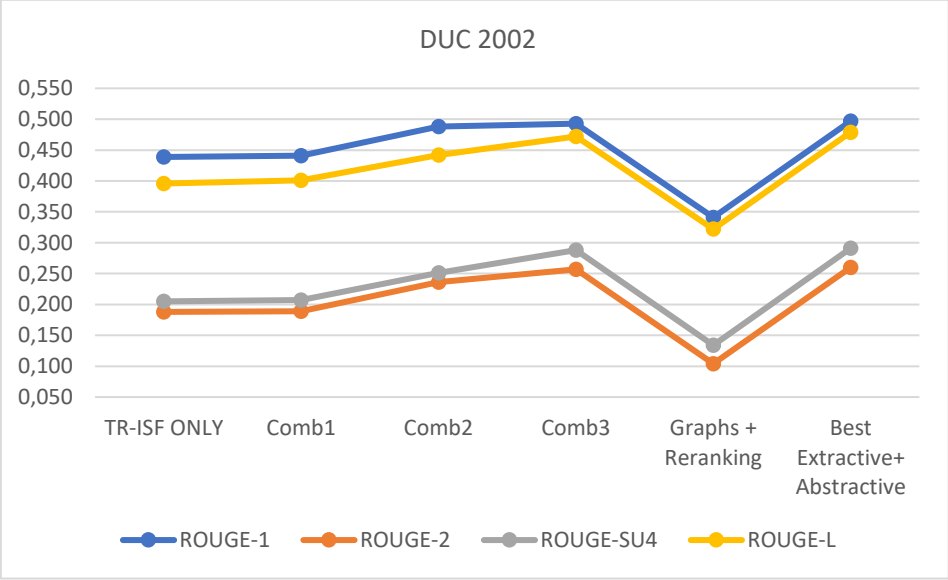


Figure. 4.11: ROUGE-1, ROUGE-2, ROUGE-SU4, ROUGE-L results of EXABSUM on the DUC 2002 collection while varying the scoring techniques and summary types.

As can be seen from Table 4.3, Table 4.4, and Table 4.5, we observe a clear tendency of $EXABSUM_{Hybrid}$ in most cases regarding the best scores, especially when all the scoring techniques are combined (comb3). For example, the ROUGE-1 results for $EXABSUM_{Extractive}$ with the combination 3 improve, on average, 13,44% with respect to the $EXABSUM_{Extractive}$ approach where the $TR - ISF$ is only used to score the sentences. In the case of using Comb 2, the same approach leads to an improvement of 11.85% compared to the obtained results for $EXABSUM_{Extractive}$ using $TR - ISF$ only.

Table 4.5: ROUGE results for $EXABSUM_{Extractive}$ and $EXABSUM_{Abstractive}$ on CNN dataset while analyzing its features, *Comb* denotes the combination of the selected scoring techniques

		CNN	
Summary Type	Scoring Techniques	ROUGE-1	ROUGE-2
$EXABSUM_{Extractive}$	$TR - ISF$ ONLY	0.519	0.351
$EXABSUM_{Extractive}$	*Comb1*: $TR - ISF$ + Sentence position	0.541	0.379
$EXABSUM_{Extractive}$	*Comb2*: $TR - ISF$ + Sentence Position + Sentence Length	0.592	0.434
$EXABSUM_{Extractive}$	*Comb3:* $TR - ISF$ + Sentence Position + Sentence Length+ Sentence resemblance To the Title	0.601	0.451
$EXABSUM_{Abstractive}$	Graphs + Reranking based on keyphrases	0.501	0.332
$EXABSUM_{Hybrid}$	$EXABSUM_{Extractive}$ (with Comb3) +Graphs+ Reranking based on keyphrases	0.680	0.513

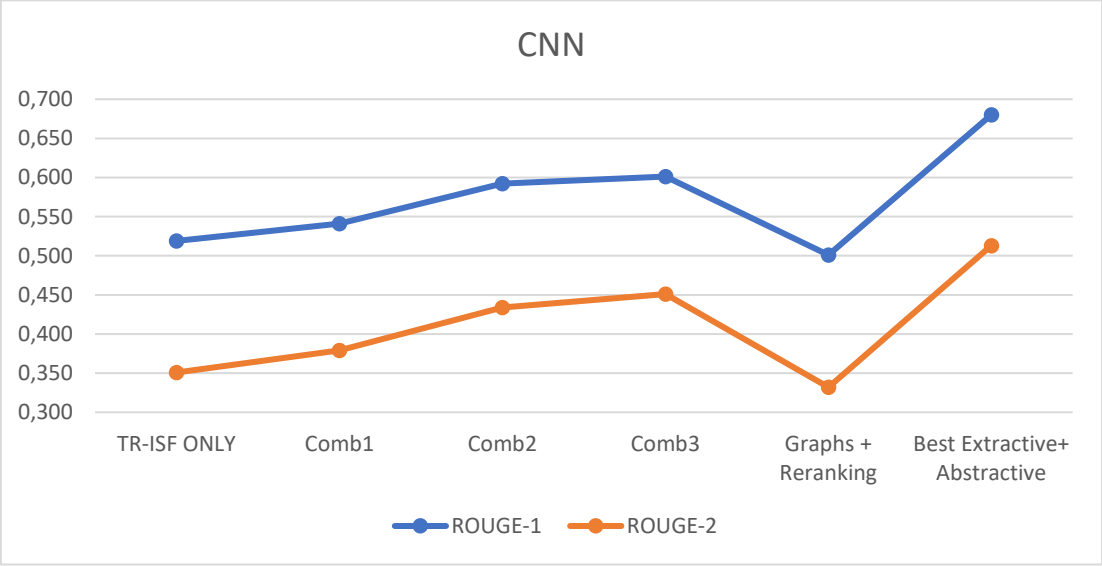


Figure. 4.12: ROUGE-1 and ROUGE-2 results of our proposed approach for the large CNN dataset varying the scoring techniques and summary types.

Regarding the analysis of the individual features, it is worth mentioning that the summaries generated using only $TR - ISF$ scoring technique perform well in general. This may be due to the fact that we do take into consideration a robust scoring technique (using the statistic and semantic features) to identify the most important terms or topics of a document. Nevertheless, when we combine the three proposed features in the same approach (Comb3), results are improved.

Hence, the well shipped proposed features are appropriate for Text Summarization task, and particularly for EXABSUM. Our system's higher ROUGE scores are attributed not only to the integration of $TR - ISF$ and other scoring techniques, but also the usage of the redundancy removal step using The Textual Entailment (TE) tool (O. Ferrández et al., 2007). This step plays a crucial role in the generation of non-semantically-statically-redundant summaries. Certainly, it identifies semantically redundant sentences and removes them from documents. In this way, the sentences whose context is already mentioned in other sentences, as the information is already known, can be discarded, and as a consequence, the precision scores increase and the overall performance of the system is increased.

Concerning $\text{EXABSUM}_{\text{Abstractive}}$ results, it appears that using only the graphs and Reranking based on keyphrases methods do not lead to high ROUGE scores, though they are promising for further research. This kind of abstractive approach's low performance can be related to the length of the summaries, which is restricted to only 100 words. Accordingly, when selecting the most important sentences before or after generating new sentences, some of the concepts may not be included. Thus, this affects the performance of the summaries, leading to low ROUGE results. Contrary to what was expected, longer sentences do not necessarily lead to better summaries, nor shorter sentences lead to more informative summaries.

To address these limitations, one possible solution to improve the selection of the newly generated summary would be to find an optimization function that could provide us with the best-generated sentences. For this aim, $\text{EXABSUM}_{\text{Abstractive}}$ is enriched with the best $\text{EXABSUM}_{\text{Extractive}}$ combination (Comb3) leading to $\text{EXABSUM}_{\text{Hybrid}}$. As can be seen from the obtained results in Table 4.3, Table 4.4, and Table 4.5, the ROUGE scores have been improved considerably using the $\text{EXABSUM}_{\text{Hybrid}}$. Moreover, there is a significant improvement of 11% on average in all ROUGE scores compared to the ones obtained from $\text{EXABSUM}_{\text{Extractive}}$. It is worth mentioning that when we take an extractive summary as a basis, and we enrich it with abstractive information generated from the source document, ROUGE scores are improved significantly compared to the initial extract. This influence positively in the resulting summaries.

Table 4.6 present our qualitative evaluation, which aim at assessing the user satisfaction with respect to the generated summaries. as can be seen from the table, the abstractive approach ($\text{EXABSUM}_{\text{Hybrid}}$) outperforms the extractive one ($\text{EXABSUM}_{\text{Extractive}}$).

Table 4.6 User satisfaction results for the different text summarization approaches

%	TS Approach	Q1	Q2	Q3
1. Strongly disagree	$EXABSUM_{Extractive}$	8,76	17,25	17,25
	$EXABSUM_{Hybrid}$	1,44	0	1,21
2. Disagree	$EXABSUM_{Extractive}$	38,6	18,01	30,1
	$EXABSUM_{Hybrid}$	28,37	19,41	27,61
3. Neither agree nor disagree	$EXABSUM_{Extractive}$	21,39	17,44	22,83
	$EXABSUM_{Hybrid}$	19,2	17,44	22,83
4. Agree	$EXABSUM_{Extractive}$	19,2	34,83	5,32
	$EXABSUM_{Hybrid}$	50,46	48,06	42,1
5. Strongly agree	$EXABSUM_{Extractive}$	6,29	13,78	22,2
	$EXABSUM_{Hybrid}$	6,29	13,78	8,55

When we have a look at the different percentages of summaries that have been rated in each category, we observe that there is a higher percentage of abstractive oriented summaries that humans agree with, compared to the extractive ones intended for the same rating. The evaluation concerning the information contained in the summaries generated with $EXABSUM_{Hybrid}$ was good as for the extractive approach, and taking into consideration their quality from a human point of view, the abstractive oriented summaries are much better than the extractive ones. Moreover, it is worth stressing upon the fact that, analogously, the percentage of summaries with lower ratings (strongly disagree and disagree) also decrease when $EXABSUM_{Hybrid}$ is employed.

4.3.6.2. Comparison to Related Works

In this subsection, we compare the performance of the top results of $EXABSUM_{Extractive}$ and $EXABSUM_{Hybrid}$ in producing single documents summaries against several state-of-the-art summarizers. In particular, we compare our summarization results with (i) the best performing participants at the DUC 2001 and 2002 shared-task; (ii) the three top-performing summarizers according to a previous evaluation reported in (Batista et al., 2015), and (iii) other (including recent ones) approaches that have used DUC01 and DUC02 collections, and ROUGE- 1 and ROUGE-2 evaluations metrics. Such approaches are briefly described in the next:

- Parveen and Strube (Parveen and Strube, 2015) proposed an unsupervised graph-based technique to summarize single documents that consider three important properties of summarization, i.e., importance, non-redundancy, and local coherence.

- Autosummarizer (Autosummarizer, 2015) consists of a web service that generates a summary by splitting and ranking the most relevant sentences. Its single-document summarization algorithm extracts the most relevant sentences from the original document, achieving good performance in a previous evaluation against different summarizers (Batista et al., 2015). Unfortunately, details of how this system works are not available.

- Classifier4J (Classifier4J, 2005) is a toolkit for text summarization and classification. It conducts an extractive single-document summarization based on word frequency and selects the first sentences containing one of the top-100 most frequent words in the document to compose the summary.

– UnifiedRank (Wan, 2010) is a method that proposes a novel unified approach to a simultaneous single document and multi-document summarization. It uses a graph-based representation and a unified ranking process.

– DE (Aliguliyev, 2009) is a summarization approach based on sentence clustering. It uses a discrete Differential Evolution algorithm and a similarity to optimize the objective function, selecting each cluster's representative sentences.

– FEOM (Song et al., 2011) proposes a Fuzzy Evolutionary Optimization Model. In this approach, sentences are categorized in terms of document content. It selects the most important sentence from each cluster to represent the overall meaning of a document.

– NetSum (Svore et al., 2007) is an approach that uses the RankNet learning algorithm to train a pair-
based sentence ranker and score every sentence in the document, thus identifying the most relevant sentences.

- Compendium (LLORET and ROMÁ-FERRI, 2013): a text summarization system applied to the generation of two types of generic summaries: extractive and abstractive, with the corresponding variants of COMPENDIUM: COMPENDIUME and COMPENDIUME–A, respectively. The extractive approach only selected and extracted the most relevant sentences. For the abstractive-oriented one, it combines extractive and abstractive techniques by incorporating an information compression and fusion stage once the most important content was identified.

- HP-UFPE Functional summarization (HP-UFPE FS) (Ferreira et al., 2014) (Ferreira et al., 2013) is a summarization system built on the seventeen extractive summarization strategies more widely acknowledged in the literature. They are extensively evaluated on scientific documents, blogs, and news articles. HP-UFPE FS system is used in this comparison by adopting the best sentence scoring combination for news articles reported in (Ferreira et al., 2014).

- Get To The Point (See et al., 2017): an abstractive summarization model that uses hybrid pointer-generator architecture with coverage. This method addresses existing abstractive summarization systems on long documents that suffer from generating repeating and redundant words and phrases.

- Fast Abstractive summarization (Chen et al., 2018) proposes an accurate and fast summarization model that first selects salient sentences and then rewrites them abstractively (i.e., compresses and paraphrases) to generate a concise overall summary. It uses a novel sentence-level policy gradient method to bridge the non-differentiable computation between these two neural networks hierarchically while maintaining language fluency.

Table 4.7, Table 4.8, and Table 4.9 show the comparative results of the best performing setting found in the experiments against the aforementioned summarizers based on ROUGE-1, ROUGE-2, on DUC 2001, DUC 2002, and CNN collections respectively. Starting with the DUC 2001 dataset, our systems ($EXABSUM_{Extractive}$ and $EXABSUM_{Hybrid}$) achieve the best performance in terms of both ROUGE-1, ROUGE-2 followed by the DE and FEOM systems (See Figure 4.13).

Table 4.7: F-measure results of our proposed approaches against baselines for the DUC 2001 collection on single-document summarization task

		DUC 2001	
Summary Type	**Summary Type**	ROUGE-1	ROUGE-2
$EXABSUM_{Hybrid}$	Abstractive	**0,484**	**0,211**
$EXABSUM_{Extractive}$	Extractive	**0,480**	**0,208**
DE	Extractive	0,478	0,185
FEOM	Extractive	0,477	0,185
NetSum	Extractive	0,464	0,176
UnifiedRank	Extractive	0,453	0,176
System T (best DUC 2001 participant)	Extractive	0,445	0,202
Classifier4J	Extractive	0,444	0,198
Autosummarizer	Extractive	0,419	0,169
HP-UFPE FS	Extractive	0,359	0,117

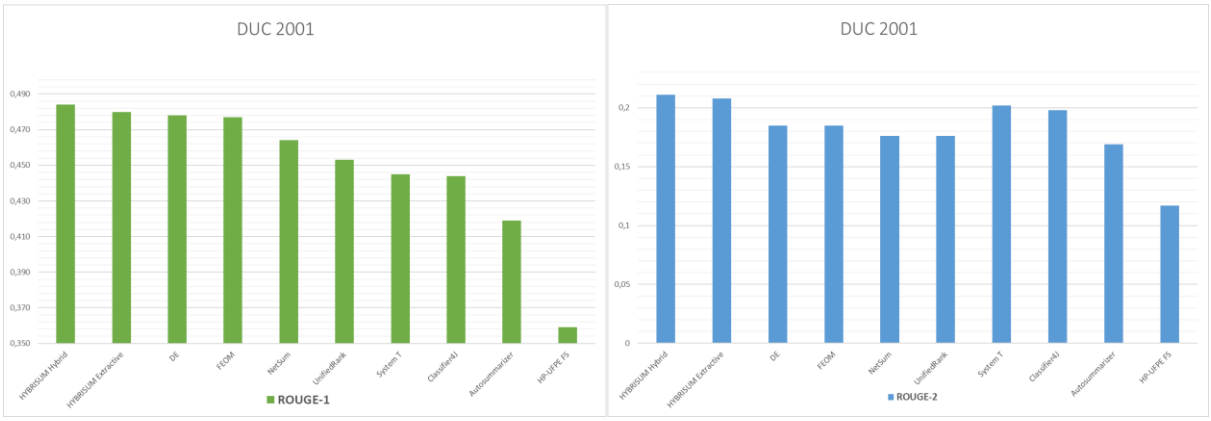


Figure. 4.13 ROUGE-1 and ROUGE-2 results of $EXABSUM_{Extractive}$ and $EXABSUM_{Hybrid}$ against various baseline systems on the DUC 2001 dataset.

After taking a closer look at the feature weights learned by DE and FEOM, we find that both involve semantic features to measure sentence saliency. This may be an important aspect that semantics techniques contribute more to the task of text summarization. In terms of ROUGE-2, System T (best DUC 2001 participant) performed well over other systems. However, it is not statistically different from the one presented by DE, FEOM, and Classifier4J (a supervised approach).

Table 4.8: F-measure results of our proposed approaches against baselines on the DUC 2002 collection on single-document summarization task

		DUC 2002	
Systems	**Summary Type**	**ROUGE-1**	**ROUGE-2**
$EXABSUM_{Hybrid}$	Abstractive	0.497	0.260
$EXABSUM_{Extractive}$	Extractive	0.493	0.257
Parveen and Strube (2015)	Extractive	0,485	0.230
UnifiedRank	Extractive	0,484	0,214
System 28 (best DUC 2002 participant)	Extractive	0,480	0,228
Classifier4J	Extractive	0,470	0,221
DE	Extractive	0,466	0,123
FEOM	Extractive	0,465	0,124
COMPENDIUM E	Extractive	0.456	0.202
NetSum	Extractive	0,449	0,111
Autosummarizer	Extractive	0,437	0,191
COMPENDIUM E–A	Abstractive	0.395	-
Fast Abstractive Summarization	Abstractive	0,394	0,173
Get To The Point	Abstractive	0,372	0,157
HP-UFPE FS	Extractive	0,359	0,117

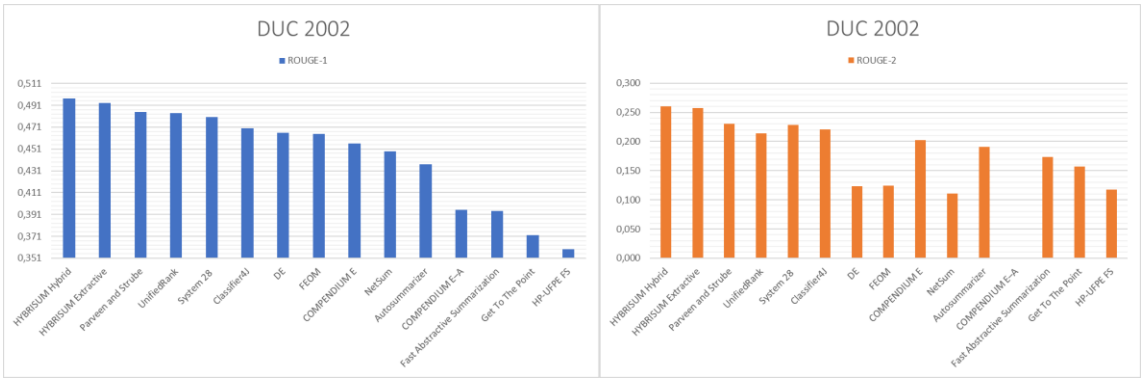


Figure. 4.14 ROUGE-1 and ROUGE-2 results of $EXABSUM_{Extractive}$ and $EXABSUM_{Hybrid}$ against various baseline systems on the DUC 2002 dataset.

On the DUC 2002 dataset, the top-3 performing systems are $EXABSUM_{Extractive}$, $EXABSUM_{Hybrid}$, and Parveen and Strube (2015) respectively (see Figure 4.14). It is worth mentioning that Parveen and Strube (Parveen and Strube, 2015) approach consider summarization as an optimization task, and they use an optimization step to ensure non-redundancy and local coherence in the generated summaries. As expected, including coherence and redundancy removal techniques always improves the results of ATS.

Although a decade has passed since the DUC2001 and DUC2002 competitions were launched, the System T and System 28 still present competitive results against some recent summarizers.

Contrariwise, the 'get to the point' and 'fast abstractive summarization' systems (state-of-the-art abstractive methods) appeared with insignificant results even though they use deep learning approaches.

Table 4.9: F-measure results of our proposed approaches against baselines on the CNN dataset for single-document summarization task

		CNN	
Systems	Summary Type	ROUGE-1	ROUGE-2
$EXABSUM_{Hybrid}$	Abstractive	0.680	0.513
$EXABSUM_{Extractive}$	Extractive	0.601	0.451
HP-UFPE FS	Extractive	0,507	0,345
Autosummarizer	Extractive	0,488	0,327
Classifier4J	Extractive	0,466	0,321

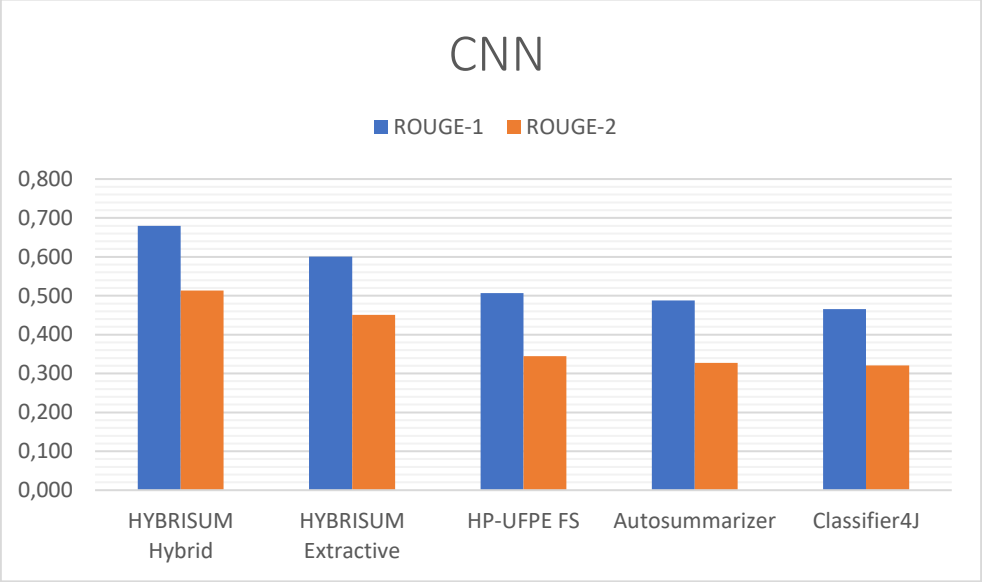


Figure. 4.15 ROUGE-1 and ROUGE-2 results of EXABSUM$_{Extractive}$ and EXABSUM$_{Hybrid}$ against various baseline systems on the large CNN dataset.

Regarding the CNN dataset, again, our system (using both versions) achieves the best results in terms of ROUGE-1 and ROUGE-2 scores (See Figure 4.15). It statistically outperforms all other systems, and an increase of 34.12% is reported compared to the best system.

Overall, our both ATS approaches (EXABSUM$_{Extractive}$ and EXABSUM$_{Hybrid}$) are effective for extractive and abstractive document summarization. They are comparable with respect to other state-of-the-art TS tools (extractive and abstractive). However, the performance of EXABSUM$_{Hybrid}$ is more significant than EXABSUM$_{Extractive}$. While using EXABSUM$_{Extractive}$, it is noteworthy that the obtained ROUGE-1 score is higher than the remaining approaches.

The reason why EXABSUM$_{Hybrid}$ performed better than EXABSUM$_{Extractive}$ for most of the metrics, is that EXABSUM$_{Abstractive}$ process do not rely on the sentences detected as relevant in the sentence relevance stage for building the abstractive summaries, but it relies on the source document. As it was stated, the ROUGE evaluation relies on the exact matches of text snippets while comparing system summary against human-produced summary (extracts). Hence, if EXABSUM$_{Abstractive}$ relies on the extractive summary as input for the abstractive one, the produced abstractive summary will contain lesser matching text snippets. In particular, during the development and experiments of EXABSUM, we followed two strategies: (1) we first built the abstractive summaries by relying on the sentences detected as relevant in the sentence relevance stage, we compressed or merged information within them, and reranked them. Consequently, the resulting summaries were shorter than the extracts (which are generated by EXABSUM$_{Extractive}$), and since no additional information is added, the recall value was always lower regarded to EXABSUM$_{Extractive}$. In the (2) second strategy, we ran EXABSUM$_{Abstractive}$ directly on the source document (starting with the first word of each sentence in the

document) and generate new sentences from it instead of the most relevant sentences, and finally enrich it with the extractive output. Lastly, we chose this last strategy that's appeared more appropriate to generating more accurate summaries.

From the perspective of ROUGE evaluation results presented in Table 4.6, Table 4.7, and Table 4.8, the proposed summarization approach EXABSUM performs well using its two variants: $EXABSUM_{Extractive}$ and $EXABSUM_{Hybrid}$. These results reveal that both kinds of summaries are relevant as they play a significant role in producing informative summaries, contributing to the performance of the TS task. Besides, our proposed TS approach equally contributes to selecting both informative and grammatically correct sentences that are semantically related to the text. Moreover, we can see that the method which further reveals the considerable significance of combining both extractive and abstractive approaches to TS task. The two approaches are complementary, and the overall performance of summarization is improved sufficiently.

4.5. Conclusion

In the first part of this chapter, we have presented our second thesis' examined research fields. The studied review shows that there is still a long trail to walk in the ATS field. First, we have illustrated the classifications and some applications of ATS. Then, we provided, a systematic review of the ATS approaches: extractive, abstractive, and hybrid, along with their general architecture, advantages, and disadvantages. We highlighted the existing techniques that are used to implement ATS systems and discussed some ATS evaluation methods. Based on the described ATS approaches, some conclusions about their tendencies can be drawn for the future. Abstractive paradigms or at least hybrid ones are and will become one of the main challenges to face. Hybrid approaches would be capable of identifying and selecting relevant fragments of information and then merge, compress or delete such information in order to generate new summary information. As a consequence, we must take into account the benefits of extractive and abstractive approaches to move forward on this task.

Thus, in the second part of this chapter, we have proposed EXABSUM, a novel ATS approach for single-document summarization that tackles some of the ATS issues. In particular, EXABSUM produces three types of informative summaries: extractive and abstractive, and hybrid (abstractive oriented) with the corresponding variants of EXABSUM: $EXABSUM_{Extractive}$, $EXABSUM_{Abstractive}$, and $EXABSUM_{Hybrid}$ respectively. In the extractive approach, we have proposed a novel approach to select and extract the most relevant sentences, and detect semantically equivalent redundant sentences as well.

For this manner, we incorporated statistic and semantic scoring techniques and proposed a novel one called 'TR-ISF'. The TR-ISF scoring technique plays a critical role in identifying the salient sentences and has demonstrated its effectiveness overall the ATS system performance. In the $EXABSUM_{Abstractive}$, we proposed a graph-based approach (that includes a compression and fusion stage) and a re-ranking based on the keyphrases approach, which builds abstracts by taking the source document as a starting point instead of the generated extractive summary.

This followed strategy significantly improved the generated summaries. Lastly, by using $EXABSUM_{Hybrid,}$ the produced extractive and abstractive summaries are combined using similarity techniques to generate a final abstractive summary

The comparison of performance measures clearly shows the advantages of our hybrid system that outweigh those of other ATS systems (extractive and abstractive). To determine if EXABSUM is able to generate good summaries and study which approach should be more appropriate to the ATS task, we carried out quantitative and qualitative evaluations on multi-domain benchmark datasets (DUC 2001, DUC 2002, and CNN). The first evaluation aimed to assess the information contained in the summaries, i.e., whether the system was able to identify the relevant information. The second evaluation aimed to demonstrate the quality of the generated abstractive summaries with respect to different criteria, such as fluidity and coherence. According to the obtained quantitative evaluation results, we achieved better ROUGE scores compared to all the other baselines (extractive and abstractive ATS systems). EXABSUM has achieved also good results regarding the qualitative evaluation.

It is worth stressing the fact that EXABSUM modular architecture allows the integration of new stages in an easy way, which can improve the ATS approach as well as extending its functionalities. This means that EXABSUM can be applied later to a wide range of contexts, depending on the user's needs. It generates the most common types of summaries. Consequently, with this proposed ATS design, EXABSUM is capable of capturing not only the important information, but also producing an abstractive oriented summary that is informative, coherent, and different from the extractive one. Moreover, the experimental results verify that our presented $EXABSUM_{Extractive}$ model can acquire effective semantic word-level and sentence-level features, and provide enough information to extract text summarizations. However, it also has some limitations. On the one hand, it does not deal with multilingually, since it is only able to produce summaries in English.

Generally, we can draw three main conclusions: i) the performance of the system for producing summaries improves with the usage of the proposed scoring techniques and the semantic redundancy removal technique, ii) taking as a basis an extractive summary, and enrich it with abstractive information (generated from the source document), influence positively the quality of the produced summaries.

5

Conclusion and Work in Progress

NLP and Text Retrieval are an active area of research for decades. The retrieval of textual information is especially important because the most frequent wanted information is often textual. Several issues have been studied over this entire period, in one hand, the development of models and approaches that estimate the information relevance and keep the retrieval tasks efficient with ever-growing text collections, and on the other hand, create a subset (a summary) that represents the most important or relevant information within the original content. For these purposes, we have seen a diversification of retrieval tasks such as Automatic KeyPhrase Extraction (AKPE) and Automatic Text Summarization (ATS). They started in the late 1950s, and since then, their methodologies and techniques have experienced great advances which make them very important with great interest among the research community. Although new approaches have been developed, there is still a lot of room for improvement. Especially due to the large amounts of available data in different formats and the rapid development of the technology that bring new challenges for these research fields. Indeed, these research fields are very dynamic, since they are continuously adapting themselves to the new needs and challenges. Consequently, all the possibilities that AKPE and ATS offer together with the extensive application it has in the real world make them an interesting research area to conduct research into.

In this Ph.D. thesis, we have investigated and improved AKPE and ATS with a particular focus on hybrid approaches.

This chapter summarizes this thesis and the contributions of our research work and presents the work in progress and for the future. In the next sections, we present the main conclusions of the thesis work (section 5.1), the work in progress and for the future (section 5.2), and the list of relevant publications related to this thesis are provided in (section 5.3).

5.1. Summary of Contributions

The main goal of this thesis was to demonstrate two novel techniques for (1) the generation of keyphrases extracted from textual documents of various types, and (2) the summarization of textual documents of various types. The proposed solutions take into account the specificities of the domains and widely rely on hybrid approaches. Here, we briefly summarize the main results of our work.

In the first part of Chapter 2, we have given a survey and trends on the automatic keyphrase extraction (AKPE) task. The objective was to (i) enhance the understanding of the AKPE process and the differences between the various used approaches and techniques, (ii) provide the guidelines and necessary components to be considered in our future AKPE solution. We concluded that the research works carried out in this area had experienced lately substantial progress which reveals its importance. However, the AKPE task still suffers from some issues that are related to every stage. One of these issues is related to the first step of AKPE: the keyphrases identification. In this step, there is difficulty in dealing with the structure of a keyphrase which can be sometimes irregular and comprised of only a single word or multi-word noun phrase or multiple multi-word noun phrases connected by prepositions. Additionally, most existing systems in this step face different challenging tasks whereby they: (1) fail at predicting candidate keyphrases with different sequential orders, or (2) encounter keyphrases that contain synonyms or abbreviations. Indeed, the AKPE systems go through this step to generate grammatically correct candidates. However, if this step is insufficient, the system will be unable to identify accurate keyphrases in the later steps. Another issue for AKPE, is the feature engineering step. Most traditional AKPE approaches make use of techniques that rely on analyzing just the syntactic aspect of texts, ignoring the meaning they convey and more importantly, the semantic effect of one word over another (context), that underlies the document content. In the last stage of AKPE, several systems used supervised approaches, these approaches depend on the predetermination of features before running the system. They use off-line analysis, and they need a considerable amount of computation time for training. It is not easy to enumerate all the features associated with a specific domain because the linguistic, statistical, and external knowledge of the text should be exploited. Additionally, some keyphrases are positively recognized as candidate keyphrases but fail to be ranked at the top for extraction. Generating statically and semantically relevant keyphrases from any domain of a document is a crucial step too.

Considering these limitations and shortcomings, we have proposed a contribution to this thesis work in order to improve each stage of the AKPE task. Our contribution regarding this task is presented in Chapter 3, in which we have proposed a novel AKPE system. The proposed techniques tackle first the issue of the candidate keyphrases identification step by proposing a novel method based on a parse tree approach, part of speech tagging, and filtering. This approach allowed us successfully to generate a comprehensive and meaningful list of candidate keyphrases and reduced their set size without increasing the computational complexity. Regarding the second and the third stage of AKPE, we suggested a novel hybrid method that takes full advantage of mining simultaneously (linguistic, statistical, structural, and semantic) text features extracted from documents to select the most

relevant keyphrases. Instead of relying solely on syntax analysis, statistical approaches, or annotated corpora, we integrated a technique that uses the semantic representation to extract possible keywords.

This representation is obtained by applying the semantic SIM measure. It considers the context surrounding each word to better represent the meaning underlying the text itself. The proposed hybrid method considered the relevance of keywords within candidates and uses them later to rank keyphrases. To this end, we provided evidence that scoring keyphrases according to their keyword's scores improves the keyphrase extraction performance, and considering the semantic relationships between features improves its accuracy. Our approach of AKPE has demonstrated its effectiveness over more than twelve state-of-the-art and recent unsupervised and
supervised approaches, using five of the top available benchmark corpora from different domains. Moreover, our methodology is efficient even though it does not require any training on a particular set of documents, nor does it depend on external corpora, dictionaries, domain, or text size. Our proposed AKPE technique helps users to extract relevant keyphrases from a document. It will enable fast and accurate search in large text collections, and the produced keyphrases can be used for other applications such as text summarization. Indeed, condensing the extracted important information in the form of summary would be very beneficial for several readers. Particularly, without summaries it would be practically hard for human beings to get access to the ever-growing mass of information available online.

In the second part of chapter 2, we have provided a review about IR systems' evolution from the classic to the contextual scope. We have presented the concept of context, the variety of its categories, levels, and uses. We also showed how to model the context in the IR. Although many works focused on contextual IR systems, some related significant challenges are still remaining. For this reason, we discussed some context IR modeling issues, and proposed some practical recommendations to enhance these issues for future research.

Due to the difficulty of manual text summarization of the huge amount of the textual content on the Internet or various archives, ATS systems have appeared as the main technology to solve this urgent and pressing issue. After conducting a review about this fast evolutive task, we found that there are many issues and tendencies that should be addressed. First, as can be concluded, trying to compose a good summary should satisfy the readability and linguistic quality of the summary, the consistency and content coverage of the summary, and the non-redundancy of the produced summary. Second, we noticed that many existing extractive approaches do not consider the semantic context among the textual units (words, phrases, sentences, text, corpus), and they are still suffering from the semantic redundancy too. Therefore, capturing relevant information in a document to be summarized and coherently presenting it by these systems is still a complex task. Third, there are continuous research efforts to overcome these difficulties and investigate new solutions. Over time, the scientific community mainly has focused on the extractive text summarization approach. It has implemented the summarization methods of this approach for various types of applications such as user reviews, news articles, blogs, email messages, scientific articles, legal documents, biomedical documents, etc. In practice, extractive ATS systems produce verbatim summaries. However, we concluded that recent

research works on this task have and still experienced a huge interest and progress, especially for the abstractive and hybrid ATS approaches. Abstractive paradigms or hybrid ones are and will become one of the main challenges to face for the next years. Hybrid approaches would be capable of identifying and selecting relevant fragments of information and then merge, compress or delete such information in order to generate new summary information. As a consequence, it is possible to take into account the benefits of extractive and abstractive approaches together. However, contributions in hybrid approaches are in progress daily.

Considering the investigated issues related to the ATS TASK, we have presented in chapter 4, a novel hybrid ATS approach. In this chapter, we detailed our third contribution. In particular, the proposed unsupervised hybrid ATS approach produces two types of informative summaries: extractive and abstractive, with the corresponding variants of EXABSUM: $EXABSUM_{Extractive}$ and $EXABSUM_{Abstractive}$, respectively.

Regarding the extractive approach, we have proposed novel scoring techniques that select and extract the most relevant sentences. They determine important information in documents and were shown appropriately for the generation of extractive summaries. This approach addresses the issue of redundancy as well as detecting and discarding semantically equivalent redundant sentences. In the abstractive approach, we have proposed a graph-based technique that involves compression and fusion methods and are-ranking based on keyphrases step. Both approaches significantly improved the informativity of the generated summaries. Lastly, to take advantage of

the hybrid paradigm, we combined the produced extractive and abstractive summaries using similarity techniques to generate the final abstractive summary. The comparison of performance measures clearly showed that the advantages of the proposed hybrid system outperform those of other evaluated ATS systems (extractive and abstractive). The proposed approach was evaluated under different settings and multi-domain benchmark datasets to ensure its utility in the case of generating extractive and abstractive summaries. In particular, these evaluations have determined that EXABSUM is able to generate good summaries and showed which approach (extractive or abstractive) is more appropriate. Generally, it is worth stressing upon the fact that EXABSUM modular architecture allows the integration of new stages easily, which improve the ATS approach as well as extending its functionalities. This means that EXABSUM can be applied later to a wide range of contexts, depending on the user's needs. However, it has also some limitations. For instance, it does not deal with multilingually, since it is only able to produce summaries in English. Thus, we–draw three main conclusions from our last contribution: i) the performance of the system for producing summaries improves with the usage of semantic redundancy removal, ii) taking as a basis an extractive summary, and enrich it with abstractive information generated from the source document, influence positively the resulting summaries. iii) EXABSUM as an ATS system using our proposed strategy for facing abstractive summarization is appropriate.

Even though abstractive summarization in a real sense is a hard task to achieve, our approach demonstrates this new track's viability for the summarization community. This is a very positive result since it indicates that we can carry out research into this type of summaries, improve the quality, and go beyond the simple selection of relevant sentences.

5.2. Work in Progress and for the Future

The promising results found out in this thesis work open several ways to further developments on the integration of AKPE and ATS in IR and NLP fields.

The AKPE task generates several possible future research directions explained as follows:

- In the future, In the candidates keyphrase selection step, adding more patterns is worth considering, such as more complex noun phrase patterns or other types of phrase patterns. More patterns mean more varieties of phrases can be extracted and could achieve better results. We plan also to work on a new version of HAKE, by using word embeddings approach to our algorithm.

- We intend to perform our AKPE approach on real world domain such as the medical one. Due to the complexity of natural language itself, future work on this domain can explore terms by using a thesaurus and abbreviations, and medical terminology.

- Relevance feedback is a useful technique for improving the effectiveness of an IR system. The idea underlying relevance feedback is simple: it uses responses from the user to mark retrieved documents as relevant or nonrelevant. When the user provides the IR system with a query, the system retrieves documents from the collection based on the query. We believe the idea of relevance feedback can be used to improve the performance of a keyphrase extraction task. We could ask the user to assess the quality of the output keyphrases in a trial run. The information provided could be used to tune the algorithm by adjusting the weight of different attributes used in the AKPE algorithm.

- The user’s interest is a common assumption in information retrieval. Instead of always getting entire documents, users want to see directly the relevant parts of long articles. In future works, we intend to enhance the document retrieval to a context entity retrieval. Entity retrieval sets the focus level of retrieval performance. It allows to search and rank named entities included in any kind of text sources.

- One shortcoming of the research project is that outputs are not fully evaluated, since they are evaluated using exact matching. Traditionally, most AKPE systems have been evaluated using the proportion of *Top-N* candidates that exactly match the gold standard keyphrases (Frank et al. 1999; Witten et al. 1999; Turney 2000). However, in many papers, the authors assign manually keyphrases based on their semantic meaning, instead of following the written content in the document. Consequently, current evaluation methods do not cover keyphrases that are semantically relevant. In that case, the exact matching of keyphrases can be very strict and problematic because it ignores near matches that are largely semantically identical. The evaluation of exact matches is deemed insufficient to cover all relevant keyphrases. We are analyzing semantical-motivated evaluation to give more accurate keyphrase suitability.

For the short and long term, we will continue to carry out research into ATS techniques and we will focus on the application of abstractive summaries in real-world domains:

- Currently, we are working on exploring the performance of the EXABSUM on multi-document summarization, and in other languages such as Arabic and Spanish.

- Working on new EXABSUM variants: we are working on two additional stages in order to increase the capabilities of EXABSUM, allowing it to generate query-focused, sentiment-based (using subjective information detection) summaries. Query-focused summaries are based on a user need, question, or topic; and sentiment-based summaries contain a high degree of subjective information, reflecting the opinions of users about a topic. Since semantic knowledge allows a higher level of abstraction. enriching EXABSUM with external semantic knowledge will be useful. For carrying out this, we intend to use external semantic resources, such as WordNet. The concepts of the document will be mapped to the concepts of WordNet and then, their hypernyms are taken into account to generalize several related concepts into the same. This line of research extends the research carried out into abstractive-oriented summarization.

- In future work, we plan also to analyze other variants for generating more abstractive summaries, such as rephrasing techniques. This may lead to improve performance of the used models. Deep learning techniques were employed in abstractive text summarization for the first time in 2015, and the proposed model was based on the encoder-decoder architecture. We plan to test our hybrid model with Deep Learning techniques and Distributed Representation (Word Embedding).

- We intend in future works to study the integration of ATS techniques into a recommender system and opinion information retrieval application. Currently, we are analyzing these integrations to build a unified framework that deals with new textual genres (blogs, reviews, etc.).

The proposed approach will be capable of retrieving subjective content from the Web (for instance, blogs, e-commerce reviews, etc.), analyzing and classifying the opinions found in them, and finally producing a summary that contains the specific user' information need.

This thesis constitutes only a small part of the IR research area, which are the AKPE and ATS tasks. Therefore, there are still some issues that are being currently tackled, and some aspects that will be addressed in the future. As it was presented in the second part of Chapter 2, sometimes, most of the retrieved information from the Information Retrieval (IR) System is irrelevant to the user because the IR cannot determine the user's context. Now and then, some elements should be considered to accrue more relevant and adapted information to the user's information need (as represented by system queries). Elements such as time, location, and past history (including relevance feedback) of information access can also shape the information needs of users (as represented by system queries). Information consumers have varying roles, missions/tasks, knowledge and background, goals and agendas, and personal preferences. These factors can be used to shape both the execution of user queries and the form in which retrieved information is packaged. However, full automation of this daunting information aggregation and customization task is not an easy task with existing approaches. Therefore, in future work, we intend to make use of our contributions in AKPE and ATS to build a unified context-aware information retrieval infrastructure. This infrastructure will enhance situational awareness and provides each user with a customized, mission-oriented system that gives access to the right information from heterogeneous sources in the context of a particular task, plan, and/or mission. Several intertwined concepts can be used to achieve this approach, such as Workflow, Context, Profile and Information Aggregation, and Ontologies. The exploitation of this knowledge, using appropriate domain ontologies, will make it feasible to provide contextual support in various ways to the work performed according to a user's task-relevant information requirements.

5.3 Relevant Publications

The research contributions carried out for the completion of this thesis work have resulted in several publications in international conferences and journals. We present these publications next, sorting them by publication type.

International Journals Articles

- **Merrouni, Z. A., Frikh, B., & Ouhbi, B. (2020).** Automatic keyphrase extraction: a survey and trends. Journal of Intelligent Information Systems, 1-34. SPRINGER DOI: **https://doi.org/10.1007/s10844019-00558-9**
- **MERROUNI, Z. A., FRIKH, B., & OUHBI, B. (2019).** Toward Contextual Information Retrieval: A Review and Trends**. Procedia computer science, 148, 191-200. DOI: 10.1016/j.procs.2019.01.036**
- **MERROUNI, Z. A., FRIKH, B., & OUHBI, B.** HAKE: An unsupervised approach for Automatic Keyphrase Extraction". Cognitive Computation Journal, SPRINGER. (Under Review)
- **MERROUNI, Z. A., FRIKH, B., & OUHBI, B.** EXABSUM: A new Text Summarization Approach for Generating Extractive and Abstractive Summaries. Neural Computing and Applications, SPRINGER. 2021 (accepted, preprint).

International Conferences Articles:

- **MERROUNI, Z. A., B. FRIKH, B. OUHBI**. Automatic Keyphrase Extraction: An Overview of The State of The Art», *4th IEEE International Conference on Information Science and Technology (IEEE CiSt'16)**, 24-26 October 2016.***
- **MERROUNI, Z. A., B. FRIKH, B. OUHBI**. Toward Contextual Information Retrieval: A Review and Trends », *The second International Conference on Intelligent Computing in Data Sciences ICDS2018**,* **03-04-05, *October 2018.***

Bibliography

Barker, K., & Cornacchia, N. (2000). Using noun phrase heads to extract document keyphrases. In: conference of the canadian society for computational studies of intelligence, pp. 40–52. Springer.

Berend, G. (2011). Opinion expression mining by exploiting keyphrase extraction. In: Proceedings of the 5th international joint conference on natural language processing. Asian Federation of Natural Language Processing.

Berend, G., & Farkas, R. (2010). SZTERGAK: Feature engineering for keyphrase extraction. In: proceedings of the 5th international workshop on semantic evaluation, pp. 186–189. Association for Computational Linguistics.

Blei, D.M., Ng, A.Y., Jordan, M.I. (2003). Latent dirichlet allocation. Journal of Machine Learning Research, 3(Jan), 993–1022.

Bougouin, A., Boudin, F., Daille, B. (2013). TOPICRANK: Graph-based topic ranking for keyphrase extraction. In: International joint conference on natural language processing (IJCNLP), pp. 543– 551.

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.

Bulgarov, F., & Caragea, C. (2015). A comparison of supervised keyphrase extraction models. In: Proceed-ings of the 24th international conference on World Wide Web, pp. 13–14. ACM.

Chandrasekar, R., James, C.F.I., Watson, E.B. (2006). System and method for query refinement to enable improved searching based on identifying and utilizing popular concepts related to users' queries. US Patent, 7, 136,845.

Chen, M., Sun, J.T., Zeng, H.J., Lam, K.Y. (2005). A practical system of keyphrase extraction for web pages. In: Proceedings of the 14th ACM international conference on information and knowledge management, pp. 277–278. ACM.

Cho, T., & Lee, J.H. (2015). Latent keyphrase extraction using LDA model. Journal of Korean Institute of Intelligent Systems, 25(2), 180–185.

Danesh, S., Sumner, T., Martin, J.H. (2015). SGRANK: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. In: Proceedings of the fourth joint conference on lexical and computational semantics, pp. 117–126.

D'Avanzo, E., & Magnini, B. (2005). A keyphrase-based approach to summarization: The LAKE system at DUC-2005. In: Proceedings of DUC

Do, N., & Ho, L. (2015). Domain-specific keyphrase extraction and near-duplicate article detection based on ontology. In: International conference on computing & communication technologies, research, innovation, and vision for the future (RIVF), pp. 123–126. IEEE.

Dostal, M., & Je˙zek, K. (2011). Automatic keyphrase extraction based on NLP and statistical method. In: Dateso Conference. Zapado´cesk˙a´ Univerzita v Plzni.

El-Beltagy, S.R., & Rafea, A. (2009). KP-MINER: A keyphrase extraction system for English and Arabic documents. Information Systems, 34(1), 132–144.

El Idrissi, O., Frikh, B., Ouhbi, B. (2014). HCHIRSIMEX: An extended method for domain ontology learning based on conditional mutual information. In: 3rd IEEE international colloquium in information science and technology (CIST), pp. 91–95. IEEE.

Elman, J.L. (1990). Finding structure in time. Cognitive science, 14(2), 179–211.

Elovici, Y., Shapira, B., Last, M., Zaafrany, O., Friedman, M., Schneider, M., Kandel, A. (2010). Detection of access to terror-related web sites using an advanced terror detection system (ATDS). Journal of the association for information science and technology, 61(2), 405–418.

Ferrara, F., Pudota, N., Tasso, C. (2011). A keyphrase-based paper recommender system. In: Italian research conference on digital libraries, pp. 14–25. Springer.

Fortuna, B., Grobelnik, M., Mladenic,´ D. (2006). Semi-automatic data-driven ontology construction system. In: Proceedings of the 9th international multi-conference information society, pp. 223–226.

Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G. (1999). Domain-specific keyphrase extraction. In Proceedings of the 16th international joint conference on artificial intelligence, IJCAI ’99 (pp. 668–673). San Francisco: Morgan Kaufmann Publishers Inc. http://dl.acm.org/citation.cfm? id=646307.687591.

Frantzi, K.T., Ananiadou, S., Tsujii, J. (1998). The C-VALUE/NC-VALUE method of automatic recognition for multi-word terms. In: International conference on theory and practice of digital libraries, pp. 585–604. Springer.

Frikh, B., Djaanfar, A.S., Ouhbi, B. (2011). A new methodology for domain ontology construction from the Web. International Journal on Artificial Intelligence Tools, 20(06), 1157–1170.

Gollapalli, S.D., & Caragea, C. (2014). Extracting keyphrases from research papers using citation networks. In: AAAI, pp. 1629–1635.

Gong, Z., & Liu, Q. (2009). Improving keyword-based web image search with visual feature distribution and term expansion. Knowledge and Information Systems, 21(1), 113–132.

Grineva, M., Grinev, M., Lizorkin, D. (2009). Extracting key terms from noisy and multitheme documents. In: Proceedings of the 18th international conference on World Wide Web, pp. 661–670. ACM.

Gutwin, C., Paynter, G., Witten, I., Nevill-Manning, C., Frank, E. (1999). Improving browsing in digital libraries with keyphrase indexes. Decision Support Systems, 27(1-2), 81–104.

Haddoud, M. (2014). Abdedda¨ım, S.: Accurate keyphrase extraction by discriminating overlapping phrases. Journal of Information Science, 40(4), 488–500.
Haddoud, M., Mokhtari, A., Lecroq, T. (2015). Abdedda¨ım, S.: Accurate keyphrase extraction from scientific papers by mining linguistic information. In: CLBib@ ISSI, pp. 12–17.

Hammouda, K.M., & Kamel, M.S. (2002). Phrase-based document similarity based on an index graph model. In: Proceedings of international conference on data mining (ICDM), pp. 203–210. IEEE.

Hammouda, K.M., Matute, D.N., Kamel, M.S. (2005). COREPHRASE: Keyphrase extraction for document clustering. In: International workshop on machine learning and data mining in pattern recognition, pp. 265–274. Springer.

Han, J., Kim, T., Choi, J. (2007). Web document clustering by using automatic keyphrase extraction. In: 2007 IEEE/WIC/ACM international conferences on web intelligence and intelligent agent technology - workshops, pp. 56–59. IEEE.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In: Proceedings of the fifteenth conference on uncertainty in artificial intelligence, pp. 289–296. Morgan Kaufmann Publishers Inc.

Huang, C., Tian, Y., Zhou, Z., Ling, C.X., Huang, T. (2006). Keyphrase extraction using semantic network's structure analysis. In: 6th international conference on data mining (ICDM'06), pp. 275–284. IEEE.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 conference on empirical methods in natural language processing, pp. 216–223. Association for Computational Linguistics.

Hulth, A., & Megyesi, B.B. (2006). A study on automatically extracted keywords in text categorization. In: Proceedings of the 21st international conference on computational linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 537–544. Association for Computational Linguistics.

Jarmasz, M., & Barriere, C. (2004). Using semantic similarity over tera-byte corpus, compute the performance of keyphrase extraction. Proceedings of CLINE.

Jiang, X., Hu, Y., Li, H. (2009). A ranking approach to keyphrase extraction. In Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval, SIGIR '09 (pp. 756–757). New York: ACM. https://doi.org/10.1145/1571941.1572113.

Jones, S., & Staveley, M.S. (1999). PHRASIER: A system for interactive document retrieval using keyphrases. In: Proceedings of the 22nd annual international ACM SIGIR conference on research and development in information retrieval, pp. 160–167. ACM.

Jungiewicz, M., & Łopuszynski,´ M. (2014). Unsupervised keyword extraction from Polish legal texts. In: International conference on natural language processing, pp. 65–70. Springer.

Kamal Sarkar Mita Nasipuri, S.G. (2010). A new approach to keyphrase extraction using neural networks. International Journal of Computer Science Issues online.IJCSI, Volume 7, Issue 2, March 2010.

Kelleher, D., & Luz, S. (2005). Automatic hypertext keyphrase detection. In: IJCAI, vol. 5, pp. 1608–1609.

Kim, S.N., & Kan, M.Y. (2009). Re-examining automatic keyphrase extraction approaches in scientific articles. In: Proceedings of the workshop on multiword expressions: identification, interpretation, disambiguation and applications, pp. 9–16. Association for Computational Linguistics.

Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T. (2010). SEMEVAL-2010 Task 5: Automatic keyphrase extraction from scientific articles. In: Proceedings of the 5th international workshop on semantic evaluation, pp. 21–26. Association for Computational Linguistics

Krovetz, R., & Croft, W.B. (1992). Lexical ambiguity and information retrieval. ACM Transactions on Information Systems (TOIS), 10(2), 115–141.

Kumar, N., & Srinathan, K. (2008). Automatic keyphrase extraction from scientific documents using n-gram filtration technique. In: Proceedings of the eighth ACM symposium on document engineering, pp. 199–208. ACM.

Landauer, T.K., Foltz, P.W., Laham, D. (1998). An introduction to latent semantic analysis. Discourse processes, 25(2-3), 259–284.

Leake, D.B., Maguitman, A., Reichherzer, T., Canas,˜ A.J., Carvalho, M., Arguedas, M., Brenes, S., Eskridge, T. (2003). Aiding knowledge capture by searching for extensions of knowledge models. In: Proceedings of the 2nd international conference on knowledge capture, pp. 44–53. ACM.

LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. Nature, 521(7553), 436.

Liu, F., Pennell, D., Liu, F., Liu, Y. (2009). Unsupervised approaches for automatic keyword extraction using meeting transcripts. In: Proceedings of human language technologies: the 2009 annual conference of the North American chapter of the Association for Computational Linguistics, pp. 620–628. Association for Computational Linguistics.

Liu, W., Chung, B.C., Wang, R., Ng, J., Morlet, N. (2015). A genetic algorithm enabled ensemble for unsupervised medical term extraction from clinical letters. Health Information Science and Systems, 3(1), 5.

Liu, Z., Huang, W., Zheng, Y., Sun, M. (2010). Automatic keyphrase extraction via topic decomposition. In: Proceedings of the 2010 conference on empirical methods in natural language processing, pp. 366–376. Association for Computational Linguistics.

Liu, Z., Li, P., Zheng, Y., Sun, M. (2009). Clustering to find exemplar terms for keyphrase extraction. In: Proceedings of the 2009 conference on empirical methods in natural language processing: vol. 1, pp. 257–266. Association for Computational Linguistics.

Lopez, P., & Romary, L. (2010). HUMB: Automatic key term extraction from scientific articles in GROBID. In: Proceedings of the 5th international workshop on semantic evaluation, pp. 248–251. Association for Computational Linguistics.

Lops, P., De Gemmis, M., Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In: Recommender Systems Handbook, pp. 73–105. Springer.

Matsuo, Y., & Ishizuka, M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools, 13(01), 157–169.

Matsuo, Y., Mori, J., Hamasaki, M., Nishimura, T., Takeda, H., Hasida, K., Ishizuka, M. (2007). POLY-PHONET: An advanced social network extraction system from the web. Web Semantics: Science. Services and Agents on the World Wide Web, 5(4), 262–278.

Medelyan, O., Frank, E., Witten, I.H. (2009). Human-competitive tagging using automatic keyphrase extraction. In: Proceedings of the 2009 conference on empirical methods in natural language processing, vol. 3, pp. 1318–1327. Association for Computational Linguistics.

Medelyan, O., & Witten, I.H. (2006). Thesaurus based automatic keyphrase indexing. In: Proceedings of the 6th ACM/IEEE-CS joint conference on digital libraries, pp. 296–297. ACM.

Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., & Chi, Y. (2017, July). Deep Keyphrase Generation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 582-592).

Mihalcea, R., & Tarau, P. (2004). TEXTRANK: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing.

Mihalcea, R., Tarau, P., Figa, E. (2004). PageRank on semantic networks, with application to word sense disambiguation. In: Proceedings of the 20th international conference on computational linguistics, p. 1126. Association for Computational Linguistics.

Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum, R., Rus, V. (2000). The structure and performance of an open-domain question answering system. In: Proceedings of the 38th annual meeting on Association for Computational Linguistics, pp. 563–570. Association for Computational Linguistics.

Mori, J., Ishizuka, M., Matsuo, Y. (2007). Extracting keyphrases to represent relations in social networks from web. In: IJCAI, vol. 7, pp. 2820–2827.

Newman, D., Koilada, N., Lau, J.H., Baldwin, T. (2012). Bayesian text segmentation for index term identification and keyphrase extraction. Proceedings of COLING, 2012, 2077–2092.

Nguyen, T.D., & Kan, M.Y. (2007). Keyphrase extraction in scientific publications. In: International conference on asian digital libraries, pp. 317–326. Springer.

Nguyen, T.D., & Luong, M.T. (2010). WINGNUS: Keyphrase extraction utilizing document logical structure. In: Proceedings of the 5th international workshop on semantic evaluation, pp. 166–169. Association for Computational Linguistics.

Osinski,S., Stefanowski, J., Weiss, D. (2004). LINGO: Search results clustering algorithm based on singular value decomposition. In: Intelligent information processing and web mining, pp. 359–368. Springer.

Page, L., Brin, S., Motwani, R., Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web, Stanford InfoLab, Tech. rep.

Sarkar, K. (2013). A Hybrid Approach to Extract Keyphrases from Medical Documents. International Journal of Computer Applications, 63(18), 14-19.

Smatana, M., & Butka, P. (2016). Extraction of keyphrases from single document based on hierarchical concepts. In: IEE 14th international symposium on applied machine intelligence and informatics (SAMI), pp. 93–98. IEEE.

Song, M., Song, I.Y., Allen, R.B., Obradovic, Z. (2006). Keyphrase extraction-based query expansion in digital libraries. In: Proceedings of the 6th ACM/IEEE-CS joint conference on digital libraries, pp. 202–209. ACM.

Tomokiyo, T., & Hurst, M. (2003). A language model approach to keyphrase extraction. In: Proceedings of the ACL 2003 workshop on multiword expressions: analysis, acquisition and treatment-volume 18, pp. 33–40. Association for Computational Linguistics.

Turney, P.D. (2000). Learning algorithms for keyphrase extraction. Information Retrieval, 2(4), 303–336.

Turney, P. D. (2003). Coherent keyphrase extraction via web mining. In Proceedings of the 18th international joint conference on Artificial intelligence (pp. 434-439).

Wan, X., & Xiao, J. (2008). Single document keyphrase extraction using neighborhood knowledge. In: AAAI, vol. 8, pp. 855–860.

Wan, X., Yang, J., Xiao, J. (2007). Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In: Proceedings of the 45th annual meeting of the association of computational linguistics, pp. 552–559.

Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G. (1999). KEA: Practical automatic keyphrase extraction. In: Proceedings of the fourth ACM conference on digital libraries, pp. 254–255. ACM.

Xie, F., Wu, X., Zhu, X. (2017). Efficient sequential pattern mining with wildcards for keyphrase extraction. Knowledge-Based Systems, 115, 27–39.

Yang, S., Lu, W., Yang, D., Li, X., Wu, C., Wei, B. (2017). KEYPHRASEDS: Automatic generation of survey by exploiting keyphrase information. Neurocomputing, 224, 58–70.

Yih, W.T., Goodman, J., Carvalho, V.R. (2006). Finding advertising keywords on web pages. In Proceedings of the 15th international conference on World Wide Web, WWW '06 (pp. 213–222). New York: ACM. https://doi.org/10.1145/1135777.1135813.

You, W., Fontaine, D., Barthes, J.P. (2009). Automatic keyphrase extraction with a refined candidate set. In: Proceedings of the 2009 IEE/WIC/ACM International joint conference on web intelligence and intelligent agent technology-volume 01, pp. 576–579. IEEE Computer Society.

Zamir, O., & Etzioni, O. (1998). Web document clustering: A feasibility demonstration. In: SIGIR, vol. 98, pp. 46–54. Citeseer.

Zesch, T., & Gurevych, I. (2009). Approximate matching for evaluating keyphrase extraction. In: Proceedings of the international conference ranLP, pp. 484–489.

Zha, H. (2002). Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In: Proceedings of the 25th annual international acm sigir conference on research and development in information retrieval, pp. 113–120. ACM.

Zhang, D., & Dong, Y. (2004). Semantic, hierarchical, online clustering of web search results. In: Asia-Pacific Web Conference, pp. 69–78. Springer.

Zhang, K., Xu, H., Tang, J., Li, J. (2006). Keyword extraction using support vector machine. In: international conference on web-age information management, pp. 85–96. Springer.

Zhang, Q., Wang, Y., Gong, Y., Huang, X. (2016). Keyphrase extraction using deep recurrent neural networks on Twitter. In: Proceedings of the 2016 conference on empirical methods in natural language processing, pp. 836–845.

Zhang, Y., Zincir-Heywood, N., Milios, E. (2004). World Wide Web site summarization. Web intelligence and agent systems: an international journal, 2(1), 39–53.

Tseng, E., & Braginsky, D. (2018). "Providing context relevant search for a user based on location and social information", U.S. Patent No. 9,978,022. Washington, DC: U.S. Patent and Trademark Office.

Sarwar, S. M., Foley, J., & Allan, J. (2018, March). Term relevance feedback for contextual named entity retrieval. In Proceedings of the 2018 Conference on Human Information Interaction & Retrieval (pp. 301-304).

Jones, T. C., Robinson, K., & Steel, B. C. (2018). "Context retrieval and description benefits for recognition of unfamiliar faces". Journal of experimental psychology. Learning, memory, and cognition.

Lindsay, R. T., & Rajaram, G. (2017). "Determining phrase objects based on received user input context information.", U.S. Patent No. 9,619,046. Washington, DC: U.S. Patent and Trademark Office.

Flake, G., Gates III, W. H., Horvitz, E. J., Goodman, J. T., Chaudhuri, S. J., Griffin, T. J., ... & Moss, K. A. (2017). “Search guided by location and context”, U.S. Patent No. 9,536,004. Washington, DC: U.S. Patent and Trademark Office.

O'donnell, S. (2017). “Context based data searching” U.S. Patent No. 9,767,164. Washington, DC: U.S. Patent and Trademark Office.

(Ingwersen and Järvelin, 2017) Ingwersen, P., & Järvelin, K. (2006). “The turn: Integration of information seeking and retrieval in context” (Vol. 18). Springer Science & Business Media.

Allen, B. (1997). “Information needs: a person–situation approach”. In P.Vakkari, R.Savolainen, & B.Dervin (Eds.), Information seeking in context: Proceedings of an international conference on research in information needs, seeking and use in different contexts (pp.111–122).London: Taylor Graham.

Vakkari, P. (1997). “Information seeking in context: a challenging metatheory”. In P.Vakkari, R.Savolainen, & B.Dervin (Eds.), Information seeking in context: Proceedings of an international conference on research in information needs, seeking and use in different contexts (pp.451–464).London: Taylor Graham.

Sonnenwald, D.H. (1999). “Evolving perspectives of human behavior: contexts, situation, social networks and information horizons”. In T.Wilson, & D.K.Allen (Eds.), Exploring the contexts of information behaviour: Proceedings of the 2nd international conference on research in information needs, seeking and use in different contexts (pp.176–190).London: Taylor Graham.

Bharat, K. (2000). “SearchPad: Explicit capture of search context to support web search”. Computer Networks, 33(1-6), 493-501.

McCarthy, J. (1993), “Notes on Formalizing Context”, in Proc. of the 13th International Joint Conference on Artificial Intelligence (IJCAI 1993), Chambery, France, pp. 81-98.

Edmonds, B., (1999) “The Pragmatic Roots of Context”, in Proc. of the 2nd International and Interdisciplinary Conference on Modeling and Using Context, Trento, Italy, Springer Verlag LNAI vol. 1688, pp. 119-132.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E., (2002) “Placing Search in Context: The Concept Revisited”, ACM Transactions on Information Systems vol.20, no. 1, pp. 116-131.
Haveliwala (2002) Haveliwala, T. H., (2002), “Topic-Sensitive PageRank”, in Proc. of the 11th International World Wide Web Conference (WWW 2002), Honolulu, Hawaii, USA, pp. 517-526.

Lawrence, S., (2000), “Context in Web Search”, IEEE Data Engineering Bulletin vol. 23, no. 3, pp. 25-32.
(Dasiopoulou et al. 2005) Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V. K., and Strintzis, M. G., (2005) “Knowledge assisted semantic video object detection”, IEEE Transactions on Circuits and Systems for Video Technology vol. 15, no. 10, pp. 1210-1224.

Mylonas, Ph. and Avrithis, Y., (2005) “Context modeling for multimedia analysis and use”, in Proc. of the 5th International and Interdisciplinary Conference on Modeling and Using Context (Context 2005), Paris, France.

Brown, P. J., Bovey, J., and Chen, X., (1997) “Context-Aware Applications: From the Laboratory to the Marketplace”, IEEE Personal Communications vol. 4, no. 5, pp. 58-64.

Coutaz, J., Crowley, J., Dobson, S., and Garlan, D., (2005) “Context is key”, Communications of the ACM vol. 48, no. 3, pp. 49-53.

Heer, J., Newberger, A., Beckmann, C., and Hong, J., (2003) “liquid: Context-aware distributed queries”, in Proc. of the 5th International Conference on Ubiquitous Computing (UbiComp 2003), Seattle, Washington, USA, pp. 140-148.

Hong, J. I. and Landay, J. A., (2001) “An infrastructure approach to context-aware computing”, Human Computer Interaction vol. 16, no. 2-4, 2001, pp. 87-96.

van Eijck, J., (2000) “On the proper treatment of context in NL”, In Monachesi, P. (Ed.), Computational Linguistics in the Netherlands 1999, Selected Papers form the 10th CLIN Meeting, Utrecht, The Netherlands.

Wiebe, J., Hirst, G., and Horton, D., (1996) “Language Use in Context”, Communications of the ACM vol.39, no. 1, pp. 102-111.

Shen, X., Tan, B., and Zhai, C., (2005) “Context-sensitive information retrieval using implicit feedback”, in Proc. of the 28th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR 2005), Salvador, Brazil, pp. 43-50.

S. Lawrence. (2000) “Context in Web Search”. Data Engineering, IEEE Computer Society, Vol. 23, No. 3, pp. 25-32.

Collins. (1987). “Collins Cobuild English Language Dictionary”. London: William Collins.

Dey, A. K. & Abowd, G. D. (2000). ‘Towards a Better Understanding of Context and Context-Awareness”, CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness.

Dourish, P. (2004). “What we talk about when we talk about context”, Personal Ubiquitous Comput. 8(1): 19–30.

Sato, K. (2001). “Context sensitive interactive systems design: a framework for representations of contexts”, Hum. Comput. Interact., Vol. 2, pp. 229–241.

Bierig, R. & Goker, A. (2006). “Time, location and interest: an empirical and user-centred study”, In Proceedings of the 1st international conference on Information interaction in context (pp. 79-87). ACM.

I. Ruthven, (2011) “Information retrieval in context,” in Advanced Topics in Information Retrieval, chapter 8, (M. Melucci and R. Baeza-Yates, eds.), Springer.

Allan J (2002), “Challenges in information retrieval and language modelling”. In: Report of a workshop held at the Center for Intelligent Information Retrieval, University of Massachusetts, Amherst.

Crestani F. and Ruthven, I. (2007). “Introduction to special issue on contextual information retrieval systems”. Journal of Information Retrieval, 10 (2):111–113.

Allan J, (2002) “Challenges in information retrieval and language modelling”. In: Report of a workshop held at the Center for Intelligent Information Retrieval, University of Massachusetts, Amherst

Ingwersen, P. (2006). “Context in information interaction – revisited 2006”. Proceedings of the Fourth Biennial DISSAnet Conference. Farm Inn, Pretoria, Infuse: 13-23.

Goker, A. & Myrhaug, H. I. (2002). “User context and personalization”, ¨ ECCBR Workshop on Case Based Reasoning and Personalisation.

Mansourian, Y. (2008). “Contextualization of web searching: a grounded theory approach”, The Electronic Library 26(2): 202–214.

Kofod-Petersen, A., & Aamodt, A. (2003). “Case-based situation assessment in a mobile context-aware system”. In Proceedings of AIMS2003, Workshop on Artificial Intelligence for Mobil Systems, Seattle.

Tamine, L., Boughanem M., and Daoud M. (2009). “Evaluation of contextual information retrieval effectiveness: overview of issues and research”. Knowledge and Information Systems, 24(1):1–34.

Anand SS, Mobasher B (2007) “Introduction to intelligent techniques for web personalization”. ACM Trans Internet Technol 7(4):18.

Melucci, M. (2012). “Contextual search: A computational framework”. Foundations and Trends® in Information Retrieval, 6(4–5), 257-405. http://dx.doi.org/10.1561/1500000023.

C. da Costa Pereira, M. Dragoni, G. Pasi, (2009) “Multidimensional Relevance: A New Aggregation Criterion”, in Proc. Of the European Conference on Information Retrieval (ECIR), pp. 264-275.

Ghosh, A. K., & Swaminatha, T. M. (2001). “Software security and privacy risks in mobile e-commerce”. Communications of the ACM, 44(2), 51-57.

Kobsa, A. (2007). “Privacy-enhanced personalization”. Communications of the ACM, 50(8), 24-33.

Luo, W., Xie, Q., & Hengartner, U. (2009). “Facecloak: An architecture for user privacy on social networking sites”. In Computational Science and Engineering, 2009. CSE'09. International Conference on (Vol. 3, pp. 26-33). IEEE.

Chen, W., Chan, H. P., Li, P., Bing, L., & King, I. (2019, June). An Integrated Approach for Keyphrase Generation via Exploring the Power of Retrieval and Extraction. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 2846-2856).

Zhang, Y., Zincir-Heywood, N., Milios, E.: World Wide Web site summarization. Web intelligence and agent systems: an international journal 2(1), 39–53 (2004)

J. M. Sotoca and F. Pla, “Supervised feature selection by clustering using conditional mutual information-based distances”, Pattern Recognition, 43 (6), 2010, pp. 2068–2081

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. Information Sciences, 509, 257-289.

Rabby, G., Azad, S., Mahmud, M., Zamli, K.Z., Rahman, M.M.: A flexible keyphrase extraction technique for academic literature. Procedia Computer Science. 2018. 135, 553–563.

Rabby, G., Azad, S., Mahmud, M., Zamli, K. Z., & Rahman, M. M. (2020). TeKET: a Tree-Based Unsupervised Keyphrase Extraction Technique. Cognitive Computation, 1-23.

Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M., & Jaggi, M. (2018). Simple Unsupervised Keyphrase Extraction using Sentence Embeddings. In Proceedings of the 22nd Conference on Computational Natural Language Learning (pp. 221-229).

Rafiei-Asl, J., Nickabadi, A.: Tsake: A topical and structural automatic keyphrase extractor. Applied Soft Computing. 2017. 58, 620–630

K.S. Jones, A statistical interpretation of term specificity and its application in retrieval, J. Document. 28 (1) (1972) 11–21

C. Florescu, C. Caragea, A new scheme for scoring phrases in unsupervised keyphrase extraction, in: Proceedings of the 39th European Conference on Information Retrieval (ECIR’17), Aberdeen, Scotland, 2017, pp. 477–483. April 9-13.

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. Text mining: applications and theory, 1, 1-20.

Mihalcea, R., & Tarau, P. (2004). TEXTRANK: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing.

Wan, X., & Xiao, J. (2008). Single document keyphrase extraction using neighborhood knowledge. In: AAAI, vol. 8, pp. 855–860.

Bougouin A, Boudin F, Daille B. Topicrank: Graph-based topic ranking for keyphrase extraction. In: Proc IJCNLP; 2013. p. 543–551.

Florescu C, Caragea C. Positionrank: an unsupervised approach to keyphrase extraction from scholarly documents. In: Proc. ACL; 2017. p. 1105–1115

Boudin F. Unsupervised keyphrase extraction with multipartite graphs. In: Proc NAACL: Human language technologies; 2018. p. 667–672.

University of Waikato NZ. Datasets of automatic keyphrase extraction. https://github.com/LIAAD/KeywordExtractorDatasets#theses. 2019.

M. Krapivin, A. Autaeu, M. Marchese, Large Dataset for Keyphrases Extraction, University of Trento, 2009 Tech Report # DISI-09-055

Medelyan, O., Witten, I. H., & Milne, D. (2008, July). Topic indexing with Wikipedia. In Proceedings of the AAAI WikiAI workshop (Vol. 1, pp. 19-24).

Sterckx, L., Demeester, T., Deleu, J., & Develder, C. (2015, May). Topical word importance for fast keyphrase extraction. In Proceedings of the 24th International Conference on World Wide Web (pp. 121-122).

Boudin F. pke: an open-source python-based keyphrase extraction toolkit. In: Proc COLING; 2016. p. 69–73

Lloret, E., & Palomar, M. (2012). COMPENDIUM: a text summarisation tool for generating summaries of multiple purposes, domains, and genres.

Lloret, E., Plaza, L., & Aker, A. (2017). The challenging task of summary evaluation: an overview. Language Resources and Evaluation. doi: 10.1007/s10579-017-9399-2

Hovy, E., & Marcu, D. (2005). Automated text summarization. *The Oxford handbook of computational linguistics*, (pp. 583–598).

Mani, I., & Maybury, M. T. (1999). *Advances in automatic text summarization*. the MIT Press.

Huang L, He Y, Wei F, Li W (2010) Modeling document summarization as multi-objective optimization. In: Proceedings of the third international symposium on intelligent information technology and security informatics, pp 382–386.

S.B. Silveira, A. Branco, Extracting multi-document summaries with a double clustering approach, Proceedings of the 17th International Conference on Applications of Natural Language to Information Systems (NLDB 2012), Lecture Notes in Computer Science, vol. 7337, 2012, pp. 70–81.

G. Costa, G. Manco, R. Ortale, E. Ritacco, Hierarchical clustering of xml documents focused on structural components, Data & Knowledge Engineering 84 (2013) 26–46.

M. Cataldi, K.S. Candan, M.L. Sapino, Narrative-based taxonomy distillation for effective indexing of text collections, Data & Knowledge Engineering 72 (2012) 103–125.

W. Chen, P. Chundi, Extracting hot spots of topics from time-stamped documents, Data & Knowledge Engineering 70 (7) (2011) 642–660.

H. Saggion, E. Lloret, M. Palomar, Can text summaries help predict ratings? A case study of movie reviews, Proceedings of the 17th International Conference on Applications of Natural Language to Information Systems (NLDB 2012), Lecture Notes in Computer Science, vol. 7337, 2012, pp. 271–276.

D. Shen, Q. Yang, Z. Chen, Noise reduction through summarization for web-page classification, Information Processing and Management 43 (6) (2007) 1735–1747.

Gupta, S., & Gupta, S. K. (2019). Abstractive summarization: an overview of the state of the art. *Expert Systems with Applications*, *121*, 49-65.

Nenkova, Ani, & McKeown, Kathleen (2012). A survey of text summarization techniques. In Mining text data (pp. 43–76). Springer.

Luhn, H. P. (1958). The automatic creation of literature abstracts. IBM Journal of Research and Development, 2(2), 159–165.

Lloret, Elena, & Palomar, Manuel (2009). A gradual combination of features for building automatic summarization systems. In Proceedings of the 12th international conference on text. Speech and dialogue (pp. 16–23). Berlin, Heidelberg: Springer-Verlag.

SAGGION, Horacio et POIBEAU, Thierry. Automatic text summarization: Past, present and future. In: *Multi-source, multilingual information extraction and summarization*. Springer, Berlin, Heidelberg, 2013. p. 3-21.

BARRIOS, Federico, LÓPEZ, Federico, ARGERICH, Luis, *et al. (2016)* Variations of the similarity function of textrank for automated summarization. T*he Argentine Symposium on Artificial Intelligence (ASAI) 2015 - 44 JAIIO; 44 JAIIO - ASAI 2015 - ISSN: 2451-7585, pages 65-72*.

Durrett, G., Berg-Kirkpatrick, T., & Klein, D. (2016, August). Learning-Based Single-Document Summarization with Compression and Anaphoricity Constraints. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1998-2008).

Liu X, Webster JJ, Kit C (2009) An extractive text summarizer based on significant words. In: Proceedings of the 22nd international conference on computer processing of oriental languages, language technology for the knowledge-based economy, Springer, pp 168–178.

Tonelli S, Pianta E (2011) Matching documents and summaries using key concepts. In: Proceedings of the French text mining evaluation workshop.

Gupta P, Pendluri VS, Vats I (2011) Summarizing text by ranking texts units according to shallow linguistic features. In: 13th international conference on advanced communication technology. pp 1620–1625.

Baralis E, Cagliero L, Mahoto N, Fiori A (2013) GRAPHSUM: discovering correlations among multiple terms for graph-based summarization. Inf Sci 249:96–109. doi: 10.1016/j.ins.2013.06.046.

Parveen D, Strube M (2015) Integrating importance, non-redundancy and coherence in graph-based extractive summarization. In: Proceedings of the 24th international conference on artificial intelligence. AAAIPress. pp 1298–1304.

Yao JG, Wan X, Xiao J (2015) Phrase-based compressive cross-language summarization. In: Proceedings ofthe 2015 conference on empirical methods in natural language processing, pp 118–127.

Lin H, Bilmes J (2010) Multi-document summarization via budgeted maximization of submodular functions. In: Human language technologies: the 2010 annual conference of the North American chapter of the association for computational linguistics, Association for Computational Linguistics, pp 912–920.

Alguliev RM, Aliguliyev RM, Hajirahimova MS, Mehdiyev CA (2011) MCMR: maximum coverage and minimum redundant text summarization model. Expert Syst Appl 38:14514–14522. doi: 10.1016/j.eswa.2011.05.033.

Ko Y, Seo J (2008) An effective sentence-extraction technique using contextual information and statistical approaches for text summarization. Pattern Recognit Lett 29:1366–1371. doi: 10.1016/j.patrec.2008.02. 008.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*, 31(3):297–328.

Katja Filippova and Michael Strube. 2008. Sentence fusion via dependency graph compression. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 177–185, Honolulu, Hawaii, October. Association for Computational Linguistics.

FILIPPOVA, Katja. Multi-sentence compression: Finding shortest paths in word graphs. In: *Proceedings of the 23rd international conference on computational linguistics*. Association for Computational Linguistics, 2010. p. 322-330.

GUPTA, Som et GUPTA, S. K. Abstractive summarization: an overview of the state of the art. Expert Systems with Applications, 2019, vol. 121, p. 49-65.

Florian Boudin and Emmanuel Morin. 2013. Keyphrase extraction for n-best reranking in multi-sentence compression. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 298–305, Atlanta, Georgia, June. Association for Computational Linguistics.

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. multi-document abstractive summarization using ilp based multi-sentence compression. In Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15, pages 1208–1214. AAAI Press.

NAYEEM, Mir Tafseer, FUAD, Tanvir Ahmed, et CHALI, Yllias. Abstractive unsupervised multi-document summarization using paraphrastic sentence fusion. In: Proceedings of the 27th International Conference on Computational Linguistics. 2018. p. 1191-1204.

Shang, G., Ding, W., Zhang, Z., Tixier, A. J. P., Meladianos, P., Vazirgiannis, M., & Lorré, J. P. (2018, January). Unsupervised Abstractive Meeting Summarization with Multi-Sentence Compression and Budgeted Submodular Maximization. In ACL.

DI FABBRIZIO, Giuseppe, STENT, Amanda, et GAIZAUSKAS, Robert. A hybrid approach to multi-document summarization of opinions in reviews. In: Proceedings of the 8th International Natural Language Generation Conference (INLG). 2014. p. 54-63.

D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Celebi, E. Drabek, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, Z. Zhang, MEAD — a platform for multidocument multilingual text summarization, Proceedings of the 4th International Conference on Language Resources and Evaluation, 2004, pp. 699–702.

DUNLAVY, Daniel M., O'LEARY, Dianne P., CONROY, John M., et al. QCS: A system for querying, clustering and summarizing documents. Information processing & management, 2007, vol. 43, no 6, p. 1588-1605.

L. Plaza, Uso de Grafos Semánticos en la Generación Automática de Resúmenes y Estudio de su Aplicación en Distintos Dominios: Biomedicina, Periodismo y Turismo, (PhD thesis) , 2011.

A. Belz, Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models, Natural Language Engineering 14 (4) (2008) 431–455.

S. Mohammad, B. Dorr, M. Egan, A. Hassan, P. Muthukrishan, V. Qazvinian, D. Radev, D. Zajic, Using citations to generate surveys of scientific paradigms, Proceedings of the North American Chapter of the Association of Computational Linguistics, 2009, pp. 584–592.

Erera, S., Shmueli-Scheuer, M., Feigenblat, G., Nakash, O. P., Boni, O., Roitman, H., ... & Konopnicki, D. (2019). A Summarization System for Scientific Documents. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations (pp. 211-216).

LLORET, Elena, ROMÁ-FERRI, María Teresa, et PALOMAR, Manuel. COMPENDIUM: A text summarization system for generating abstracts of research papers. Data & Knowledge Engineering, 2013, vol. 88, p. 164-175.

De Marneffe, M.C., MacCartney, B., Manning, C.D., et al.: Generating typed dependency parses from phrase structure parses. In: Lrec, vol. 6, pp. 449–454 (2006).

O. Ferrández, D. Micol, R. Muñoz, M. Palomar, A perspective-based approach for solving textual entailment recognition, Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, 2007, pp. 66–71.

D. Tatar, A.D. Mihis, D. Lupsa, Text entailment for logical segmentation and summarization, Proceedings of the 13th International Conference on Applications of Natural Language to Information Systems, Lecture Notes in Computer Science, vol. 5039, Springer, 2008, pp. 233–244.

Parikh, A., Täckström, O., Das, D., & Uszkoreit, J. (2016). A Decomposable Attention Model for Natural Language Inference. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (pp. 2249-2255).

Pasunuru, R., & Bansal, M. (2018). Multi-Reward Reinforced Summarization with Saliency and Entailment. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers) (pp. 646-653).

Li,Y. Luo.C and Chung, S.M. 2008. Text Clustering with Feature Selection by using Statistical Data Knowledge and Data Engineering. IEEE Transactions on Know and Data Eng. (2008), vol. 20(5), 641–651.

Dagan, I., Marcus, S., Markovitch, S.: Contextual word similarity and estimation from sparse data. In: Proceedings of the 31st annual meeting on Association for Computational Linguistics, pp. 164–171. Association for Computational Linguistics (1993).

Benghabrit,A. Ouhbi,B. Frikh,B. Behja,H. 2013. Text Clustering using Statistical and Semantic Data. In Proceedings of the 2013 World Congress on Computer and Information Technologies (June 2013), 1-6.

Edmundson, H. P. (1969). New methods in automatic extracting. *Journal of the ACM, 16*(2), 264–285.

Ferreira, R., de Souza Cabral, L., Lins, R. D., Pereira e Silva, G., Freitas, F., Cavalcanti, G. D., et al. (2013). Assessing sentence scoring techniques for extractive text summarization. Expert Systems with Applications, 40(14), 5755–5764.

Ouyang, Y., Li, W., Lu, Q., & Zhang, R. (2010). A study on position information in document summarization. In Proceedings of the 23rd international conference on computational linguistics: Posters. COLING '10 (pp. 919–927). Stroudsburg, PA, USA: Association for Computational Linguistics.

Abuobieda, A., Salim, N., Albaham, A. T., Osman, A. H., & Kumar, Y. J. (2012). Text summarization features selection method using pseudo genetic-based model. In Proceedings of the international conference on information retrieval & knowledge management (pp. 193–197).

Oliveira, H., Ferreira, R., Lima, R., Lins, R. D., Freitas, F., Riss, M., & Simske, S. J. (2016). Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization. Expert Systems with Applications, 65, 68-86.

Merrouni, Z. A., Frikh, B., & Ouhbi, B. (2019). Automatic keyphrase extraction: a survey and trends. Journal of Intelligent Information Systems, 1-34.

Xiaojun Wan and Jianguo Xiao. (2008). Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 969–976, Manchester, UK, August. Coling 2008 Organizing Committee.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. In Dekang Lin and Dekai Wu, editors, Proceedings of EMNLP 2004, pages 404–411, Barcelona, Spain, July. Association for Computational Linguistics.

Cao, Z., Li, W., Li, S., & Wei, F. (2017, February). Improving multi-document summarization via text classification. In Thirty-First AAAI Conference on Artificial Intelligence.

Lins, R. D., Oliveira, H., Cabral, L., Batista, J., Tenorio, B., Ferreira, R., ... & Simske, S. J. (2019, September). The cnn-corpus: A large textual corpus for single-document extractive summarization. In Proceedings of the ACM Symposium on Document Engineering 2019 (pp. 1-10).

Lin, C. Y. (2004, July). Rouge: A package for automatic evaluation of summaries. In Text summarization branches out (pp. 74-81).

Batista, J., Ferreira, R., Tomaz, H., Ferreira, R., Dueire Lins, R., Simske, S., ... & Riss, M. (2015, September). A quantitative and qualitative assessment of automatic text summarization systems. In Proceedings of the 2015 ACM Symposium on Document Engineering (pp. 65-68).

Autosummarizer (2015). Retrieved from http://autosummarizer.com/

Classifier4J (2005). Retrieved from http://classifier4j.sourceforge.net/

Wan, X. (2010, August). Towards a unified approach to simultaneous single-document and multi-document summarizations. In Proceedings of the 23rd international conference on computational linguistics (Coling 2010) (pp. 1137-1145).

Aliguliyev, R. M. (2009). A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Systems with Applications, 36(4), 7764-7772.

Song, W., Choi, L. C., Park, S. C., & Ding, X. F. (2011). Fuzzy evolutionary optimization modeling and its applications to unsupervised categorization and extractive summarization. Expert Systems with Applications, 38(8), 9112-9121.

Svore, K., Vanderwende, L., & Burges, C. (2007, June). Enhancing single-document summarization by combining RankNet and third-party sources. In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL) (pp. 448-457).

Ferreira, R., de Freitas, F. L. G., de Souza Cabral, L., Lins, R. D., Lima, R., de França Pereira e Silva, G., et al. (2014). A context-based text summarization system. In Proceedings of the 11th international workshop on document analysis systems (das) (pp. 66–70).

Ferreira, R., de Souza Cabral, L., Lins, R. D., Pereira e Silva, G., Freitas, F., Cavalcanti, G. D., et al. (2013). Assessing sentence scoring techniques for extractive text summarization. Expert Systems with Applications, 40(14), 5755–5764.

See, A., Liu, P. J., & Manning, C. D. (2017). Get To The Point: Summarization with Pointer-Generator Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1073-1083).

Chen, Y. C., & Bansal, M. (2018). Fast Abstractive Summarization with Reinforce-Selected Sentence Rewriting. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 675-686).

Rafael Dueire Lins, Rafael Ferreira, and Steven J. Simske. 2019. DocEng'19 Competition on Extractive Text Summarization. In Proceedings of the 2019 ACM Symposium on Document Engineering (DocEng '19). ACM, New York, NY, USA, 216–217. https://doi.org/10.1145/3342558.3351874

Bhat, I. K., Mohd, M., & Hashmy, R. (2018). SumItUp: A Hybrid Single-Document Text Summarizer. In M. Pant, K. Ray, T. K. Sharma, S. Rawat & A. Bandyopadhyay (Eds.), Soft Computing: Theories and Applications: Proceedings of SoCTA 2016, Volume 1 (pp. 619-634). Singapore: Springer Singapore.

Mahajani, A., Pandya, V., Maria, I., & Sharma, D. (2019). A Comprehensive Survey on Extractive and Abstractive Techniques for Text Summarization. Paper presented at the Ambient Communications and Computer Systems, Singapore.

Sahni, A., & Palwe, S. (2017). Topic Modeling on Online News Extraction. Paper presented at the Intelligent Computing andInformation and Communication, Singapore

Sethi, P., Sonawane, S., Khanwalker, S., & Keskar, R. B. (2017). Automatic text summarization of news articles. Paperpresented at the 2017 International Conference on Big Data, IoT and Data Science (BID)

Lovinger, J., Valova, I., & Clough, C. (2019). Gist: general integrated summarization of text and reviews. Soft Computing,23(5), 1589-1601. doi: 10.1007/s00500-017-2882-2

Roul, R. K., & Arora, K. (2019). A nifty review to text summarization-based recommendation system for electronic products. Soft Computing, 23(24), 13183-13204. doi: 10.1007/s00500-019-03861-3

Chakraborty, R., Bhavsar, M., Dandapat, S. K., & Chandra, J. (2019). Tweet Summarization of News Articles: An ObjectiveOrdering-Based Perspective. IEEE Transactions on Computational Social Systems, 6(4), 761-777. doi:10.1109/TCSS.2019.2926144

Rudra, K., Goyal, P., Ganguly, N., Imran, M., & Mitra, P. (2019). Summarizing Situational Tweets in Crisis Scenarios: AnExtractive-Abstractive Approach. IEEE Transactions on Computational Social Systems, 6(5), 981-993. doi:10.1109/TCSS.2019.2937899

Vijay Kumar, N., & Janga Reddy, M. (2019). Factual Instance Tweet Summarization and Opinion Analysis of Sport Competition.Paper presented at the Soft Computing and Signal Processing, Singapore.

Mihalcea, R., & Ceylan, H. (2007). *Explorations in Automatic Book Summarization.* Paper presented at the Joint Conferenceon Empirical Methods in Natural Language Processing and Computational Natural Language Learning,Prague.

Menéndez, H. D., Plaza, L., & Camacho, D. (2014). Combining graph connectivity and genetic clustering to improve biomedical summarization. Paper presented at the 2014 IEEE Congress on Evolutionary Computation (CEC).

Nasr Azadani, M., Ghadiri, N., & Davoodijam, E. (2018). Graph-based biomedical text summarization: An itemset miningand sentence clustering approach. Journal of Biomedical Informatics, 84, 42-58. doi:https://doi.org/10.1016/j.jbi.2018.06.005

Jiang, X.-J., Mao, X.-L., Feng, B.-S., Wei, X., Bian, B.-B., & Huang, H. (2019). HSDS: An Abstractive Model for Automatic Survey Generation. Paper presented at the Database Systems for Advanced Applications, Cham.

Anand, D., & Wagh, R. (2019). Effective deep learning approaches for summarization of legal texts. Journal of King SaudUniversity - Computer and Information Sciences. doi: https://doi.org/10.1016/j.jksuci.2019.11.015

Merchant, K., & Pande, Y. (2018). NLP Based Latent Semantic Analysis for Legal Text Summarization. Paper presented at the2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI).

Moratanch, N., & Chitrakala, S. (2017). A Survey on Extractive Text Summarization. Paper presented at the 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai.

Sanchez-Gomez, J. M., Vega-Rodríguez, M. A., & Pérez, C. J. (2020). A decomposition-based multi-objective optimization approach for extractive multi-document text summarization. Applied Soft Computing, 91, 106231. doi: https://doi.org/10.1016/j.asoc.2020.106231

El-Kassas, W.S., Salama, C.R., Rafea, A.A., Mohamed, H.K., Automatic Text Summarization: A Comprehensive Survey, Expert Systems with Applications (2020), doi: https://doi.org/10.1016/j.eswa.2020.113679

Patil, A. P., Dalmia, S., Ansari, S. A. A., Aul, T., & Bhatnagar, V. (2014). Automatic text summarizer. Paper presented at the 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI).

Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: a survey. Artificial Intelligence Review, 47(1), 1-66.

Pitler E, Nenkova A (2008) Revisiting readability. In: Proceedings of the 2008 conference on empirical methods in natural language processing. pp 186–195

Oren Glickman. 2005. Applied Textual Entailment Challenge. Ph.D. thesis, Bar Ilan University.

ALCÓN, Oscar et LLORET, Elena. SemPCA-Summarizer: Exploiting Semantic Principal Component Analysis for Automatic Summary Generation. Computing and Informatics, 2018, vol. 37, no 5, p. 1126-1148.

LIN, Chin-Yew et HOVY, Eduard. Automatic evaluation of summaries using n-gram co-occurrence statistics. In: Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics. 2003. p. 150-157.

Ferrández, ´O. 2009. ´ Textual Entailment Recognition and its Applicability in NLP Tasks. PhD thesis, University of Alicante.

FATTAH, Mohamed Abdel et REN, Fuji. GA, MR, FFNN, PNN and GMM based models for automatic text summarization. Computer Speech & Language, 2009, vol. 23, no 1, p. 126-144.

El-Kassas, W. S., Salama, C. R., Rafea, A. A., & Mohamed, H. K. (2020). Automatic text summarization: A comprehensive survey. Expert Systems with Applications, 113679.