جامعة سيدي محمد بن عبد الله بفاس
UNIVERSITE SIDI MOHAMED BEN ABDALLAH DE FES
كلية العلوم ظهر المهراز فاس
FACULTE DES SCIENCES DHAR EL MEHRAZ FES



N° d'Ordre : **/2020**

Formation Doctorale : **Sciences et Technologies de l'Information et de la Communication - STIC**

Spécialité : **Informatique**

Laboratoire : **Laboratoire d'Informatique, Signaux, Automatique et Cognitivisme- LISAC**

THESE DE DOCTORAT

Présentée par

Yassine EL ADLOUNI

Contributions to the Improvement of Arabic Information Retrieval: Application to Arabic Biomedical Community Question-Answering

Soutenue le 28 / 12 / 2020 devant le jury composé de :

Pr. Mohamed LAMRINI	***Faculté des Sciences Dhar El Mahraz - Fès***	***Président***
Pr. Ouafae AMMOR	***Faculté des Sciences et Techniques -Fès***	***Rapporteur***
Pr. Brahim OUHBI	***Ecole Nationale Supérieure d'Arts et Métiers Meknès***	***Rapporteur***
Pr. Ayoub AIT LAHCEN	***ENSA-Université Ibn Tofail de Kénitra***	***Rapporteur***
Pr. Hamid TAIRI	***Faculté des Sciences Dhar El Mahraz - Fès***	***Examinateur***
Pr. Jaouad BOUMHIDI	***Faculté des Sciences Dhar El Mahraz - Fès***	***Examinateur***
Pr. Said OUATIK EL ALAOUI	***ENSA-Université Ibn Tofail de Kénitra***	**Co-directeur**
Pr. Noureddine EN NAHNAHI	***Faculté des Sciences Dhar El Mahraz – Fès***	**Directeur de thèse**
Pr. Mohammed MEKNASSI	***Faculté des Sciences Dhar El Mahraz - Fès***	***Invité***

Année universitaire : 2019-2020

Contributions to the Improvement of Arabic Information Retrieval: Application to Community Question-Answering

Yassine EL ADLOUNI

This dissertation is submitted on November, 2020 for the degree of Doctor of Philosophy

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification except as declared in the Preface and specified in the text.

Yassine EL ADLOUNI
November, 2020

ملخص

استحوذت منصات الاسئلة و الاجوبة المجتمعية المعروفة اختصارا بالمنتديات على اهتمام الباحثين في السنوات الاخيرة لقدرتها على التموقع كبديل لمحركات البحث التي تتيح لمستعمليها إمكانية البحث عن المعلومات عبر كلمات مفتاحية، وانظمة الاسئلة و الاجوبة بصفة عامة. بالإضافة إلى قدرتها على اقتراح أجوبة لسؤال ما، فهى تعتبر عاملا لتوفير الوقت للمستعمل علاوة على ترشيد استخدام المنصة من خلال استئصال الاسئلة المتكررة.

نتوخى من خلال هذه الاطروحة الى تحسين تمثيل، بحث و فهم معنى النص باستخدام متجهات متراصة في فضاءات متعددة الابعاد بالاضافة الى استعمال قواعد معرفية.بينما ارتكزت الانظمة السابقة على تحليل النص من منظور لغوي أو صرفي، مما نتج عنه أداء متواضع راجع الى عدة تحديات كاللجوء الى استعمال الدارجة، ارتكاب اخطاء املائية او صرفية، استعمال حروف لاتينية للتعبير عن مفردات تقنية ، نجاهل علامات الترقيم، طرح اسئلة مركبة او تواجد صيغ مختلفة لنفس السؤال.

تهدف هذه الاطروحة الى تجاوز المقاربات السابق ذكرها الى التحليل المعنوي عبر استعمال التعلم الآلي والتعلم العميق بالاضافة الى القواعد المعرفية.

نستهل هذه الأطروحة باستكشاف أداء المقاربات الاعتيادية لاسترجاع المعلومات كالتعلم الآلي بالإضافة إلى بناء نماذج قادرة على استيعاب معنى النصوص و تعلم ترتيبها من خلال مقاربة مزدوجة للأجوبة المحتملة والتى نبرهن على نجاعتها مقارنة بالمقاربة الفردية.

نستعمل بعد ذلك مجموعة من المقالات الطبية لتدريب متجهات متراصة كبيانات لشبكات عميقة و متراصة. نبرهن بعد ذلك أن أخذ السياق بعين الاعتبار يمكن من تحسين اداء النظام.

في الختام نستخدم مقاربة تعتمد على التحليل الصرفي للجمل المكونة للسؤال الاصلي ولمجموعة الازواج من خلال خوارزمية تعلم آلي او باستخدام شبكة عصبية.

كلمات مفتاحية— المعالجة الالية للغة العربية، منصات الاسئلة والاجوبة المجتمعية، البحث عن المعلومة، التشابه المعنوي للنصوص، التعلم الالي، التعلم العميق

Abstract

Community Question Answering has attracted much attention in recent years due to their ability to stand in between the less verbose search engines offering a keyword-based lookup and the more verbose question answering systems. Besides their obvious purpose of suggesting potential answers to a user question, they enable users to save considerable time by avoiding duplicate questions.

Previous community question answering systems were mostly based on lexical and syntactic approaches leading to weak performances as they are unable to identify paraphrasing. This dissertation goes beyond these approaches seeking to improve text comprehension and retrieval by leveraging the semantic dimension of text using learning to rank, neural and knowledge-based models. To this end we propose a bunch of novel methods including topic modeling, distributional embeddings, and knowledge bases.

This thesis research starts by exploring the performance of classic approaches to information retrieval based on machine learning, presenting a topic modeling approach to improve text representation and depicting our learning to rank models by showing how pairwise models are more powerful than their counterpart pointwise approaches.

Then this dissertation presents our distributional embeddings trained on a biomedical corpus and used to power a novel neural architecture combining lexical features with a stack of convolutional networks. Afterward, we show how contextualized embeddings perform better than context-free embeddings by fine-tuning them on our task.

This thesis research concludes with a tree-based approach either using kernel trees combined with a machine learning algorithm or using binary constituency trees as inputs to a recursive neural network.

Keywords— Community Question Answering, Information retrieval, Arabic Natural Language processing, Topic Modeling, Deep Learning, Contextualized Embeddings, Tree-Based Approach

Résumé

Les systèmes de question réponse communautaires ont suscité beaucoup d'intérêt durant les dernières années vu leur capacité de se positionner comme alternative aux systèmes de recherche d'information basés sur les mots clés et les systèmes de question réponse. Hormis leur but de suggérer une réponse à la question posée par l'utilisateur, ils permettent d'épargner le temps aux utilisateurs tout en optimisant la performance du système à travers l'élimination des doublons.

Cette thèse a pour objectif l'amélioration de la représentation, la restitution ainsi que la compréhension du texte en utilisant la modélisation des thèmes, les plongements distributionnels des mots et les bases de connaissances. Les systèmes de question réponses préalables ont été généralement basés sur l'analyse lexicale et l'analyse syntaxique du texte, ce qui engendrait de faibles performances dues à leurs incapacité de détecter les reformulations entre autres. Cette thèse a pour but d'aller au delà de ces représentations en exploitant la dimension sémantique du texte à travers les modèles d'apprentissage pour la classification et le classement, les réseaux de neurones et les bases de connaissances.

Cette thèse commence par explorer la performance des modèles classiques pour la recherche d'information, présenter une approche à base de la modélisation des sujets et démontrer par la suite comment l'approche par pairs est plus performante que celle par points quand il s'agit de l'apprentissage du classement.

Ensuite, la thèse présente l'utilité d'un ensemble de plongements de mots générés à partir d'une collection d'articles du domaine biomédical afin de générer des représentations vectorielles denses des mots et qui sont par la suite injectées dans une pile de réseaux de neurones convolutifs. On démontre également l'utilité du contexte pour ces représentations en les ajustant pour notre plateforme.

Cette thèse se termine par une présentation de l'approche à base des arbres à travers l'utilisation des fonctions noyaux d'arbres ou les réseaux de neurones récursifs avec comme entrée les arbres binaires de constituants.

Mots clés — Systèmes Réponse aux questions de la communauté, Recherche d'information, Traitement du langage naturel arabe, Modélisation de sujets, Apprentissage en profondeur, Intégration contextualisée, Approche arborescente.

Acknowledgements

> It is not the mountain we conquer, but ourselves.
>
> *Sir Edmund Hillary*

Obtaining a doctoral degree is similar in many ways to climbing a high mountain; it is a long and tedious task. It is fraught with hazards and hurdles along the path that can dampen your spirit. However, it also offers incomparable opportunities for personal and professional growth.

First and foremost, I wish to express my deepest gratefulness to my advisors, Professors Meknassi Mohammed, Noureddine En-nahnahi and Said Ouatik Alaoui from Sidi Mohammed Ben Abdellah University - Fez and Horacio Rodriguez from Universitat Politècnica de Catalunya - Barcelona, who sparked my curiosity, fueled my imagination and made my journey to the summit a satisfying and rewarding experience.

I wish to thank my Ph.D. committee for insightful suggestions and comments. This dissertation benefits a lot from their advices.

I also would like to thank Universitat Politècnica de Catalunya for hosting me a wonderful internship and giving access to their cluster during my research endeavor, Al Akhawayn University for the opportunity to attend the Maghrebi Natural Language Processing (NLP) Summer School and Universität Hildesheim for the opportunity to attend the Autumn School for Information Retrieval and Information Foraging.

I am also grateful to my family and friends for their support, encouragements and unconditional love.

Contents

General Introduction **23**

1 Related Work **33**

1.1 Introduction . . . 33
1.2 Datasets . . . 33
1.3 Question Answering . . . 39
1.3.1 Question Types . . . 40
1.3.2 Factoid QA . . . 41
1.3.3 Knowledge Based QA . . . 42
1.3.4 Machine Comprehension . . . 42
1.3.5 QA in restricted domains: The Biomedical domain. . . . 43
1.3.6 Community QA . . . 45
1.4 Models . . . 46
1.4.1 Learning to Rank . . . 46
1.4.2 Looking for similar questions . . . 46
1.4.3 Graph Based . . . 47
1.4.3.1 Graph representation and Construction . . . 50
1.4.3.2 Representing events with graphs . . . 51
1.4.3.3 Random Walk Algorithms . . . 51
1.4.3.4 Matrices . . . 52
1.4.3.5 Subgraph matching . . . 53
1.4.3.6 Graph alignment . . . 54
1.4.3.7 Belief propagation . . . 54
1.4.3.8 Feature extraction . . . 54
1.4.3.9 Iterative methods . . . 55
1.4.3.10 Statistical Methods . . . 55
1.4.3.11 Graph Neural Models . . . 56
1.4.3.12 Knowledge Graphs . . . 57
1.4.4 Deep Learning . . . 58
1.4.5 Embeddings . . . 58

1.4.5.1 Non Contextual Embeddings . . . 58
1.4.5.2 Contextual Embeddings . . . 59
1.5 Shallow Vs Deep Models . . . 59
1.5.1 Shallow Approaches . . . 59
1.5.1.1 Lexical analysis . . . 60
1.5.1.2 Syntactic analysis . . . 60
1.5.1.3 Semantic analysis . . . 60
1.5.2 Deep models . . . 61
1.6 SemEval Contests . . . 61
1.6.1 Tasks . . . 61
1.6.2 Task 3: Community Question Answering . . . 62
1.6.2.1 Evaluation Metrics . . . 62
1.7 Conclusion . . . 63

2 Community Question Answering using Topic Modeling 65
2.1 Introduction . . . 65
2.2 Our Method . . . 66
2.2.1 Preprocessing . . . 67
2.2.2 Feature Engineering . . . 67
2.2.2.1 Bag Of Words . . . 67
2.2.2.2 Lexical Features . . . 67
2.2.2.3 Distributional Embeddings . . . 68
2.3 Our Approaches . . . 70
2.3.1 Unsupervised Approach . . . 70
2.3.2 Supervised Approaches . . . 71
2.4 Experimental Setup . . . 73
2.5 Results & Discussion . . . 77
2.6 Conclusion . . . 78

3 Community Question Answering using Neural Networks 81
3.1 Introduction . . . 81
3.2 Our Method . . . 82
3.2.1 Word Embeddings . . . 82
3.2.1.1 Data Cleansing . . . 83
3.2.1.2 Text Normalization . . . 83
3.2.1.3 Distributional Embeddings . . . 84
3.2.2 Preprocessing . . . 84
3.2.3 Preprocessing Pipeline . . . 84
3.2.4 Features Engineering . . . 85

3.2.4.1 Text Tokenization . . . 85
3.2.4.2 Ngram Cosine . . . 85
3.2.5 Our Model . . . 85
3.2.6 Cross-lingual Approach . . . 87
3.2.7 Sentence Encoders . . . 87
3.3 Experimental Setup . . . 87
3.3.1 Dataset Description . . . 87
3.3.2 Training Phase . . . 88
3.3.3 Validation Phase . . . 88
3.4 Results & Discussion . . . 89
3.5 Conclusion . . . 90

4 Community Question Answering based on Contextualized Embeddings 93
4.1 Introduction . . . 93
4.2 Our Method . . . 94
4.2.1 Pre-processing . . . 94
4.2.2 Sentence Representation . . . 95
4.2.3 Model Implementation . . . 95
4.3 Experimental Setup . . . 97
4.3.1 Datasets description . . . 97
4.3.2 Baseline Models . . . 97
4.3.3 Sequence length . . . 98
4.3.4 Obtained Results . . . 98
4.3.4.1 Ranking Results . . . 98
4.3.4.2 Classification Results . . . 98
4.4 Results & Discussion . . . 100
4.5 Conclusion . . . 102

5 Tree-Based Approach to Community Question Answering 103
5.1 Introduction . . . 103
5.2 Our Method . . . 104
5.2.1 Preprocessing . . . 106
5.2.2 Constituency Neural Network . . . 106
5.2.3 Tree Kernels . . . 108
5.3 Experimental Setup . . . 108
5.3.1 Datasets Description . . . 110
5.3.2 Training setup . . . 110
5.3.3 Implementation details . . . 110
5.3.3.1 Recursive Neural Network . . . 110

5.3.3.2 Tree Kernels . . . 111
5.4 Results & Discussion . . . 111
5.5 Conclusion . . . 113

General Conclusion **115**

Publications **119**

Bibliography **121**

A Résumé Détaillé **137**
A.1 Introduction . . . 137
A.1.1 Motivations . . . 138
A.1.2 Défis . . . 139
A.1.2.1 Présence de fautes de frappe . . . 139
A.1.2.2 Lettres non normalisées . . . 139
A.1.2.3 Utilisation des mots latins ou translittérés . . . 140
A.1.2.4 Jeux de données déséquilibrés . . . 140
A.1.2.5 Ressources rares . . . 140
A.1.3 Brève introduction de l'approche . . . 140
A.1.3.1 Modélisation de sujet . . . 141
A.1.3.2 Apprentissage du classement par paires . . . 141
A.1.3.3 Plongements biomédicaux arabes . . . 141
A.1.3.4 Réseau convolutif empilé . . . 141
A.1.3.5 Plongements contextualisés . . . 142
A.1.3.6 Approche à base des arbres . . . 142
A.2 Méthodes . . . 142
A.2.1 Modélisation des sujets . . . 142
A.2.2 Réseau de neurones convolutionnel empilé . . . 143
A.2.3 Plongements contextualisés . . . 144
A.2.4 Approche à base des arbres . . . 145
A.3 Conclusion . . . 147
A.3.1 Contributions . . . 148
A.3.1.1 Modélisation des sujets et approches par paires . . . 148
A.3.1.2 Combinaison des réseaux de neurones convulotionnels avec les caractéristiques lexicales . . . 148
A.3.1.3 Plongements contextualisés . . . 148
A.3.1.4 Approches à base des Arbres . . . 149
A.3.2 Prochains Travaux . . . 149

List of Abreviations

AS Answer Selection.

BIDAF Bi-Directional Attention Flow.

CNN Convolutional Neural Networks.

cQA Community Question Answering.

DAG Directed Acyclic Graph.

GPU Graphical Processing Unit.

GRU Gated Recurrent Units.

IR Information Retrieval.

LDA Latent Dirichlet Allocation.

LSA Latent Semantic Analysis.

LSTM Long Short Term Memory.

LTR Learning To Rank.

mAP Mean Average Precision.

MC Machine Comprehension.

ML Machine Learning.

MR Machine Reading.

MSA Modern Standard Arabic.

MT Machine Translation.

NER Named Entity Recognition.

NL Natural Language.

NLP Natural Language Processing.

NMF Non-negative Matrix Factorization.

NN Neural Networks.

OOV Out Of Vocabulary.

OWL Web Ontology Language.

QA Question Answering.

QR Question Ranking.

RDF Resource Description Framework.

RNN Recurrent Neural Networks.

RTE Recognizing Textual Entailment.

SNLI Stanford Natural Language Inference.

TAC Text Analysis Conference.

TREC Trec REtrieval Conferences.

List of Figures

1 Information Retrieval Architecture 23
2 Question-Answering Systems Architecture 24

1.1 A simplified example of a cQA thread 45
1.2 A cQA thread in Semeval 2017 challenge 46

2.1 PCA projection for Sodium Embeddings 68
2.2 tSNE projection for Sodium Embeddings 69
2.3 System Architecture 71
2.4 BiGRU Architecture 74
2.5 DotNet Architecture 74
2.6 PyramidNet Architecture 75

3.1 Neural Network composed of the input layer where both the question Q_i and the pair question q_{ij} are embedded using pre-trained embeddings, followed by a stack of convolutional networks then a fully connected layer with a softmax. 86
3.2 Training Accuracy for different stacks of kernels. The optimal values are obtained using up to 6gram $k \in 1, 2, .., 6$ where the accuracy plotted using red has the maximum values. 89

4.1 The preprocessing step of our model: the input XML files are parsed to extract the original question and the question-answer pair thereafter processed using a pipeline composed of a normalizer, a tokenizer, a tagger and a parser. 94
4.2 The model architecture: Context-free embeddings and contextualized embeddings are fed into a neural network classifier to output a score for each question/answer pairs. 96
4.3 The fine-tuning step of our model: the transformer is fed with three inputs, the token type, the attention flag and the token id. The transformer is trained to recognize masked out tokens. 96
4.4 Training loss value change with respect to the iteration steps. 99
4.5 The effect of sequence length on ranking metrics 99
4.6 The effect of sequence length on classification metrics 100

5.1 The architecture main components are the preprocessing step, tree kernels and the constituency neural network fed by embeddings generated from the webteb corpus. 105
5.2 Constituency Neural Network is a multilayer network where binary trees are fed as stacks, an embedding layer is used to transform tokens to vectors and a classifier is used to output the similarity between the original question and the question-answer pair. 107
5.3 Training loss value change with respect to the iteration steps. 108
5.4 Kernel trees for an original question (q) and a pair question (ρ). 109

A.1 Architecture globale du système. 143
A.2 Modèle Réseaux de Neurones Empilés. 144
A.3 Architecture du modèle: les plongements sans et avec contexte alimentent le classificateur à base des réseaux de neurones pour mesurer la pertinence pour chaque pair de question-réponse. 146
A.4 Les composants principaux de l'architecture sont le pré-traitement, les fonctions arborescentes et le réseau de neurones à bases des arbres de constituants augmenté les plongements générés depuis webteb. 147

List of Tables

1 SemEval 2017 Task 3 Subtask D Training Excerpt 26

1.1 QA Datasets 34
1.1 QA Datasets (cont.) 35
1.1 QA Datasets (cont.) 36
1.1 QA Datasets (cont.) 37
1.1 QA Datasets (cont.) 38

2.1 Tf-Idf Parameters 68
2.2 Word2vec Parameters/Hyperparameters 69
2.3 Models Hyperparameters 75
2.4 Summmary of Notations 76
2.5 SemEval 2017 Task 3 Subtask D 76
2.6 Semeval 2017 Task 3 Subtask D Competition Results 77
2.7 Test Dataset Results 77
2.8 Pairwise Vs Pointwise Results 77

3.1 Webteb Articles 83
3.2 Arabic Text Normalization 83
3.3 FastText Hyperparameters 84
3.4 JSON Format 85
3.5 K-CNN Hyperparameters 86
3.6 SemEval2017 Task3 SubtaskD Results 90

4.1 Model Hyperparameters 96
4.2 Semeval 2016 & 2017 CQA-MD datesets 97
4.3 Semeval 2016 ranking and classification results 101
4.4 Semeval 2017 ranking and classification results 101

5.1 CoNN hyperparameters values and descriptions 107
5.2 Semeval 2016 CQA-MD dateset instances count by category and split. . . . 110
5.3 SemEval 2016 Task 3 Subtask D Training Excerpt 111

5.4 Webteb corpus size by category 112
5.5 Performance comparison of our tree-based methods with SemEval 2016 results 112

A.1 Résultats pour le jeux de données SemEval2017 Task3 SubtaskD en utilisant le modèle réseaux de neurones empilés. 145
A.2 Résultats du classement et de la classification du jeux de données Semeval 2016. 146
A.3 Résultats du jeux de données Semeval 2017 pour le classement et la classification. 150
A.4 Comparaison des performances entre nos approches à base des arbres et les résultats lors de la compétition SemEval 2016. 150
A.5 Résultats du jeux de données SemEval 2017 en utilisant la modélisation de sujets. 150

General Introduction

Context and Research Path

Natural Language Processing NLP refers to the set of techniques involving the application of statistical models for the sake of understanding text. NLP apply lexical, syntactic, and semantic analyses, named entity recognition, and semantic role labeling to detect patterns in text, among its most widely used applications we can cite automatic summarization, machine translation, sentiment analysis, opinion mining, information extraction, plagiarism detection, natural language inference, and information retrieval.

The two main technologies used to extract information from large collections of documents are Information Retrieval (IR), and Question Answering (QA). The goal of Information Retrieval (IR) systems is to extract documents that best match a query. The two main tasks in IR are document indexing and search- ing as depicted in fig 1. Indexing is the task of representing a document by its key features to speed up its finding when a query is invoked. The goal of searching is to retrieve the most relevant documents and to rank them in order of decreasing match with a query.

While a QA system inputs questions in natural language form, searches for answers over a collection of documents, extracts, and formulates concise answers. As shown in Figure

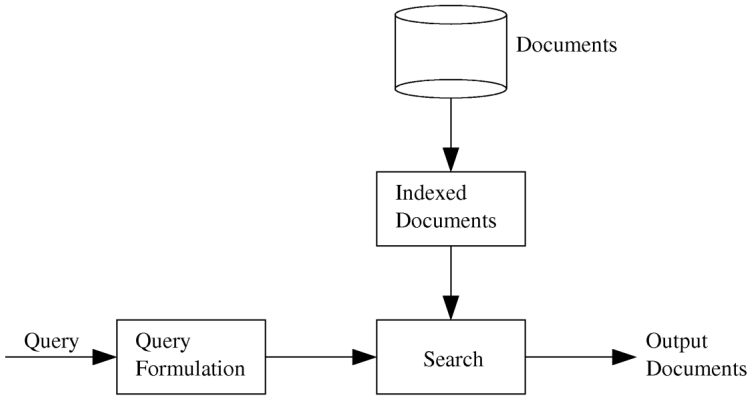


Figure 1: Information Retrieval Architecture

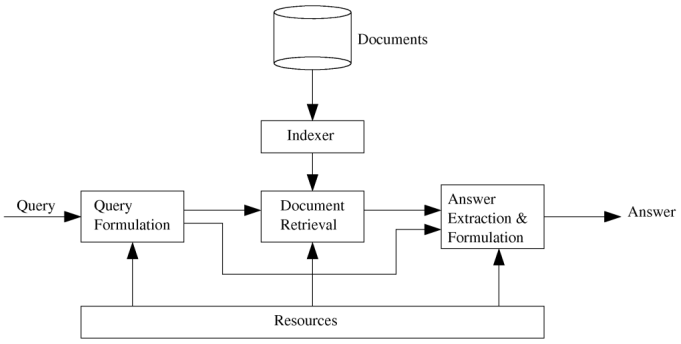


Figure 2: Question-Answering Systems Architecture

2, the three essential modules in almost all QA systems are question processing, document retrieval, and answer extraction and formulation.

Community Question Answering systems (cQA) are witnessing a growing interest recently with the advent of fora sites in social media as Yahoo!Answers, Quora, Reddit, and StackExchange. Much of this interest stems from their capacity to become a hub for communities to ask questions, seek information, and share knowledge around various topics in a trustworthy and timely manner. Besides their obvious purpose of getting the answer to a question, cQA should be able to help users check similar questions before submitting her request which could help save considerable time, avoid waiting for answers, reduce unanswered questions, minimize duplicate questions, thus, contributing to the efficiency of the platform.

The workflow of such systems is as follows. The asker firstly posts a question, and then other users can answer this question. When there are some answers to the question, the asker can select one answer as "Best Answer" from all the answers. In some cases, the "Best Answer" also can be voted by other users. There are two main limitations. Firstly, the asker can't get answers in real-time, due to the workflow mentioned above. Secondly, because of the limitations of a single user's knowledge, the relevance of many answers may be low, some answers are even wrong. Many cQA systems also support search services to overcome the unreal-time limitation. Users can also search queries in cQA forums. To return more relevant answers to users, the system predicts the relevance of answers and then returns those with high predicted relevance to users.

Content in these online fora is characterized by a high amount of redundancy, presence of misspelled words, and usage of paraphrasing which lead to low accuracies. Furthermore, their performance is highly influenced by the unbalanced nature of the dataset as few instances are relevant to the user's query and the lengthy aspect of questions and answers where the content is usually supplemented by greetings and other unrelated constructs.

The scarcity of resources for some languages impact also cQA systems as a semantic

analysis of the text is usually mandatory expecting diverse resources such as dictionaries, thesauri, knowledge bases, and so forth.

For Arabic, It is worth noting the following characteristics of biomedical community question-answering systems: i) the high density of medical terms, ii) the occurrence of English terms embedded within the Arabic texts, that could complicate the linguistic process of Arabic texts, iii) the relatively high overlapping of terms between the query texts and the texts of query/answer pairs in both Arabic texts and English translations, and iv) The relatively low quality of English translations, that could result on low accuracy of the linguistic process of English texts.

Community Question Answering could be defined as, given a question denoted the original question and a set of threads, a sequence of question-answer pairs, rank the threads based on their similarity related to the original question. cQA are usually subdivided into two subtasks:

- Classification: classify each thread as relevant, irrelevant, or perfect match related to the original question.
- Ranking: rank the threads based on their similarity related to the original question.

More formally, a cQA task could be defined as a tuple $(\mathcal{Q}, \mathcal{D}, \mathcal{Y})$ where $\mathcal{Q}$ is a set of original questions, $\mathcal{D}$ a set of question-answer pairs and $\mathcal{Y}$ a set of labels.

$$\mathcal{Q} = \{q_i\}_{i=1}^{n} \tag{1}$$

$$\mathcal{D}_i = \{(Q_{ij}, A_{ij})\}_{j=1}^{m_i} \tag{2}$$

$$\mathcal{Y}_i \in \{Relevant, Irrelevant, Direct\}^{1}. \tag{3}$$

Where:

- n is the number of original questions.
- m_i is the number of pairs related to the query q_i

Table 1 depicts an excerpt from a cQA dataset where:

- qid: the id of the original question.
- qaid: the id of the question-answer pair.
- query: the original question body.

[1]We employ frequently the abbreviate forms R, I and D

Table 1: SemEval 2017 Task 3 Subtask D Training Excerpt

qid	qaid	query	question	answer	relevance	confidence
200235	47148	Who gets tooth decay من يصاب بتسوس الاسنان	Tooth decay in children تعفن الاسنان عند الاطفال	Children's teeth are exposed to tooth decay ... تتعرض اسنان الطفل لمرض تسوس الاسنان...	D	1.0
200235	37871	Who gets tooth decay من يصاب بتسوس الاسنان	I have a problem of caries in the two front teeth ... عندي مشكلة التسوس في السنتين الاماميتين من...	Must follow the scientific specificities by the dentist ... يجب اتباع الخصوصيات العلميه من قبل طبيب الاسنان...	R	0.6641
200235	37427	Who gets tooth decay من يصاب بتسوس الاسنان	I have cavities and slight bleeding from the same decay لديا توسوس ونزيف دم خفيف من نفس التسوس مالحل	The solution is to visit the dentist to make fillings ... ... الحل هو زيارة طبيب الاسنان لعمل حشوات...	I	0.6866

- question: the question component of the question-answer pair.
- answer: the answer component of the question-answer pair.
- relevance: the relevance of the pair related to the original question, $relevance \in (D, R, I)$ where D denotes a strong relevance, R denotes relevance and I denotes irrelevance.
- confidence: the confidence value with respect to the relevance annotation, a decimal value between 0 and 1, $confidence \in [0, 1]$.

Motivations Of Our Work

Community question answering systems are increasingly used nowadays to help users find information more easily. Some are open domain such as Quora, Yahoo!Answers and Reddit while others are specific domain such as StackExchange which is a multi-language network of fora for diverse domains like technology, science, and business.

Related to Question Answering systems (QA), users ask short questions whereas answers are usually lengthy except in few cases when the respondent redirects to similar questions or the question is a yes/no or factoid question. This makes the task of lexical, syntactic, and semantic similarity laborious.

Questions frequently contain misspelled words introduced by users while typing their questions or stemming from orthographic errors which results in a huge number of out of vocabulary words which hinders the training and learning phases of the model.

In specific domain fora, questions and answers usually contain scientific terms, abbreviations requiring the usage of knowledge bases and gazetteers in order to produce accurate models.

We aimed to build effective models that are able to determine relatedness between a pair of questions and re-rank a set of questions based on their similarity related to an original question.

To build robust models, we had to overcome various challenges related to the Arabic language and others implied by the nature of community question-answering systems.

Challenges

The performance of Arabic cQA is highly influenced by the imbalanced nature of target datasets, the lengthy aspect of questions and answers, the technical topics discussed on these fora, the scarcity of linguistic resources characterizing Arabic and usage of Latin, transliterated and noisy words, these issues produce low values of mAP and accuracy[2] compared to English cQA.

Use of dialects

Unlike other cQA platforms involving users of the same background, biomedical cQA platforms gather two kinds of users, health-savvy users such as physicists and people seeking information.

The former uses more formal language, Modern Standard Arabic (MSA) for instance, while the latter uses a dialect Arabic depending on their location. This gap between the

[2]These and other evaluation metrics are defined in section 2.5

question vocabulary and the answer vocabulary results in a huge number of out of vocabulary words (OOV) hindering the classification and the re-ranking capabilities of the platform. This gap is further broadened by a large number of dialects for Arabic counting more than 30 dialects.

Lack of punctuation

Users in cQA usually blatantly misuse the punctuation marks as they rush to submit their questions which influence all subsequent phases for the pre-processing pipeline such as sentence segmentation, part-of-speech tagging, constituency parsing, and dependency parsing.

Presence of typos

Misspelled words often occur if the user is typing especially fast or due to a lack of knowledge. Some common typos are duplicate letters, missing letters, concatenated words. These introduce a high number of OOV terms and consequently impact the performance of the platform.

Unnormalized letters

The underlying letters composing two words in Arabic could be different even if the words seem visually similar. This issue stems from the fact that a letter in Arabic has three forms, the beginning form, the middle form, and the ending form, depending on the position of the letter in a specific word. This issue influences crucial steps of analysis, lexical, syntactic, and semantic analysis.

Lexical gap

The lexical gap between the original question and related questions, and between related questions and related answers is a consequence of paraphrasing. We could overcome this issue by using a semantic approach rather than a lexical or syntactic approach.

Use of foreign or transliterated terms

For reasons related to missing corresponding technical terms in Arabic, users, mainly more savvy ones, tend to use foreign spelled or transliterated words, English is largely used in eastern countries, while in the West, French is endorsed. The variance for transliterated terms is high.

Imbalanced Dataset

Most benchmark datasets for conventional QA and cQA are unbalanced, this stems from the fact that few questions are relevant to the original question. This problem, which could be serious when using ML approaches, could be resolved using augmentation, sampling, or using an algorithm that performs well on imbalanced data.

Scarce resources

The Arabic language suffers from the scarcity of resources, both processing tools and data, as thesauri, ontologies, dictionaries, annotated corpora, ...

A short introduction of the approach

cQA could be approached from different perspectives as a classification or as a ranking task. We have applied various approaches to produce an accurate model able to determine relatedness and re-rank the questions related to the original question. A thorough analysis should take into account the lexical, syntactic, and semantic similarity of questions.

A topic model approach is a simple yet powerful model capable to measure the semantic similarity between a pair of questions, the similarity score is used to rank the questions which is then used combined with a threshold to classify the question as relevant or irrelevant.

We have used TfIdf, the ubiquitous measure used in IR to rank documents, as features and various matrix factorization techniques, Latent Dirichlet Allocation denoted LDA, Latent Semantic Analysis denoted LSA, and Non-negative Matrix Factorization denoted NMF.

Deep learning approaches have proven quite efficient for tasks such as Named Entity Recognition NER, Recognizing Textual Entailment RTE, and Machine Translation MT. We have applied diverse neural architectures which lead to weak results. Following the intuition that ngrams are powerful for highly inflectional languages, we combined a stacked convolutional neural network with a lexical similarity measure to improve the performance of our system. We also used word embeddings trained on a medical domain corpus to improve further our results.

We further apply a cross-lingual approach from Arabic to English to check results. We use the same model (a stacked convolutional neural network combined with n-gram cosine) with pre-trained embeddings words produced by pyysalo, these pre-trained embeddings are trained on PMC and Pubmed with n-grams up to 7-grams. The obtained results are close to those obtained using the previous approach.

As paraphrasing usually characterizes the question-answer pairs, we apply a tree-based approach to leverage the semantic relationships between the original question and the questions thread. We also combine constituency trees to train a deep neural network to measure

the similarity between the original question and question-answer pair.

To sum up, the main contributions are as follows:

Topic modeling algorithm

The LSA-based model was able to outperform the state of the art system for the targeted dataset following an unsupervised approach. Unlike the SemEval 2017 winning system, the LSA-based model does not require external resources and therefore is more practical and more efficient for this task.

Pairwise learning to rank

Using the pairwise approach for cQA helps to improve the results for deep learning models.

Arabic biomedical embeddings

We use after pre-processing a bunch of Arabic biomedical articles to train dense distributional embeddings for vocabulary terms used to power various neural network architectures. This largely improves the performance of our system.

Stacked Convolutional Network

Though some neural architectures produce good results for benchmark datasets for other languages such as English, applying the same models on Arabic biomedical cQA results in weak accuracies, this could be explained by the noisy nature of our dataset. We propose a neural architecture powered by the aforementioned embeddings combined with lexical features to compute the similarity between the original question and the related question.

Contextualized embeddings

Contextualized embeddings have proven effective on a wide range of downstream tasks such as conventional question answering, paraphrasing detection, and textual entailment. Finetuning these embeddings on Arabic biomedical cQA improves neatly the performance of our system.

Tree-Based approach

Tree-kernels are powerful functions used to measure the similarity between multiple trees. They have demonstrated their efficacity for various other NLP tasks, we use kernel trees to classify the question-answer pairs related to the original question. We also feed constituency trees obtained from the original question and those obtained from the question-answer pair

to a deep neural network to rank the pairs regarding their relevance to the original question. The final model leads to superior results compared to state-of-the-art systems for community question-answering.

Outline of this document

The general introduction presents the context and field of our research, motivates the choice of the subject of this thesis, and highlights the main challenges faced in achieving an efficient cQA system to meet cQA forums user's needs.

Chapter 1 is devoted to present cQA systems, describe existing cQA databases, and synthesize the different state-of-the-art methods proposed in the literature.

Chapter 2 is dedicated to presenting our first contribution which aims to improve the process of extracting a response in a cQA system based on Topic modeling. The experimental results show the proposed method's performance.

Chapter 3 focuses on exposing the second contribution based on deep neural networks. The proposed model is described in a floor-by-floor walk-through manner before concluding this chapter with a detailed experimental study.

Chapter 4 addresses the author's third contribution to improving the process of classifying and ranking responses by exploiting contextualized embeddings. Description of the proposed approach is provided as well as details on its implementation before moving on to its validation through a set of tests carried out on two known datasets.

Chapter 5 presents a tree-based approach designed to construct our latest contribution. The different components of the method are defined in detail before shifting to evaluate its performance.

The general conclusion summarizes our findings and provides a discussion on our main prospects for future related research.

CHAPTER 1

Related Work

1.1 Introduction

The introduction of large manually annotated data has attracted much research on natural language processing and downstream tasks such as textual entailment, machine translation, summarization, question answering, and natural language inference. This revival of interest is attributed to the creation of many crowdsourcing platforms for manual annotation, the increased scale-up factor owing to advances in machine processing performance and usage of graphical processing units GPU for time-consuming tasks, the increased scale-out factor owing to the possibility of distribution of tasks and the availability of powerful clusters. All these factors have led to deep models that are able to model any Borel-measurable function thus producing platforms more accurate than humans in some cases.

1.2 Datasets

For years the datasets used for learning and evaluating QA approaches were extremely small and reduced to the QA tracks in Text REtrieval Conferences TREC contests and, later, in Text Analysis Conference TAC challenges. Fortunately, in last years bigger datasets (several orders of magnitude bigger) have been built and offered to the research community allowing the application of deep learning models to the task. In this section, we present some of these resources.

Table 1.1: QA Datasets

Dataset	Domain	Records	Example	
QUASAR-T [1]	Open	43,000	**Question**	7-Eleven stores were temporarily converted into Kwik E-marts to promote the release of what movie?
			Answer	the simpsons movie
			Context excerpt	In July 2007 , 7-Eleven redesigned some stores to look like Kwik-E-Marts in select cities to promote **The Simpsons Movie**
QUASAR-S [1]	Open	37,000	**Question**	javascript – javascript not to be confused with java is a dynamic weakly-typed language used for XXXXX as well as server-side scripting.
			Answer	client-side
			Context excerpt	JavaScript is a **Client Side** Scripting Language.
SquAD 2.0 [2]	Software	142,192	**Article**	Endangered Species Act
			Paragraph	"...These **later laws** had a low cost to society—the species were relatively rare—and little opposition was raised."
			Question	"Which laws faced significant opposition?"

Continue on the next page

Table 1.1: QA Datasets (cont.)

Dataset	Domain	Records	Example	
CuratedTREC [3]	Open	2,180	**Plausible Answer**	later laws
			Article	Live Free or Die
			Paragraph	"Live Free or Die" is the official motto of the U.S. state of **New Hampshire**, adopted by the state in 1945..."
			Question	What U.S. state's motto is "Live free or Die"?
WikiMovies [4]	Movie Dialog	181,679	**Plausible Answer**	New Hampshire
			Doc	Wikipedia Article for Blade Runner
			KB subset	Blade Runner directed_by Ridley Scott Blade Runner written_by Philip K. Dick, Hampton Fancher
			IE subset	Blade Runner, Ridley Scott directed dystopian, science fiction, film Hampton Fancher written Blade Runner
			Questions subset	Ridley Scott directed which films? What year was the movie Blade Runner released?

Continue on the next page

Table 1.1: QA Datasets (cont.)

Dataset	Domain	Records	Example	
TriviaQA [5]	Open	109,767	**Question**	The Dodecanese Campaign of WWII that was an attempt by the Allied forces to capture islands in the Aegean Sea was the inspiration for which acclaimed 1961 commando film?
			Answer	The Guns of Navarone
			Excerpt	The failed campaign, and in particular the Battle of Leros, inspired the 1957 novel **The Guns of Navarone** and the successful 1961 movie of the same name
WikiQA [6]	Open	29,261	**Question**	How a water pump works
			Answers subset	A small , electrically powered pump A large , electrically driven pump (electropump) for waterworks near the Hengsteysee , Germany.
PARALEX [1] [7]	Open	18M	**Text 1**	How big is New York
			Text 2	What is the population of New York
MedQuAD [2] [8]	Medical	47,457	**Text 1**	What is the latest news on tennitis, or ringing in the ear, I am 75 years old and have had ringing in the earsince my mid 5os. Thank you
			Text 2	What is the latest research on Tinnitus?

Continue on the next page

Table 1.1: QA Datasets (cont.)

Dataset	Domain	Records	Example	
SemEval-cQA(2016) [3] [9]	Open	3,169	**Text 1**	Books. Where can i donate books?
			Text 2	englishbooks. Where to buy english books? Is there a publiclibrary in doha? thanks
Clinical-QE(2016) [4] [10]	Medical	8,588	**Text 1**	Patient is reluctant to take medications so I have beentreating with smaller doses than I would with some otherpatients. How do I control her hypertension and still gether co-operation?
			Text 2	Patient reluctant to take medication.How to control hypertension and still get her cooperation?
Quora Question Pairs [5]	Open	170M	**Text 1**	newspaper
			Text 2	paper
			Text 3	journal
			Text 4	press
			Text 5	sections
			Text 6	daily
			Text 7	news
			Text 8	gazette
			Text 9	newsprint
			Text 10	magazine
			Text 11	diary

Continue on the next page

Table 1.1: QA Datasets (cont.)

Dataset	Domain	Records	Example	
Ask Ubuntu [6]	open	326,958	**Question**	How can I save a python script written on terminal?
			Answer	please provide the editor you used by editing the question
Quora Question Pairs [7]	Open	404,302	**Question 1**	What is the step by step guide to invest in share market in india?
			Question 2	What is the step by step guide to invest in share market?

1.3 Question Answering

QA can be defined as the task of, given a user information need, expressed as a natural language NL question, provide to the user the correct answer to the question, not, as usual in Information Retrieval IR systems, a set of documents where likely the answer can be found.
Although the origin of QA can be found in the eighties of the last century, with the development of many NL Interfaces to computer applications, especially to databases, the term QA was started to be used in the framework of QA tracks within the Text Retrieval Conferences, TREC[8] challenges (starting with TREC-8[9] in 1999). QA in general and its sub-disciplines have evolved along recent years following two divergent lines:

- increasing complexity of questions
- narrowing of the search space where answers are likely to be found.

QA systems can be viewed as a natural extension of IR systems. In IR systems user information needs are expressed through a query, usually consisting of a set of keywords (sometimes more complex and expressive query languages can be used). The query is then used for retrieving from a dataset (a collection of documents in TREC challenges, the whole Web, a domain restricted collection, a corporative textual database, etc.) the most relevant documents according to the query. The output of IR consists of a, sometimes ranked, set of documents retrieved from the dataset. If the query is well formulated, a good IR system should retrieve in its best ranked documents those satisfying the user information needs. In QA systems, the query consists of an NL question and the output of the system is not a set of likely relevant documents but the exact answer to the query.

While conventional IR systems use basically statistical approaches, QA systems use, increasingly, NLP techniques, both for processing the question and for extracting the answer. Some popular QA systems accessible through the Web are START (MIT)[10], AnswerBus[11], Webclopedia (ISI)[12] , Ask (before AskJeeves)[13], LCC[14], PowerAnswer[15], IBM's Watson[16], or Wolfram Alpha[17].

Perhaps the most popular system is Watson [11]. Watson has won in the best human champions at a game of the US TV show Jeopardy. It is being turned into a tool for medical

[8]http://trec.nist.gov
[9]http://trec.nist.gov/data/qa/t8_qadata.html
[10]http://start.csail.mit.edu
[11] http://www.answerbus.com
[12]http://www.isi.edu/natural-language/projects/webclopedia
[13]http://www.ask.com
[14]http://www.languagecomputer.com
[15]http://www.languagecomputer.com/demos/question_answering
[16]http://www.ibm.com/smarterplanet/us/en/ibmwatson
[17]http://www.wolframalpha.com

diagnosis. According to the IBM statements, its ability to analyze huge amounts of data is better than that of human doctors, and its deployment through the cloud reduces healthcare costs.

Conventional QA systems use to be structured into four modules (or steps):

1. Question Processing
2. IR of relevant documents
3. IR of relevant passages (fragments)
4. Answer Extraction

So, the language technologies involved have to cover these three tasks:

1. Linguistic analysis of questions using general or specific parsers (and grammars). This includes the definition of an appropriate tagset for classifying the questions (Question Type, QT)
2. IR engines (general or specific for the task)
3. IE techniques for extracting the answer

Many NLP sub-tasks are involved in QA and many approaches can be followed for facing these tasks. [12] have recently proposed a framework and a set of synthetic tasks with the goal of helping to develop learning algorithms for text understanding and reasoning and applying them to QA. The goal is to categorize different kinds of questions into skill sets, which become their tasks1. The framework is useful for describing the problems of the QA systems most faced[18].

1.3.1 Question Types

An important point when building a QA system is the definition of the Question Types (QT) and the Expected Answer Types (EAT). For general QA the most popular QT set is defined by Li & Roth [13]. The hierarchy[19] contains 6 coarse classes (ABBREVIATION, ENTITY, DESCRIPTION, HUMAN, LOCATION, and NUMERIC VALUE) and 50 fine classes.

In the Medical Domain [14] have build a taxonomy of 64 generic question types, and have used it for classifying 1,396 clinical questions. The three commonest generic types were "What is the drug of choice for condition x?" (150 questions, 11%); "What is the cause

[18]The tasks are publicly available at http://fb.ai/babi. Source code to generate the tasks is available at https://github.com/facebook/bAbI-tasks.

[19]available at http://L2R.cs.uiuc.edu/ cogcomp/

of symptom x?" (115 questions, 8%); and "What test is indicated in situation x?" (112 questions, 8%).

Assigning the correct QT is for factual questions quite simple. Given a QT tagset of N classes, the problem can be set as a multiclass classification task or as a set of N binary classifiers followed by a voting mechanism.

In the medical domain, however, questions use to be complex and answers appear frequently spread into several pieces of text. As a consequence of this, some approaches decompose the complex question into a set of simpler questions, usually connected by semantic relations (cause, condition, etc.). Performing the QA task over these simpler questions would be likely easier than on the initial complex one, although an additional step for integrating the partial answers is needed and this last step is far to be easy.

Some of these decomposition approaches are based on the syntactic structure of the question and apply techniques based on sentence simplification and normalization. [15], [16], [17] follow this approach. Other systems follow a semantic approach trying to identify symptoms, diseases, drugs, clinical signs, etc. [18] and [19] are examples of this approach.

Other systems, finally, as [20], try to generate similar questions by expanding the vocabulary used in the original question. More details on these approaches are included in Section 1.4.2.

1.3.2 Factoid QA

Initially, QA was limited to Factoid Questions, where the questions consisted of asking for a fact and the answer is likely to be found in a single document. Examples of factoid questions (from TREC 8) are the following:

- How much folic acid should an expectant mother get daily?
- Who invented the paper clip?
- What university was Woodrow Wilson president of?
- Where is Rider College located?
- Name a film in which Jude Law acted.
- Where do lobsters like to live?
- Who was Picasso?

Answering these questions was not especially challenging because an assertive formulation of the answer is likely to be found in one or more documents in the collection. The formulation of the question uses to be simple (there are no additional constraints) and the answer can be retrieved from a single document with no additional process. Most of the currently available QA systems satisfy this definition.

1.3.3 Knowledge Based QA

A knowledge-based QA system relies on information existing not as a text but in more structured forms. The structured information can be a full relational database or simpler structured databases like sets of RDF triples. There is an enormous amount of information, which is available in the semantic repositories, named Knowledge Graphs, as ontology with a format such as RDF, OWL, N-Triple, etc. Systems for mapping from a text string to any logical form are called semantic parsers. Semantic parsers for QA usually map either to some version of predicate calculus or to a query language like SPARQL or SQL. Although more complex forms of representation have been recently proposed, [21], [22] and [23], for instance, embed question and answer space into a common low dimensional vector space, using NN, QA is then stated as a problem of finding the closest answer candidate to the query. In Section 1.4.3.12 more details about Knowledge Graphs are given.

1.3.4 Machine Comprehension

As the question classification task depends heavily on the correct understanding of the question, machine comprehension abilities are needed.

Machine Comprehension MC, also named Machine Reading MR is the task of extracting the meaning of a text, usually representing it as a low dimensional vector. This representation can be used for many semantic-based tasks, like QA. Many of these tasks have been recently enhanced by the availability of huge size resources that allow the application of deep learning models. Besides datasets built for QA, shown in Table 1.1, popular resources are the Stanford Natural Language Inference SNLI Corpus and the CNN/DailyMail cloze test [24] and Children's Book Test, CBT [25]. SNLI was released by the Stanford Natural Language Processing Group. It is composed of human-written English sentence pairs with labels entailment, contradiction and neutral supporting the task of natural language inference also known as recognizing textual entailment denoted RTE.

In a QA setting, MC requires modeling complex interactions between the context and the query. Recently, attention mechanisms have been successfully applied to MC. Typically these methods use attention to focus on a small portion of the context and summarize it with a fixed-size vector.
[26] proposed a Gated-Attention (GA) Reader, integrating a multi-hop architecture with a novel attention mechanism, which is based on multiplicative interactions between the query embedding and the intermediate states of a recurrent neural network document reader.

[24] introduces a dynamic attention model to attend on context words from Convolutional Neural Networks CNN & DailyMail datasets. They employ Long Short Term Memory LSTM a popular implementation of Recurrent Neural Networks RNN to compute a combined document-query representation which is used to rank the candidate answers. These

include the DeepLSTM Reader, the Attentive Reader, and the Impatient Reader.

[27] uses an attentive reader with three components, an encoder, an attention module, and a prediction module. The relevance is computed using the bilinear attention function instead of the more popular tanh. [28] is based on a match-LSTM combined with a Pointer Net to constrain the output tokens to be from the input sequences.

[29] computes the attention weights once, which are then fed into an output layer for final prediction. Their model, Attention-Sum (AS) uses two bi-directional Gated Recurrent Units GRU networks, another popular type of RNN, to encode both the document and the input vectors.

In their Attention-over-attention model [30] use a 2D similarity matrix between the query and context words. [31] and [26] compute an attention vector between the query and the context through multiple layers, typically referred to as multi-hop. EpiReader [32] consists of two networks, where one proposes a small set of candidate answers, and the other reranks the proposed candidates conditioned on the query and the context.

[33] introduce the Bi-Directional Attention Flow BIDAF network, a multi-stage hierarchical process that represents the context at different levels of granularity and uses a bidirectional attention flow mechanism to obtain a query-aware context representation.

[34] proposed a Pointer Network (Ptr-Net) model to solve a special kind of problems where we want to generate an output sequence whose tokens must come from the input sequence.

[28] put together match-LSTM and Pointer Network for facing a QA task. Finally, the Natural Language Computing Group, Microsoft Research Asia has proposed R-NET, an end-to-end NN model for RC, which aims to answer questions from a given passage. R-NET first matches the question and passage with gated attention-based recurrent networks to obtain the question-aware passage representation. Then it proposes a self-matching attention mechanism to refine the representation by matching the passage against itself, which effectively encodes information from the whole passage. Finally, it employs the pointer networks to locate the positions of answers from the passages.

1.3.5 QA in restricted domains: The Biomedical domain.

[35] describe the specific characteristics of QA systems in the Biomedical domain.

Most relevant questions in the Biomedical domain follow roughly the PICO format. PICO is an acronym for Patient/Problem, Intervention, Comparison, and Outcome. Clinical questions containing elements that pertain to each of these semantic roles are considered well-formed. In Section 1.3.1 we have presented a QT taxonomy for this domain.

[36] investigate various ML approaches for question filtering that automatically determine whether a clinical question is answerable according to the evidence taxonomy proposed by [14]. Most approaches to medical QA in some way make use of domain-specific semantic

knowledge for information extraction and retrieval.

[37] describes a semantics-based approach for the development of a French medical QA system. The approach is notable for the use of pattern-based semantic models of medical questions and the use of UMLS concepts, semantic types, and relations for identifying named entities and extracting answers.

[38], [39] describe an approach to clinical question answering based on the semantic unification of a query PICO frame with those of candidate answers. The authors describe semantic knowledge extractors for identifying PICO elements in medical texts, a semantic matcher for scoring and ranking MEDLINE citations according to a query PICO frame, and an answer generator for extracting answers from the scored citations. AskHERMES[20] [40] is an online clinical question answering system capable of processing long and complex medical questions.

[41] also describe the characteristics of biomedical QA. Several biomedical QA researchers have discussed the differences between general open-domain QA and domain-specific QA, and the peculiar characteristics and challenges of the biomedical domain as an application domain for QA. [36], for example, note that restricted-domain QA can exploit domain-specific knowledge resources for deeper question processing and that it may take advantage of a domain-specific typology of questions to develop answer extraction strategies appropriate for each question type. Restricted-domain QA can exploit deeper text analysis/processing, taking advantage of domain-specific formatting and style conventions as well as domain-dependent terminology. In contrast to open-domain QA for which generic NER is a major concern, in restricted-domain QA domain-dependent terminology plays a major role and presents a major challenge. [37] also notes that biomedical QA, in contrast to open-domain QA, is challenged with a more acute need to cater for specialized terminological variation and that the gap in technicality between non-expert user questions and target documents may be larger than in other restricted domains. He also notes the role of knowledge and reasoning for biomedical QA, versus open-domain QA. Some of the characteristic features of QA in the biomedical domain follow:

- Large-sized corpora.
- Highly complex domain-specific terminology.
- Domain-specific lexical, terminological, and ontological resources.
- Tools and methods for exploiting the semantic information embedded in the above resources.
- Domain-specific format and typology of questions.

[20] http://www.AskHERMES.org

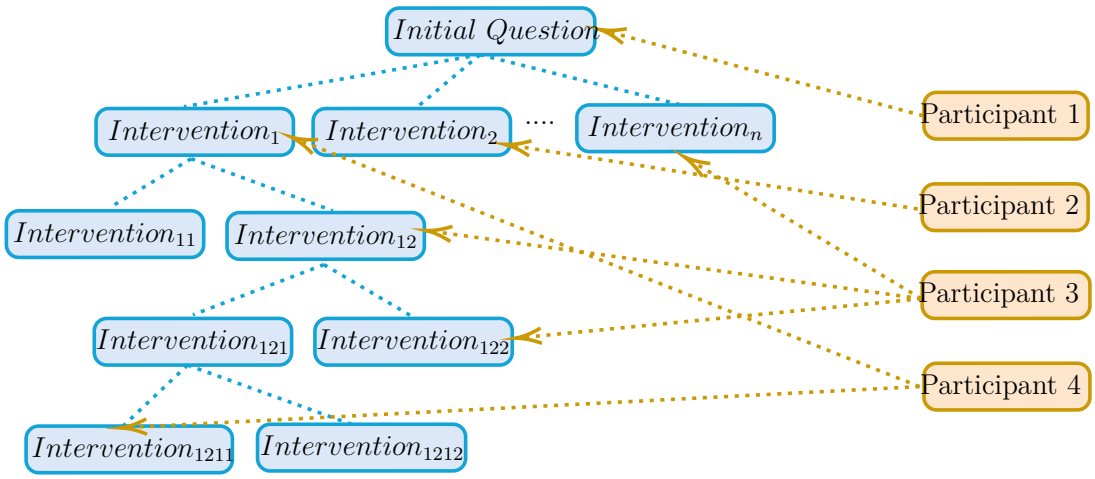


Figure 1.1: A simplified example of a cQA thread

1.3.6 Community QA

In this scenario, a member of the community formulates an initial query (an NL question) that triggers a thread of interventions of the community members that answer, refine, comment, the interventions of previous interventions forming a usually complex tangled thread. Members' interventions used to be related questions and answers. Both general-purpose and topic-specific communities are growing in numbers for posting questions and obtaining direct answers in a short period of time. In contrast to the traditional search engines such as Google, CQA services provide an alternative paradigm for seeking targeted information. [42] is a good example of this kind of systems, see also [43] and [9].

Community Question Answering (cQA) is one of the most interesting evolutions of conventional QA systems. In Fig.1.1 a simplified example of a cQA thread is shown. There are four community members. The first one uploads the initial query and from it, a complex thread is produced. Two of the paths starting in the initial query are two steps paths, probably consisting of simple query/answer patterns. The other is a complex thread including several ramifications and, probably, several communication actions from several participants. Participant_3, for instance, has performed three communicative acts within the thread. In Fig.1.2 a more realistic, though also simplified, thread, corresponding to Semeval-2017 challenge scenario is presented. In this case, the initial query is followed by 9 two steps threads, each composed of a question and an answer. No information is provided about the participants so we can consider that all of them are different.

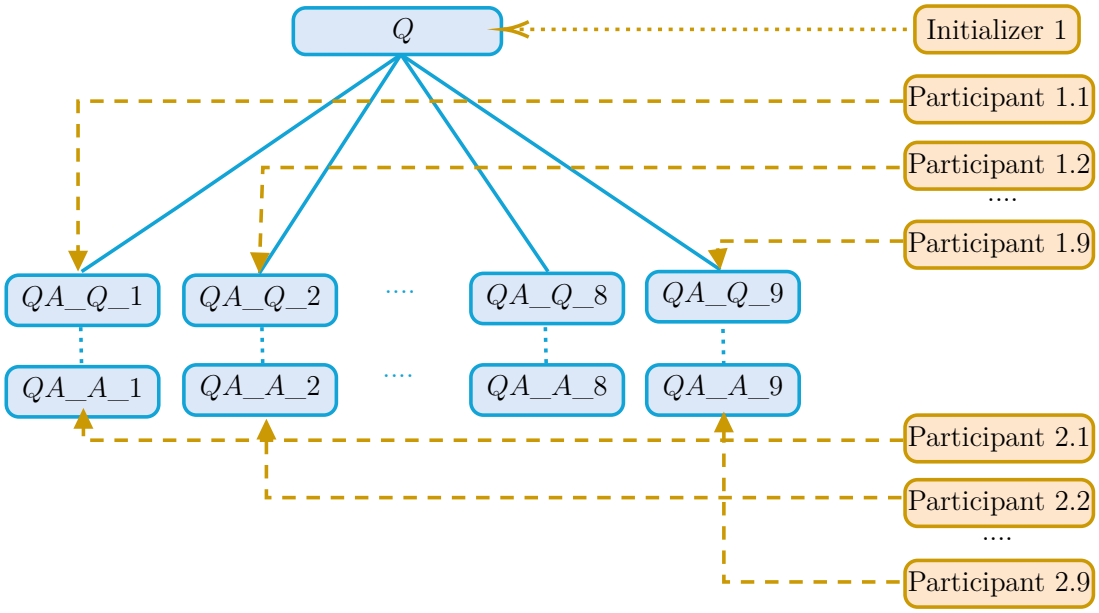


Figure 1.2: A cQA thread in Semeval 2017 challenge

1.4 Models

1.4.1 Learning to Rank

Learning to rank LTR is an emerging area in the intersection of machine learning ML, information retrieval IR and natural language processing NLP, it is the central problem of many tasks involving ranking such as document retrieval, entity search and linking, collaborative filtering, text summarization, and machine translation. For cQA, it is used for re-ranking the existing questions based on their similarity to the user query.

1.4.2 Looking for similar questions

When using supervised ML approaches for QA (for QT classification or answer extraction) frequently there is not enough learning material for performing the task. In these cases, it could be useful to collect additional training material. The best way is to look for similar questions to those included in the original training dataset.

Retrieving semantically equivalent questions is a challenging task due to three main factors: (1) The definition of similarity of questions is more difficult than the similarity between ordinary texts. Two questions are similar if their answers are similar. So, computing the similarity is a second-order task. (2) the same question can be rephrased in many different ways, and (3) two questions may be different but may refer implicitly to a common problem with the same answer. Therefore, traditional similarity measures based on word overlap are not applicable here.

Different strategies have been used such as machine translation [44], knowledge graphs [45], or topic modeling [46], [47]. [48] propose BOW-CNN, which combines a bag-of-words representation with a distributed vector representation created by a CNN.

1.4.3 Graph Based

Within the framework of NLP tasks and applications, the use of graphs for representing linguistic data arises very naturally in many situations. Examples vary from the web graph of documents, the representation of lexical resources, the representation of documents, paragraphs, sentences, phrases, at different steps of their processing, etc.

Regarding graph representation we have to deal with two issues:

- Representing a linguistic unit (a document, a paragraph, a sentence, etc.) as a graph, i.e. building the graph, i.e. what to put into the nodes and edges.
- Exploiting the graph structure for NL tasks.

We have to be extremely careful with the representational concepts involved in graphs. An Ontology is a frequent way of representing a lexical resource as a WordNet, a Wikipedia, a DBPedia, a domain resource, etc. But there are other possibilities, a taxonomy, a thesaurus, a lexicon, a simple closed list of items or controlled vocabulary, etc. An Ontology is frequently represented as a graph but other possibilities arise, as an OWL or an RDF database. Graphs are often represented using matrices (adjacency matrix, for instance) but this is not always the case.

If we represent ontologies by means of graphs we are able to apply the extremely rich (and sound) algorithmic paraphernalia of graph theory. If we represent graphs by means of matrices we are able to apply the rich resources of Linear Algebra. These are the strengths of these representations.

A graph G is a pair G = (V, E) of sets of nodes (or vertices) V, and edges (or arcs) E, where each edge connects, directly or not, a pair of nodes, i.e., $E \subseteq V \times V$. In general, V(G) refers to the set of nodes of graph G, and E(G) refers to the edges of graph G. If (v_1, v_2) is an edge of G, v_1 and v_2 are defined respectively as the origin, or source, and destination, or target, nodes of the edge.

In the case of representing the graph with an adjacency matrix, this matrix $A = (a_{i,j})$ of size $N \times N$ is defined by:

$$a_{i,j} = \begin{cases} 1, if a_{i,j} \in E \\ 0, otherwise \end{cases} \tag{1.1}$$

Graphs can be directed (named as DiGraphs) when the direction of edges is relevant, i.e. for $v_2, v_1 \in V$, $e_1 = (v_1, v_2)$ and $e_2 = (v_2, v_1)$ are different edges. When this distinction does

not exist we say that the graph is undirected. In the later case if $e_1 = (v_1, v_2) \in E$ then $e_2 = (v_2, v_1) \in E$.

A graph can be labeled on nodes when there is a function $\alpha : V \to L_V$. Similarly, a graph can be labelled on edges with a function $\beta : E \to L_E$. being L_V and L_E the domains of the labels for nodes and edges respectively.

A graph can be weighted when there is a function weight: $\theta : E \to \mathbb{R}$. The weight of an edge is a measure of its relevance, confidence, strength, etc.

More details on this topic can be found in [49],

Given a node $v \in V$, the edges having v as origin or destination are named incident to n. For an edge $e = (v_1, v_2)$ we can say that v_1 and v_2 are adjacent. In this case, v_2 belongs to the adjacency of v_1 (and the same for v_2). In the case of DiGraphs we have to distinguish the terms of input and output adjacent, incident, adjacency, etc.

For a node $v \in V$ we define the degree of v as the number of edges incident to it. In the case of DiGraphs we use the terms of input and output degree. The ratio between the number of edges (|E|) and the number of nodes (|V|) is named the density of the graph. When there is an edge between each pair of nodes the graph is said to be complete. A graph with low density is named sparse and with high density, dense.

An important concept on graphs is the concept of path. A path of length l, between two nodes n_s and n_t, source and target, is a sequence of nodes $n_0, ..., n_{l-1}$, so that $n_0 = n_s$, $n_{l-1} = n_t$ and for each pair of adjacent nodes in the path, n_i, n_{i+1}, for $0 \leqslant i < l$, an edge $e = (n_i, n_{i+1}) \in E$ exist. Similar definitions can be used for directed and undirected graphs. Given two nodes in graph G, if a path exists between them we say that the two nodes are connected. If the two nodes are connected there could exist several paths between them. From them, we could be interested in paths satisfying some property, as the path of minimum length. If the graph is weighted we could be interested in obtaining the path with the highest (or lowest) weight. We can use as weighting measure simply the length of the path and, so, we can define the minimum distance between two (connected) nodes of the graph. If we define this distance as d(u, v), we can define the eccentricity of a node u as $\max(\min d_{v \in V} d(u, v))$. The maximum eccentricity of all the nodes in G is defined as the diameter of the graph. The node with minimum eccentricity is named the center of the graph.

A path is simple when there are not nodes occurring more than once in the path. When the source and target nodes are the same the path is a cycle.

If all the nodes in G are connected (there exists at least one path between) we say that the graph is connected. In the case of digraphs when the pairs of nodes are connected in both directions we say that the graph is strongly connected. Frequently although the graph could be not connected some subgraphs are. We name these subgraphs connected components or strongly connected components. The maximum length connected components of a graph

are named cliques.

From graph G we can obtain its connected components. The graph defined with nodes corresponding to the connected components of G is defined as the reduced graph of G, G_{red}.

Special cases of graphs are:

- Tree: Is a connected acyclic graph
- Directed Acyclic Graph DAG
- Bipartite Graphs. A bipartite graph is a graph in which the set of nodes V can be split into two disjunct parts V_1 and V_2 so that all the edges connect a node in V_1 to a node in V_2, i.e. the subgraphs V_1 and V_2 contain no edges. [50], [51] model terms and documents as vertices in a Bipartite Graph with edges indicating the co-occurrence of terms and documents.
- A spanning tree of a graph is a tree containing all the nodes of the graph. If the graph is connected it contains at least one spanning tree. Sometimes more than one spanning tree exists. Sometimes we are interested on computing the minimum cost spanning tree.

Popular algorithms for computing all these concepts are[21]:

- Dijkstra: Computes in quadratic time the minimum paths from a node to all the other nodes
- Floyd: Computes in cubic time the minimum paths between all the nodes in the graph
- Warshall: Computes in cubic time the transitive closure of a graph
- Graph navigation:
  - Depth First Search (DFS)
  - Breadth-First Search (BFS)
- Computing the minimum cost spanning tree of a graph in quadratic time:
  - Prim
  - Kruskal
- Graph matching: Given a graph G = (V,E), a matching is a subset of E so that it contains no edges incident to the same node. Two variants of this algorithm are:

[21]Most used graph python libraries as networkX (https://networkx.github.io/) or i-graph (https://igraph.org/python/) provide implementations of these algorithms

  - maximal matching problem
  - complete matching problem

- Pathfinding: Find a path between two nodes satisfying some constraints (minimum length, minimum cost, satisfying node or edge labels)
- MaxFlow/MinCut methods
- Random Walk Algorithms
- Probabilistic Graphical Models. Including both Directed (as Bayesian networks) and Undirected models (as Hidden Markov Models)
- Matrix representation
- graph similarity
- subgraph matching

1.4.3.1 Graph representation and Construction

Graph representation implies deciding what information has to be included in nodes and edges. Many approaches have been followed in the past. A document can be represented as a graph whose nodes are the sentences of the document and the edges connect adjacent sentences and sometimes similar sentences. A document can be viewed as a Graph of Graphs (GoG) where the nodes are themselves graphs representing the tokens of each sentence as nodes. [52] presents a good survey of the representation techniques.

For instance, if words of a sentence are represented as nodes, the following approaches can be followed for representing edges:

- Word occurrence together in a sentence, paragraph, section, or document.
- Common words in the linguistic unit.
- Co-occurrence on the fixed window of n words
- Semantic relation - Words have a similar meaning, or are antonyms or have similar spelling.

Often the topology of the graph (nodes and edges) is manually handcrafted, sometimes, however, at least the edge structure is learned. [53] proposes a semi-supervised system for learning the edges of a word-based graph. An interesting proposal of in-word features is included. More details on this type of approach can be found in Xiaojin Zhu's Ph.D. [54].

1.4.3.2 Representing events with graphs

An event can be seen as a generalization of a binary relation between two entities. In an event there is a core entity, frequently a predicate, and a set of satellites, entities playing semantic roles over the core. It is clear that this structure can be easily represented as a graph. Core and satellites are represented as nodes and the relations between the core and its satellites as edges labeled with the semantic role.

[35] survey the event detection task in the medical domain. [55] is the most used system and the one giving the best results on contests (as the BioNLP '09 event extraction tas). Their method trains separate multi-class SVMs for detecting event triggers and arguments using an extensive set of features, especially those derived from dependency parse graphs. Their system then uses a rule-based approach for attaching arguments to their corresponding events. [56] describe a similar event extraction approach based on a prominent use of domain-specific terminology.

1.4.3.3 Random Walk Algorithms

These algorithms are based on computing the importance of a vertex within a graph. [57] discuss several possible measures of importance. A link between two vertices implies a vote for the importance of the adjacent nodes. The process, named iterative voting, consists of ranking over all vertices. The process can be modeled as a random walk on the graph. A walker takes random steps for updating the importance of a node when visited. The process converges to a stationary distribution of probabilities. The algorithm is guaranteed to converge if the graph is Aperiodic (holds for non-bipartite graphs) and Irreducible (holds for strongly connected graphs). The most important algorithms of this family are HITS and PageRank and their variants.

HITS [58] defines 2 types of importance: Authority and Hub. Authority can be thought of as being an expert on the subject, "You are important if important hubs point to you". Hub can be thought of as being knowledgeable about the subject "You are important if important authorities point to you".

Hubs and Authorities mutually reinforce each other random walks where odd steps you walk from hubs to authorities (follow outlink) and even steps you walk from authorities to hubs (follow inlink).

PageRank [59] is famous because it was used as the original Google algorithm for scoring Web pages. The algorithm performs Random walks over the Web following hyperlinks but introduces some small probability of jumping randomly out of the page. The fraction of time spent on each page is the PageRank. PageRank forms a probability distribution over the Web. PageRank is the principal eigenvector of the normalized link matrix of the Web.

Based on PageRank other variants have been proposed. The most popular variants are the Topic-sensitive PageRank [60] and Personalized PageRank [61]. Although PageRank was proposed for scoring Web Pages it has been applied to whatever type of graph to score their nodes. In the case of textual graphs two widely used algorithms are LexRank [62] and TextRank [63].

1.4.3.4 Matrices

The most frequent way of representing a graph is by means of its adjacency matrix. Given a graph $G = (V, E)$, so that the number of nodes is $N = |V|$, is adjacency matrix A is a square matrix of Boolean of size $N \times N$, so that the cell $A_{i,j}$ is set to True if there is an edge from i to j and to False otherwise. If G is undirected A is symmetric and can be represented by its half (a triangular matrix).

Many excellent books cover the topics of Linear Algebra needed for mastering the matricial representation of graphs. A good choice could be [64].

The basic techniques for the adjacency matrix are based on factorizing the matrix. Many forms of factorizing exist as Jordan normal form, Schur, Cholesky, SVD. Most of these methods are based on the Spectral Theorem.

Spectral Theorem The Spectral Theorem establishes the conditions to be satisfied by a matrix for being diagonalizable. The spectral theorem states that a matrix is normal if and only if it is unitarily similar to a diagonal matrix, and therefore any matrix A satisfying the equation $A^*A = AA^*$[22] is diagonalizable. In this case, the matrix is said to be normal. All the eigenvalues of a normal matrix are real. When two eigenvalues are different their corresponding eigenvectors are orthogonal. We can find, then, a basis of the space consisting only of eigenvectors.

An orthogonal matrix[23] is a square matrix whose columns and rows are orthogonal unit vectors (i.e., orthonormal vectors), i.e.

$$Q^T Q = Q Q^T = I \tag{1.2}$$

where I is the identity matrix.

This leads to the equivalent characterization: a matrix Q is orthogonal if its transpose is equal to its inverse:

$$Q^T = Q^{-1} \tag{1.3}$$

[22] We note as A^* the conjugate transpose of A and as A^T its transpose.

[23] This nomenclature is for the real space $\mathbb{R}$, in the complex space $\mathbb{C}$ these matrices are named unitary.

If a square matrix A is normal the following decomposition can be efficiently computed:

$$A = U\Lambda U^* \tag{1.4}$$

Λ is diagonal, being their entries the eigenvalues of A U is orthogonal, its columns are the eigenvectors of A

A generalization of the Spectral theorem for -square matrices leads to the Singular Values Decomposition (SVD) of the matrix, theoretical basis of the dimension reduction technique named Latent Semantic Indexing (LSI).

One big category of graph similarity methods is based on feature extraction; admittedly, a feature that contains important information about a graph is the vector of its eigenvalues. The Eigenvalue method is as follows: Let A_1 and A_2 be the adjacency matrices of graphs G_1 and G_2 respectively. Let also $L_1 = D_1 - A_1$ and $L_2 = D_2 - A_2$ be the laplacians of the graphs, where D_1 and D_2 are the corresponding diagonal matrices of degrees.

Graph similarity Following [65] we can define the problem of graph similarity as follows:

Given two graphs $G_1 = (V_1, E_1)$ and $G_2(V_2, E_2)$, with possibly different number of nodes and edges, and the mapping between the graphs' nodes, find a measure of similarity.

Graph similarity involves determining the degree of similarity between these two graphs. Intuitively, since we know the node correspondences, the same node in both graphs would be similar if its neighbors are similar (and its connectivity, in terms of edge weights, to its neighbors). Again, its neighbors are similar if their neighborhoods are similar, and so on. This intuition guides the possibility of using belief propagation (BP) as a method for measuring graph similarity, precisely because of the nature of the algorithm and its dependence on neighborhood structure.

One approach to evaluating graph similarity is graph isomorphism. Two graphs are similar if they are isomorphic, or one is isomorphic to a subgraph of the other, or they have isomorphic subgraphs.

1.4.3.5 Subgraph matching

Given a collection of T graphs, e.g. a graph time series, where there are T number of graphs, find approximate subgraphs that occur in a subset of the T graphs, each of them over the same set of N nodes, but with possibly different edges (weighted or unweighted). Assume that we know the correspondence between the nodes. Subgraph matching involves identifying the coherent or well-connected subgraphs that appear in some or all of the T graphs.

An interesting system facing this task is Substructure Index-based Approximate Graph Alignment (SAGA) [66],

CODENSE[24] [67] is a system for mining Coherent Dense Subgraphs. It constructs a summary graph formed with edges that appear greater than k times in the set of graphs. It then mines subgraphs within the summary graph using an algorithm that recursively partitions the graph into subgraphs based on normalized cut.

MODES, from the same authors, is short for Mining Overlapping DENSE Subgraphs. MODES is used in CODENSE to identify dense subgraphs.

GraphScope [68] is another method for finding coherent clusters in graphs over time. GraphScope assumes the sequence of graphs $G_1, ..., G_n$ are bipartite. It then partitions this sequence of graphs into segments using an information theoretic-criterion and then finds clusters within each segment.

1.4.3.6 Graph alignment

An important application of graphs, of special interest in the case of representing ontologies, is the alignment of pairs of graphs (or sometimes collections of graphs). [69] faced this problem in the case of OWL-Lite ontologies but the results can be easily generalized to other cases.

The alignment is based on searching for concordances between concepts (nodes) of the two graphs, taking into account the name of the nodes, their properties, their similarity, and their topological environment. [69] distinguish between local (first-order) and global (higher-order) approaches.

1.4.3.7 Belief propagation

Belief propagation (BP) is the basic technique for performing inference on Graphical Models. It has been applied with success to the problem of computing the similarity of two graphs. The original BP-based algorithm is naive and runs in $O(n^2)$ time. Some improvements have been proposed since then, as the Linearized BP (FaBP) graph similarity proposed in [70]. Also the known "Eigenvalue method" is based on these techniques.

1.4.3.8 Feature extraction

The key idea behind these methods is that similar graphs probably share certain properties, such as degree distribution, diameter, eigenvalues. After extracting these features, a similarity measure is applied in order to assess the similarity between the aggregated statistics and, equivalently, the similarity between the graphs. These methods are powerful and scale well, as they map the graphs to several statistics that are much smaller in size than the graphs.

[24] http://zhoulab.usc.edu/CODENSE/codense_dl.htm

An interesting example of the system performing feature extraction is *node2vec*. *node2vec*[25] [71] is a semi-supervised algorithm for scalable feature learning in networks. Given any graph, it can learn continuous feature representations for the nodes, which can then be used for various downstream machine learning tasks.

The approach returns feature representations that maximize the likelihood of preserving network neighborhoods of nodes in a d-dimensional feature space. It uses a 2nd order random walk approach to generate (sample) network neighborhoods for nodes.

Inspired by the Skip-gram model of Word2Vec, recent research established an analogy for networks by representing a network as a document [72]. The same way as a document is an ordered sequence of words, one could sample sequences of nodes from the underlying network and turn a network into an ordered sequence of nodes.

1.4.3.9 Iterative methods

The philosophy behind the iterative methods is that "two nodes are similar if their neighborhoods are also similar". In each iteration, the nodes exchange similarity scores and this process ends when convergence is achieved. The most popular method belonging to this category is the similarity flooding algorithm by [73].

1.4.3.10 Statistical Methods

Literature regarding network comparisons is well established. There are generally two different approaches to address this problem. The first studies the similarity of unlabelled or/and unequal sized graphs using node-level measures (e.g. node centrality measures). Examples of such methods include graph and subgraph isomorphism matching [74], graph edit distance (GED) and their extensions [75], maximum common subgraph-based measures [76], and Graph let Degree distribution (GDD) [77]. Another approach analyses the structural difference between labeled graphs. The central issue for comparing graphs, i.e. measuring distance, has been studied extensively. For instance, distance-based methods with labeled graphs, GED, [78] defined as the cost (obtained by using a cost function) of the least expensive sequence of edit operations that are needed to transform one graph into another.

GraphGrep[26] is a software package to search for a query graph in a database of graphs. Given a collection of graphs and a pattern graph, GraphGrep finds all the occurrences of the pattern in each graph. The pattern is a subgraph and it can be also a tree, a path, or a node. The pattern is expressed as a list of nodes and a list of edges.

[25] https://snap.stanford.edu/node2vec/

[26] https://cs.nyu.edu/shasha/papers/graphgrep/

1.4.3.11 Graph Neural Models

Graph Neural Networks (GNNs) were proposed by [79] as a general framework for mapping graphs into NN dense representations. GNNs are a general NN architecture defined according to a graph structure G = (V,E). Nodes of the graph are mapped into their embeddings in a way that the local and global topologies of the graph should be maintained in the NN space, i.e. if two nodes are connected by an edge their corresponding vectors should be closed. If the graph is edge-weighted the distance between the two vectors should be proportional to the weight of the edge.

GNNs are a recurrent neural network architecture defined on graphs. GNNs apply recurrent neural networks for walks on the graph structure, propagating node representations. The resulting node representation sare then used as features in classification and regression problems.

An important application of GNN is the use of feature learning techniques for graph-structured inputs. GNNs map graphs to outputs through two steps. First, there is a propagation step that computes embeddings for each node v; second, an output model maps from embeddings and corresponding labels to an output for each v V.

There are two settings for feature learning on graphs: (1) learning a representation of the input graph, and (2) learning representations of the internal state during the process of producing a sequence of outputs.

From the initial model of [79] GNN models have evolved in different directions following the different NN models proposed for NL tasks, RNN (LSTM and their variants, GRU), attention-based, CNN and their variants, pointer networks, sequence NN, etc. Graph kernels can be used for a variety of kernel-based learning tasks with graph-structured inputs. [80] generalize convolutions to graph structures. [81] converts an arbitrary undirected graph to a number of different DAGs with different orientations and then propagates node representations inwards towards each root, training an ensemble of models. [82] convert graphs into sequences by following random walks on the graph then learns node embeddings using sequence-based methods. [83] propose the Gated Graph Sequence NN, GGS-NN that extend the well-known GRU for extending them to a sequence.

Gated Graph Sequence Neural Networks modify GNNs to use gated recurrent units and to output sequences [83]. The biggest modification of GNNs is the use of GRU and unroll the recurrence for a fixed number of steps T and use backpropagation through time to compute gradients.

[84], [85] extend basic CNN models for learning CNN for arbitrary graphs. These graphs may be undirected, directed, and with both discrete and continuous node and edge attributes. This approach tries to extracting locally connected regions from graphs.

Various methods of graph embedding have been proposed in the machine learning literature. They generally perform well on smaller networks. The problem becomes much more

challenging when a real-world information network is concerned, which typically contains millions of nodes and billions of edges. Most existing graph embedding algorithms do not scale for networks of this size. An exception is LINE [72] which can scale to very large, arbitrary types of networks: undirected, directed, and/or weighted. The model optimizes an objective that preserves both the local and global network structures. The inverse process is faced by [86]. They generate a graph representation that captures data properties in a topological space (i.e., through vertices and their associations) from a set of embeddings. Using a single layer embedding for that purpose would result in a rather poor topology, as fully-connected layers represent but a portion of all patterns learned by the NN, and only a small subset of neurons activate for each data instance. To guarantee that the graph representation contains a topology rich enough to empower network analysis algorithms they the full-network embedding, which produces a vector embedding including all convolutional and fully-connected layers of a CNN.

1.4.3.12 Knowledge Graphs

A knowledge graph is a multi-relational graph composed of entities as nodes and relations as different types of edges. An instance of the edge is a triplet of fact <head entity, relation, tail entity> (denoted as (h,r,t)). We follow [87] in the following description.

TransE [23] represents a relation by a translation vector r so that the pair of embedded entities in a triplet (h,r,t) can be connected by r with low error. TransE is very efficient while achieving state-of-the-art predictive performance. However, it has flaws in dealing with reflexive/one-to-many/many-to-one/many-to-many relations.

In [22], a simplified case of TransE (named Unstructured) is used as a baseline. This model considers the graph as mono-relational and sets all translations r = 0, i.e., the scoring function is ‖h−t‖. Obviously, it cannot distinguish different relations.

The Distant Model [21] introduces two independent projections to the entities in a relation. It represents a relation by a left matrix W_{rh} and a right matrix W_{rt}.

Dissimilarity [88] is measured by L_1 distance between $W_{rh}\mathbf{h}$ and $W_{rt}\mathbf{t}$. This model is weak in capturing correlations between entities and relations a s it uses two separate matrices.

Bilinear Model [89] models second-order correlations between entity embeddings by a quadratic form: $\mathbf{h}^T W_r \mathbf{t}$. Thus, each component of an entity interacts with each component of the other entity.

Single Layer Model [88] introduces non-linear transformations by neural networks. It concatenates h and t as an input layer to a non-linear hidden layer then the linear output layer gives the resulting score: $\mathbf{u}_r^T f(W_{rh}\mathbf{h} + W_{rt}\mathbf{t} + \mathbf{b}_r)$.

NTN [88] is the most expressive model so far. It extends the Single Layer Model by considering the second-order correlations into non-linear transformation (NN).

[87] propose a method named translation on hyperplanes (TransH) which interprets a relation as a translating operation on a hyperplane. In TransH, each relation is characterized by two vectors, the norm vector (wr) of the hyperplane, and the translation vector ($\mathbf{d}_r$) on the hyperplane. For a golden triplet (h,r,t), that is correct in terms of worldly facts, the projections of h and t on the hyperplane are expected to be connected by the translation vector $\mathbf{d}_r$ with low error.

1.4.4 Deep Learning

Open-domain QA has made considerable progress recently thanks to the availability of new huge datasets (see Table 1.1) that allow the implementation of new deep learning architectures like attention-based and memory-augmented neural networks ([90], [91], and [92]).

[93] presents DrQA, a system that answers open-domain questions by reading Wikipedia.

I[94] approach the answer selection subtask using deep learning models. With a simple bag-of-words sentence model they obtain excellent results on several datasets. By using a more complex sentence model based on a convolutional neural network over bigrams, they improved the performance further.

Other relevant works include [95] and [96] which used a type of siamese network for learning to project question and answer pairs into a joint space.

Both [97] and [1] apply the BIDAF model. The former system uses a Bi-Directional LSTM contextual layer, a context-to-query attention layer, a modeling layer and, an output layer.

[98] propose DCN (Dynamic Coattention Network). DCN includes a coattentive encoder that captures the interactions between a question and a document. It also includes a decoder neural network that processes the coattention encoding to estimate start and end positions of a phrase in the document that responds to the question.

[99] face the CQA task retrieving similar questions using a convolutional neural tensor network (CNTN) architecture. Most of these approaches start with a pre-trained embedding layer, sometimes static and sometimes dynamic.

1.4.5 Embeddings

1.4.5.1 Non Contextual Embeddings

The use of embeddings instead of simple count-based representations of words is for several NLP tasks, close to our, by [100]. The most popular embeddings are Word2Vec [101] and GloVe [102] both learned from huge collections of open domain texts. Recently Sampo Pyysalo [103] has delivered an embedding dataset for the medical domain.

Several efforts have been performed recently for combining embeddings of different characteristics and mapping embeddings between different languages. The task is not easy because the vectorial spaces are different. [104] signaled that continuous word embedding spaces exhibit similar structures across languages and performed mappings between embeddings. [105] improved these cross-lingual word embeddings using all of them bilingual word vocabularies. A notable achievement is the system MUSE developed by the Facebook group [106] that using FastText and resulting in a high number of embeddings covering more than a hundred languages.

1.4.5.2 Contextual Embeddings

The year 2018 has witnessed an inflection point in transfer learning for NLP models. [107] has produced state-of-the-art results for many tasks using ELMo which is a deep contextualized word representation harnessing the complex characteristics of words such as syntax and semantics and their variation across contexts. [108] describes a Universal Language Model Fine-tuning for Text Classification, denoted ULMFit, which outperformed state of the art on six text classification tasks by applying inductive transfer learning. [109] uses generative pre-training of a language model on a diverse corpus of unlabeled text. [110] builds on top of the previous models and depicts a transfer learning for language understanding denoted Bert taking into account left and right context. The authors released pre-trained models for English, Chinese, and a multi-language model.

1.5 Shallow Vs Deep Models

Community question answering is intended to suggest similar questions to the user query which we refer to as Question Ranking QR or to pick a relevant answer to the user query which is called Answer Selection AS.

We distinguish between two approaches, shallow and deep approaches. The former applies a machine learning model on features obtained from the feature engineering phase, the latter is based on neural networks usually composed of many layers bypassing the feature engineering phase.

1.5.1 Shallow Approaches

Shallow approaches are based on a feature engineering step where features are obtained using a lexical, syntactic, or, semantic analysis.

1.5.1.1 Lexical analysis

Extensive work focused on lexical matching between the original question and the thread pair. A basic score could be obtained using TfIdf or the more elaborate BM25 [111], these are used usually as baselines [112].

[113] and [114] use n-gram, longest common subsequences, jaccard, word containment, and cosine to measure similarity.

[115], [116], and [117] have used machine translation evaluation scores such as BLEU, NIST, TER, BADGER, TERp, METEOR, SEPIA, and MAXSIM as features.

Lexical features are used to compute a rank score in an unsupervised way using cosine or fed into a classification model such as SVM, random forests, or a neural network.

Although a lexical analysis combined with a robust model is able to measure the lexical similarity between two chunks of sentences they are less effective for community question answering as they are unable to detect paraphrasing which is usually present in these cases.

1.5.1.2 Syntactic analysis

The syntactic approach uses tree matching to measure the similarity between the original question and the candidate answers. Various kernel trees including subtree kernel, subset tree kernel, partial tree kernel, smoothed partial tree kernel, and compositionally smoothed partial tree kernel are applied to constituency and dependency parse trees to rank the candidate answers as described by [118].

[119] and [120] select relevant text at the sentence and the constituency parse tree to boost the performance of a kernel tree, the former using a bunch of lexical and positional features combined with an SVM classifier and the latter using attention-based Long Short-Term Memory networks.

[121] applies long short-term memory neural networks to identify the best fragments in questions then apply tree kernel approach on constituency parse trees produced using Farasa [122], an Arabic NLP pipeline.

1.5.1.3 Semantic analysis

The majority of work harnessing semantic features to rank candidate answers is based on distributional embeddings using Word2Vec [123], GloVe [124], or FastText [125].

[126] leverages semantic relations between concepts extracted from Wikipedia to augment the questions in the semantic space.

Knowledge-based approaches are also used to expand the question and candidate answers from knowledge bases such as YAGO [127], BabelNet [128], DBpedia [129]. Graphs are then leveraged using embeddings over graphs approach [130] or a neural network over knowledge graphs approaches [131].

1.5.2 Deep models

[132] provides an LSA based algorithm to tackle cQA and shows that pairwise approaches improve significantly the results compared to pointwise approaches.

[130] provides a knowledge-based approach to tackle cQA using external knowledge. It uses a multi-head interactive attention network for multi-task learning combined with question categorization as an auxiliary task.

[133] proposes a recurrent convolutional neural network, a combination between convolutional networks and recurrent neural networks to match the question and the answer semantically.

[134] proposes a multi-modal multi-view semantic embedding model using a bayesian model to deal with the sparse nature of question-answer pair and the multi-view properties of cQA data.

[135] applies tree kernels on constituency trees to measure the similarity between the question and answer. They also use a long short term memory neural network to summarize the questions.

[136] uses named entity disambiguation based on knowledge bases then apply a topic modeling approach to these entities.

[137] applies a learning to rank approach on semantic representations of question and answer obtained using a neural network model.

[138] learns distributional representations of questions and candidate answers using LSTM and memory networks combined with a knowledge base.

[139] combined the subject and body of questions then uses a condensing neural network to learn the representation of the question and candidate answers.

[140] uses an adversarial neural network to relieve the problem of dataset labels imbalance by generating negative samples iteratively.

1.6 SemEval Contests

SemEval (Semantic Evaluation) is a shared task, it was first organized in 1998 to evaluate computational semantic analysis systems. The evaluation intends to explore text semantics.

1.6.1 Tasks

SemEval deals with various tasks including but not limited to textual entailment, semantic similarity, paraphrase detection, question answering, opinion mining. These tasks are targeting a bunch of languages such as English, Arabic, Spanish, Chinese, French, and German.

1.6.2 Task 3: Community Question Answering

The SemEval Task 3 subtask D, asks, given a new question, denoted the original question, and a set of the first 30 related questions (retrieved by a search engine), each associated with one correct answer, to re-rank the 30 question-answer pairs according to their relevance to the original question.

1.6.2.1 Evaluation Metrics

We adopt the official evaluation metrics which are also widely used for classification and ranking.

classification:

- Accuracy is the ratio of correctly classified instances divided by the total number of instances.
- Precision is the ratio of correctly classified positive instances to the total of positive instances.
- Recall is the ratio of correctly classified positive instances to the count of positive instances.
- F1-measure is the harmonic mean of the precision and the recall.

Ranking:

- mAP denoting the mean average precision is the official metric.

$$mAP = \frac{\sum_{i=1}^{|\mathcal{Q}|} AP_i@k}{|\mathcal{Q}|} \tag{1.5}$$

$$AP_i@k = \frac{\sum_{j=1}^{n} P(j).y}{\sum_{j=1}^{n} y} \tag{1.6}$$

$$P(j) = \frac{\sum_{j=1}^{n} y}{\pi_i(j)} \tag{1.7}$$

- AvgRec is the average of the recall for the original questions.

$$AveRec = \frac{1}{|\mathcal{Q}|} \sum_{i=1}^{|\mathcal{Q}|} R \tag{1.8}$$

- MRR, the mean reciprocal is the inverse of the position for the first correct answer in the ranked list.

$$MRR = \frac{1}{|\mathcal{Q}|} \sum_{i=1}^{|\mathcal{Q}|} \frac{1}{\pi_i(i)} \tag{1.9}$$

where

- $\mathcal{Q}$ is the set of original questions and |Q| is the count of $\mathcal{Q}$ elements
- k the number of considered question-answer pairs, k=10 for the Semeval Task
- π is a ranking list of the question-answer pairs

1.7 Conclusion

Recent years have witnessed increasing interest for natural language processing downstream tasks such as question answering, textual entailment and natural language inference, text summarization and machine translation. These performances owing to usage of deep architectures sometimes combined with commonsense knowledge have lead to state-of-the art models leveraging semantic approaches. This shift from lexical and syntactic approaches to semantic ones is paramount to a bunch of tasks as content is usually paraphrased and and context is critical to tackle them.

Our research is focused on community question answering for Arabic biomedical domain. Our choices stem from the fact that little research was done for Arabic and for the biomedical domain. We customize LSA, a highly used algorithm in information retrieval to adjust for the peculiarities of community question answering systems. We also harness the power of advances in deep learning models, we use top-notch contextualized-embeddings which have proven effective for main stream tasks in NLP leading to state-of-the-art results for two of the widely used cQA datasets. We also leverage the tree structure of sentences in order to improve further our results. Meanwhile, due to scarce resources for Arabic, our target was also to contribute to cQA for Arabic biomedical domain by publishing our trained.

CHAPTER 2

Community Question Answering using Topic Modeling

2.1 Introduction

The performance of Arabic cQA is highly influenced by the unbalanced nature of target datasets, the lengthy aspect of questions and answers, the technical topics discussed on these fora, the scarcity of linguistic resources characterizing Arabic and usage of Latin, transliterated and noisy words, these issues produce low values of mAP and accuracy compared to English cQA.

Most implemented models are time-consuming as they require the feature engineering step beforehand, they furthermore rely on pointwise approaches while pairwise and listwise approaches are more natural for ranking. The pointwise, the pairwise, and the listwise differ with regards to the loss function and are used for ranking where for the first two ones, the group structure is ignored transforming the ranking problem into classification, regression, or ordinal classification while for the listwise approach, ranking lists are used to harness the group structure. The loss function is defined as follows for each approach:

- The pointwise loss function is defined on a single object mapping a feature vector $\vec{x}$ to a label y.
- The pairwise loss function is defined on a pair object mapping a pair of feature vectors $(\vec{x}_1, \vec{x}_2)$ to a label y.
- The listwise loss function is defined on a tuple object mapping a tuple feature vectors $(\vec{x}_1, \vec{x}_2 ... \vec{x}_n)$ to a label y.

Relatively little information exists regarding the effectiveness of applying pairwise approaches combined with deep learning methods for cQA. To help address these gaps, we

seek to check the following hypotheses:

- A topic modeling approach could be effective for cQA.
- Reducing the features space could help reduce overfitting.
- Pretrained word embeddings could help improve the results.
- Using the pairwise approach for cQA is more natural than using the pointwise approach.

Our contributions are three-fold:

- The LSA-based model was able to outperform the state of the art system for the targeted dataset following an unsupervised approach. Unlike the SemEval 2017 winning system, the LSA-based model does not require external resources and therefore is more practical and more efficient for this task.
- The overfitting of neural networks models was partially reduced using pre-trained word embeddings trained on a curated list of medical domain articles, it was also alleviated by reducing the features space using the overlappings terms of the triplets instead of their concatenation.
- Using the pairwise approach for cQA helps to improve the results for deep learning models.

The rest of this chapter is organized as follows: In Section 2.2, we present the followed methodology. The used approaches are depicted in Section 2.3. The experimental setup is introduced in Section 2.4 and the obtained results are discussed in Section 2.5. Finally, the conclusions are presented in Section 2.6.

2.2 Our Method

A community question answering task could be defined formally as a tuple of three elements $\mathcal{S} = \{(\mathcal{Q}, \mathcal{D}), y\}$. $\mathcal{Q}$ denotes a set of queries. $\mathcal{D}$ denotes the pairs of question answers. $y \in \mathcal{Y}$ is a label representing the level of relatedness between the query and the pair where $\mathcal{Y} = \{Direct, Relevant, Irrelevant\}$ is a graded set. Given a query $q_i \in \mathcal{Q}$, consisting of a query, and a set of question-answer pairs $\mathcal{D}_i \in \mathcal{D}$, the task of cQA is to re-rank the question-answer pairs according to the conditional probability $Pr(y_i|q_i, \mathcal{D}_i)$. Table 2.4 depicts notations used in this chapter.

Fig.2.3 illustrates the architecture of our supervised model where training, validation, and test datasets are parsed from XML, preprocessed yielding tokenized and indexed terms

for each original question, question, or answer, the relations files contain for each dataset the list of relations between them. The output from the preprocessing step is combined with the embeddings then fed to different neural network models as inputs to Keras or Scikit-Learn to predict the similarity score and relevance.

2.2.1 Preprocessing

We used a preprocessing pipeline to tokenize and normalize our sentences before feeding them to our model.

2.2.2 Feature Engineering

During our study, we have used various features as inputs to our models depending on the model.

2.2.2.1 Bag Of Words

We have used Tf-Idf (scikit-learn implementation) as inputs for the topic modeling approaches based on LSA, LDA, and Negative Matrix Factorization denoted NMF, the tuned parameters are reported in Table 2.1 where :

- max_features: specify the size of the vocabulary considering the most frequently used terms.
- min_tf: Used to ignore terms with a document frequency strictly lower than the given threshold.
- max_tf: Used to ignore terms with a document frequency strictly higher than the given threshold. As the value used is a float number (0.1), it is considered as a proportion and therefore multiplied by the document count.
- sublinear_th: We used $1 + log f_{t,d}$ rather than $f_{t,d}$.
- norm: We used the euclidean norm for terms vectors.

$$tfidf_{t,d} = (1 + log f_{t,d}) * (log \frac{N}{1 + n_t}) \tag{2.1}$$

2.2.2.2 Lexical Features

We have used lexical features including Longest Common Subsequence, Jaro-Winkler, Levenshtein, Damerau-Levenshtein, Sorensen-Dice coefficient, N-Gram, and Q-Gram to measure the similarity between a pair of text chunks. These features were used for the SVM^{Rank} model.

Table 2.1: Tf-Idf Parameters

Parameter/Hyperparameter	Value
max_features	10,000
min_df	1
max_df	0.1
sublinear_tf	True
norm	L2

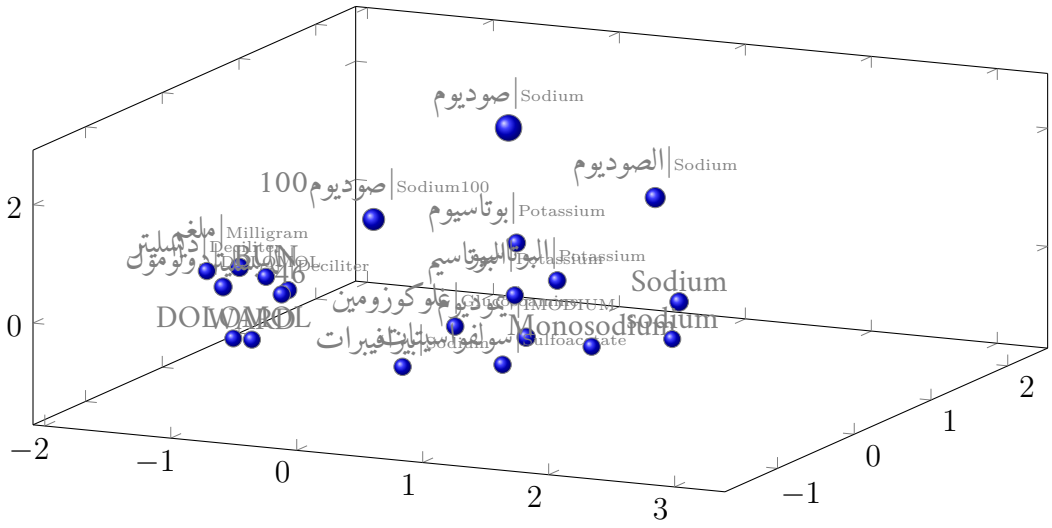


Figure 2.1: PCA projection for Sodium Embeddings

2.2.2.3 Distributional Embeddings

For neural models, BiGRU, DotNet, and PyramidNet, we have used distributional embeddings as they have a dense structure and encapsulate the semantics of words. We used word2vec as described in [141] on a scraped corpus of Arabic Medical articles containing 314,000 unique terms from various topics like diseases, body parts, and drugs.

The parameters and hyperparameters for Word2vec as depicted in Table 2.2 are fine-tuned based on the validation set performance, embeddings size specify the size of targeted vectors, vocabulary size the number of words to take into account, for our model, we take the first 10,000 most frequent words and use <unk> as a placeholder for uncommon words. We train our model incrementally using batches of size 128 records and 300,000. In order to predict a word from context, we use a window of two juxtaposing words to build our samples.

Fig.2.1 visualizes a projection using Principle Components Analysis with three components of the obtained embeddings for the word Sodium/صوديوم where similar minerals or drugs containing this mineral are adjacents.

Using the t-distributed stochastic neighbor embedding 'tSNE' as depicted in Fig.2.2, we notice that similar words to Sodium/صوديوم are clustering in the same region.

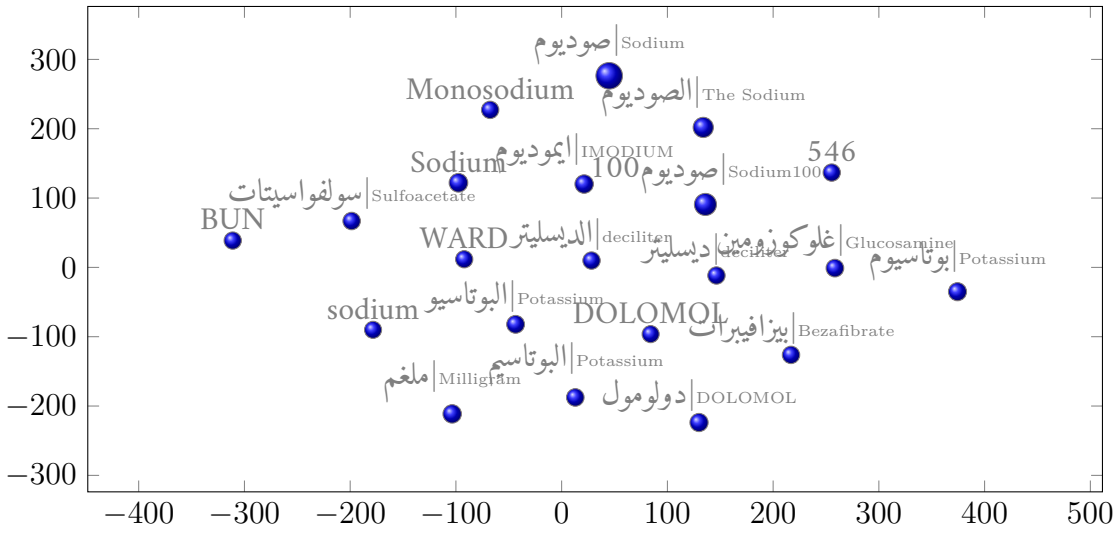


Figure 2.2: tSNE projection for Sodium Embeddings

Table 2.2: Word2vec Parameters/Hyperparameters

Parameter/Hyperparameter	**Value**
Embeddings size	128
Vocabulary size	10,000
Batch size	128
Number of steps	300,000
Skip window	1
Number of skips	2
Number of samples	64

2.3 Our Approaches

In our study, two approaches were explored:

2.3.1 Unsupervised Approach

The unsupervised approach is based on topic modeling models: LSA, LDA, and NMF. As depicted in Algorithm 1, to build our model, we first preprocess the query text by removing the stopwords and tokenizing the text using the ISRI light stemmer [142], finally, we apply a matrix decomposition using the SVD technique to the queries set, where the number of components is 900, this value is chosen based on the performance of the model on our validation set. The number of iterations for the SVD solver is 3, the decomposition is followed by a normalization step.

$$similarity(d, q) = cosine(x_{i,d}, x_{i,q}) \tag{2.2}$$

where $x_{i,q} = \Phi(q_i)$ are the vector features obtained from the query q_i and $x_{i,d} = \Phi(d_{i,j})$ are those obtained from the question $d_{i,j}$.

The cosine similarity is used as a score to rank our pairs, a threshold of 0.5 is chosen to deduce the relevance. The tunning of this threshold is chosen to maximize the performance on the validation set.

Algorithm 1: LSA based algorithm.

$components \leftarrow 900$
$iterations \leftarrow 3$
$\mathcal{M} \leftarrow TFIDF(\mathcal{Q})$
$\mathcal{W} \leftarrow SVD(\mathcal{M}, components, iterations)$;
for $i \leftarrow 1$ **to** m **do**
  for $j \leftarrow 1$ **to** l **do**
    $\mathbf{q} \leftarrow q_i \cdot \mathcal{W}$
    $d_{ij} \leftarrow TFIDF(d_{ij}^0 \frown d_{ij}^1)$
    $\mathbf{d} \leftarrow d_{ij} \cdot \mathcal{W}$
    $\sigma \leftarrow max(0, 1 - cosine(\mathbf{q}, \mathbf{d}))$
    if $\sigma > 0.5$ **then**
      $\rho \leftarrow true$
    else
      $\rho \leftarrow false$
    end
  end
end

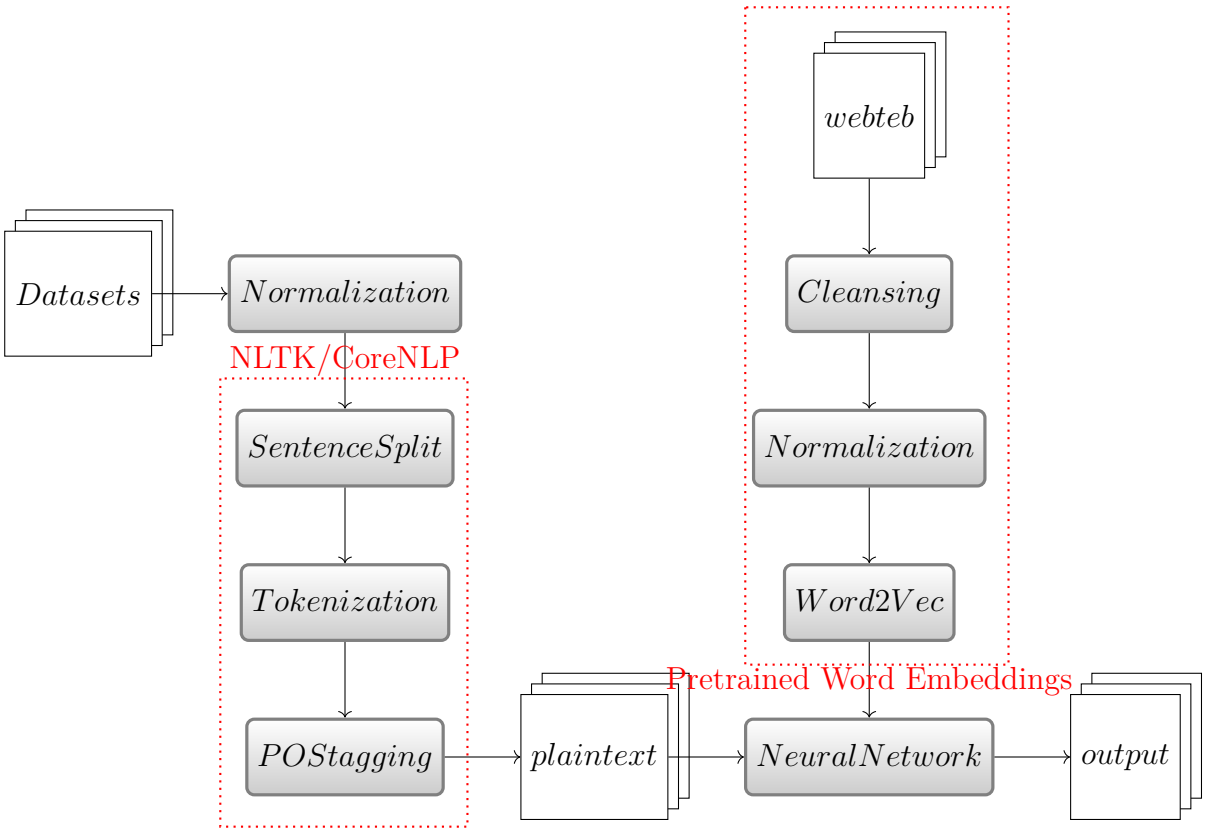


Figure 2.3: System Architecture

2.3.2 Supervised Approaches

Unlike the pointwise method where each original question q_i is compared to a pair of question-answer denoted (d_j^q, d_j^a) to measure their similarity, the pairwise method is based on a set of pairs of questions belonging to the same original question thus taking into account the relationships that could exist among questions. We transform our dataset in a way that makes it easy to learn the rankings. The outputs for the query q_i are $\mathcal{D}^+$ and $\mathcal{D}^-$ where the former contains pairs of questions (d_m, d_n) and $y_m \succ y_n$ while the latter contains pair of questions (d_m, d_n) where $y_n \succ y_m$ and $\succ$ denote the ranking operator, the set of pairs denoted $\mathcal{C}_i$ are obtained by combining, for the same original question q_i, questions that have different labels $y_m \neq y_n$. We chose 1 as the positive label and 0 as the negative label. We interleave $\mathcal{D}^+$ and $\mathcal{D}^-$ to obtain positive labels in even positions and negative labels in odd positions.

Therefore the transformed dataset is composed of:
feature vectors: $(x_i)^{(1)}, (x_i)^{(2)}$ where: $(x_i)^{(1)} = \Phi(d_0)$ and $(x_i)^{(2)} = \Phi(d_1)$
labels:

$$y_i = f((x_i)^{(1)}, (x_i)^{(2)}) \tag{2.3}$$

The value of y_i is 1 if i is even and 0 otherwise.

For Φ, the features extraction function, we have tried 2 approaches, one based on a word tokenizer while the other is based on trigrams. We have used different deep learning architectures for training following both pointwise and pairwise methods. These models accept as input sequences of text and output a score or a label depending on the target measure (ranking or classification). We proceed in 3 steps:

- A training step where many batches of queries, question-answer pairs, and its associated labels are presented to the model to minimize a loss function. We denote a transformed training dataset containing m queries by:

$$S' = \{(\mathbf{x_i^1}, \mathbf{x_i^2}, \mathbf{y_i})\}_{i=1}^{m} \tag{2.4}$$

- A validation step where a validation set is used to tune the different hyperparameters in a way to get high values for our target metrics, the mAP, and Accuracy.

$$V' = \{(\mathbf{x_i^1}, \mathbf{x_i^2}, \mathbf{y_i})\}_{i=1}^{m} \tag{2.5}$$

- A test step where for a new query q_{m+1} the transformed test dataset is denoted by:

$$T' = \{(\mathbf{x_{m+1}^1}, \mathbf{x_{m+1}^2})\} \tag{2.6}$$

  Where the label y_{m+1} should be predicted.

The architectures shown are deep models composed of several layers of different sizes combining various building blocks of neural networks such as Embedding, GRU and Fully Connected Layers denoted MLP. The intuition behind using these architectures is to detect semantic similarity by merging the embeddings layers directly which is equivalent to compare the left question and the right question at the word level as depicted in Fig.2.6, merging a flat representation of the embeddings which is equivalent to compare the left question and right question at the sentence level as depicted in Fig.2.5 or applying the embeddings to the overlapping to reduce the dimensionality of the network as depicted in Fig.2.4.

- **SVM$^{\text{Rank}}$** is a pairwise method based on SVM for ranking. We used the lexical features described in 2.2.2.2 and a linear kernel.

- **BiGRU** denotes the architecture described in Fig.2.4 where the input layer accepts as input either sequences taken either from the intersection or the concatenation of the original question terms and the union of the question-answer pair terms. These sequences are then embedded to get dense vectors using the previously described word2vec method. The next layer is a Bidirectional GRU [143] containing 8 units in

both directions with a RELU activation. The last layer is a fully connected network containing one unit and squashing the output through the sigmoid function. During training, we use the cross-entropy as a loss function

$$\mathcal{L} = -(ylog(\hat{y} + (1-y)log(1-\hat{y}))) \tag{2.7}$$

where y is the true value and $\hat{y}$ is the predicted value, we also use Adam [144] as the optimizer. Table 2.3 shows the parameters and hyperparameters used to train the network.

- **DotNet** denotes the architecture described in Fig.2.5 where the inputs are trigram sequences for each input question related to the same query as presented in the pairwise method paragraph. These sequences are then embedded where the embeddings are trained during the task. The next layer is a dense layer containing 10 units where the activation function is tanh and using a dropout of 0.2. The output layer is a scalar obtained using a normalized dot product of its inputs. For training we use adam as the optimizer and ranking hinge loss as a loss function, more formally defined as:

$$\mathcal{L} = max(0, m - \sigma(q, d^-) + \sigma(q, d^+)) \tag{2.8}$$

where σ is the ranking function, $q \in \mathcal{Q}$, $d^- \in \mathcal{D}^-$ and $d^+ \in \mathcal{D}^+$ and m is called the margin.
Table 2.3 exposes the other parameters and hyperparameters used to train the network.

- **PyramidNet** denotes the architecture described in Fig.2.6 where the inputs are sequences of words obtained from the inputs questions that are later embedded using word2vec pre-trained embeddings described previously. The next layer denoted the hidden layer in Fig.2.6 is composed of two legs, the first leg use element-wise multiplication of the inputs embeddings, later on, the k largest entries are processed through multiple dense layers with shrinking sizes as mentioned in Table 2.3. and softplus as the activation function. The second leg process the first question using a softmax function. Both legs flatten their outputs and element-wise multiplication is used to produce the rankings.

2.4 Experimental Setup

In this section, we evaluate the proposed methods on Semeval 2017 Task 3 Subtask D benchmark datasets. This dataset was extracted from Arabic medical fora. The challenge provided a dataset containing three XML files for training, validation, and test, Table 2.5

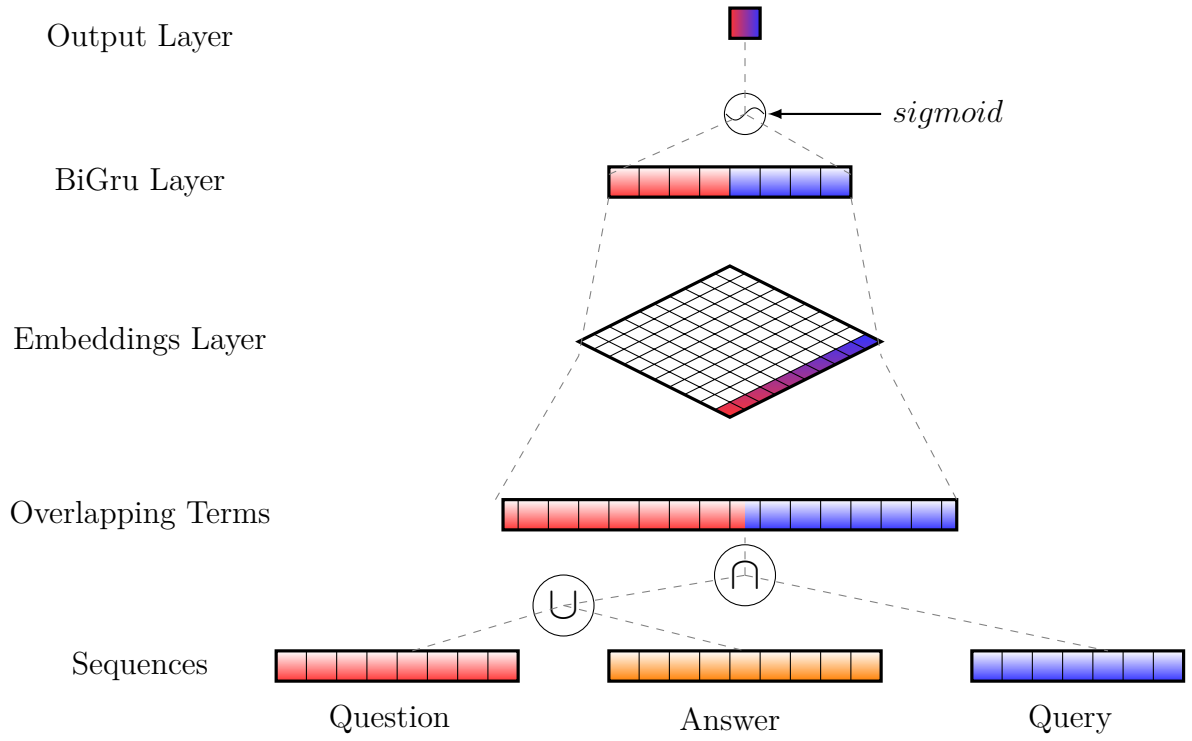


Figure 2.4: BiGRU Architecture

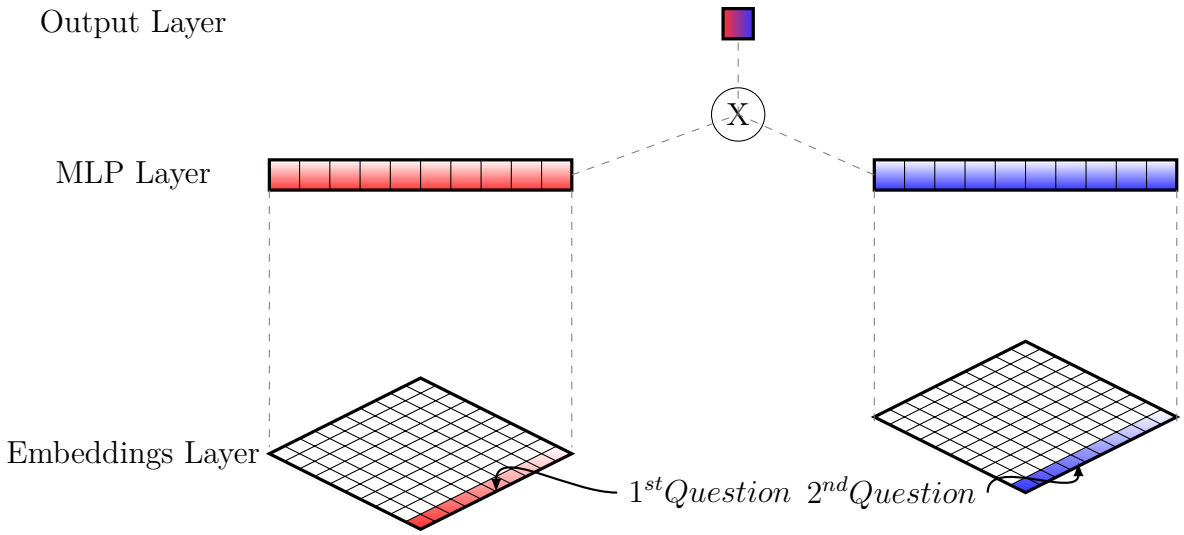


Figure 2.5: DotNet Architecture

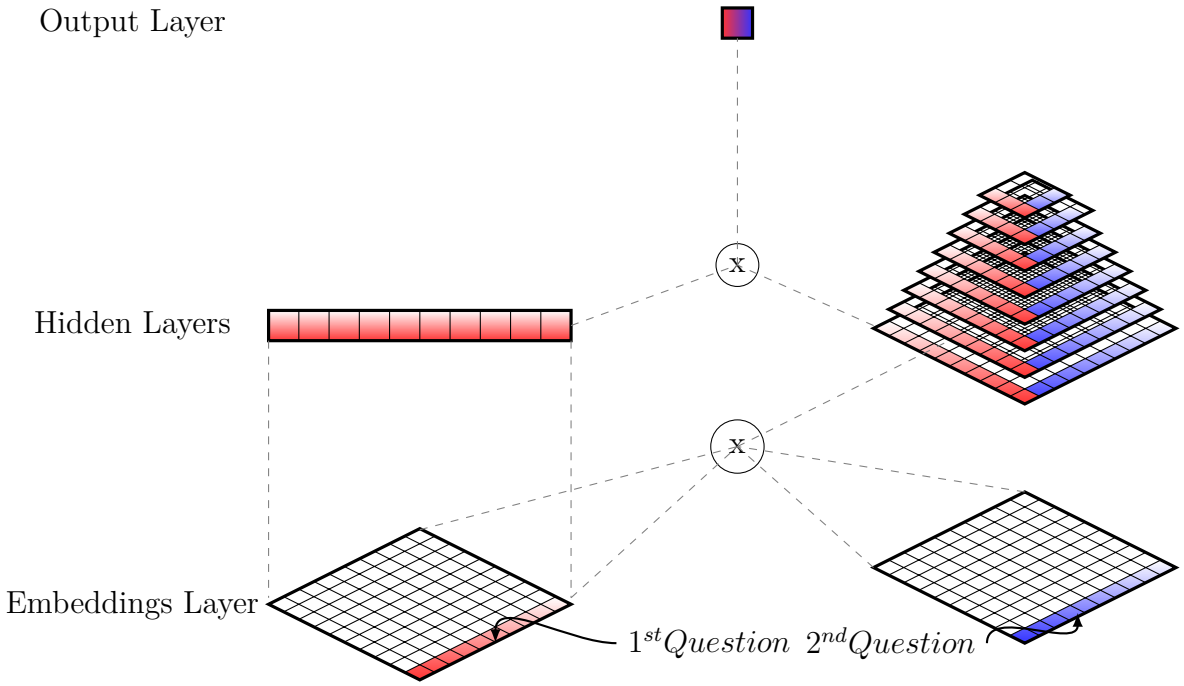


Figure 2.6: PyramidNet Architecture

Table 2.3: Models Hyperparameters

Parameter/Hyperparameter	**PyramidNet**	**DotNet**	**BiGRU**
Embeddings Size	128		
Vocabulary Size	89,150	69,414	10,000
Query Length	64	64	1000
Question Length	64	64	314
Batch size	100	100	128
Query per iteration	50	50	-
Batch per iteration	5	5	-
Epochs	100	100	20
Optimizer	Adam		
Learning Rate	0.01		
Number of hidden layers	6	1	1
Sizes of the hidden layers	[128, 64, 32, 16, 8, 1]	[32]	[8]
Margin	1	1	-
Dropout	-	-	0.5

Table 2.4: Summmary of Notations

Notation	Description
$\mathcal{Q}$	query set
$\mathcal{D}$	pairs set
$\mathcal{Y}$	label set/grade set
$\mathcal{S} = \{(q_i, \mathcal{D}_i), y_i\}$	dataset
$q_i \in \mathcal{Q}$	i-th query in a dataset
$\mathcal{D}_i = \{d_{i,1}, d_{i,2}, ..., d_{i,n_i}\}$	pairs set of the query q_i
$x_i = \Phi(q_i, d_{i,j})$	Feature vector from $(q_i, d_{i,j})$
$\mathbf{i}$	Set of possible rankings on $\mathcal{D}_i$ with regard to q_i
$\pi_\mathbf{i} \in \mathbf{i}$	Permutation on $\mathcal{D}_i$ with regard to q_i
$\mathcal{C}_i$	Set of combinations of 2 pairs on $\mathcal{D}_i$
$\equiv$	Modulo symbol
$\succ$	Ranking symbol
$\frown$	Concatenation symbol

mention for each dataset the number of the original questions and the number of pairs. For training and validation datasets, each query tag (Question) identified by QID relate to up to 30 QA pairs (QApair) identified by QAID while for the test dataset each query relates to up to 10 QA pairs. The query text is contained in Qtext while the question and answer are contained in QAquestion and QAanswer respectively, they are extracted by parsing the XML files. A QA pair contains two attributes, QArel which denotes the relevance of the pair regarding the query on a graded scale (I if irrelevant, R if relevant and D if direct). QAconf refers to the relevance of the answer with respect to the question which is used to rank the pairs by decreasing relevance.

Table 2.5: SemEval 2017 Task 3 Subtask D

Dataset	Questions	Pairs
training	1,031	30,411
validation	250	7,384
test	1,400	12,581

Given a new query and its related QA pairs, this task purpose is to predict these two attributes and therefore could be reduced to the following subtasks:

- Classification: Classify the QA pairs based on their similarity with the query.
- Ranking: Rank the QA pairs by decreasing similarity with regard to the query.

Table 2.6 reports the results obtained by the participating teams for SemEval 2017 Task 3 Subtask D. To the best of our knowledge, these works were the only ones published for the target dataset.
Table 2.7 shows the results obtained using the topic modeling approaches, SVM^{Rank}, and

Table 2.6: Semeval 2017 Task 3 Subtask D Competition Results

Model	MAP	Acc	P	R	F1
GU_QA	61.16	60.77	00.00	00.00	00.00
UPC_USMBA	57.73	66.24	63.41	33.00	0.4341
QU_BIGIR	56.69	49.64	41.59	70.16	52.22

Table 2.7: Test Dataset Results

Model	MAP	Acc	P	R	F1
LDA	53.58	63.06	60.94	16.25	25.66
NMF	56.42	62.52	58.95	14.75	23.59
LSA+NLTK	61.31	62.16	83.08	04.47	08.49
LSA+CoreNLP	61.66	62.34	85.87	04.80	09.09
SVM^{rank}	56.22	61.08	100	00.81	06.10
BiGRU-intersection	56.93	59.07	48.58	73.58	58.52
BiGRU-concatenation	48.53	39.23	39.23	100	56.36

Bidirectional GRU. Table 2.8 is a comparison between the pairwise approach and the pointwise approach for the DotNet and PyramidNet models.

For each architecture, BiGRU, DotNet, and PyramidNet, we report the parameters and the hyperparameters used during the training phase in Table 2.3.

2.5 Results & Discussion

The results show that the LSA-based model using the algorithm and parameters described in subsection 2.3.1 can lead to better results compared to the state of the art model for this task, using the Stanford CoreNLP tokenizer improves further the results. Though the insufficient evidence according to a Welch's t-test t(2797.7) = -0.35, $p < 0.363$ to support a true difference between our system (mAP = 0.6166) and the state of the art one (mAP = 0.6116) highlighted in Table 2.6, overall, according to a Welch's t-test t(25159) = -2.56 , $p < 0.005$, our system (Accuracy = 0.6234) outperforms the state of the art one (Accuracy = 0.6077). Using other topic modeling approaches such as NMF and LDA, for instance, leads to comparable accuracies but low mAPs.

Using SVM^{Rank}, the ubiquitous method for ranking, produced a high precision but a low mAP, this could be explained by the unbalanced nature of the dataset and the length of text chunks.

Table 2.8: Pairwise Vs Pointwise Results

Model	Pairwise					Pointwise				
	MAP	Acc	P	R	F1	MAP	Acc	P	R	F1
DotNet	53.71	56.68	43.88	37.34	40.35	52.44	56.04	42.93	36.53	39.47
PyramidNet	57.57	58.89	47.19	40.15	43.39	55.00	57.30	44.81	38.13	41.20

The BiGRU model is usually used to encode sentences, the encoded sentences are then fed to a dense layer for classification, a comparison between the results obtained using the intersection of the triplets as input are considerably better than those obtained using the concatenation as portrayed in Table 2.7, this could be explained by the ratio of the dataset size and the features vector size, deep learning methods require a huge training corpus in order to learn model weights while avoiding overfitting. Minimizing features using triplets intersection reduce also the training time.

The comparison between the pairwise approach and the pointwise counterpart as depicted in Table 2.8 shows that the former outperforms the latter for all evaluation metrics which confirms our initial hypothesis. For the two models, we have used the same architecture and applied the same hyperparameters.

To sum up:

- The LSA-based model was able to produce a good MAP following an unsupervised approach. Unlike the SemEval 2017 winning system, the LSA based model does not require external resources and therefore is more practical and more efficient for this task.

- The overfitting of neural networks models was partially reduced using pre-trained word embeddings trained on a curated list of medical domain articles, it was also alleviated by reducing the features space using the overlappings terms of the triplets instead of their concatenation.

- Using the pairwise approach for cQA helps to improve the results for deep learning models.

2.6 Conclusion

Community Question Answering systems are witnessing increasing popularity recently. Much of this interest stems from their capacity to represent a hub for communities to ask questions, seek information, and share knowledge around various topics in a trustworthy and timely manner. Content in these online fora is characterized by a high amount of redundancy, the presence of misspelled words, and usage of paraphrasing which leads to low accuracies.

Motivated by these aspects, in this chapter we addressed the problem of cQA for an Arabic medical domain forum by harnessing the connection between the original question and the thread question. Our main idea is to (i) adjust the LSA model to leverage the relationships between the triplet components (ii) use stemming to alleviate the impact of noise and misspelled words on the performance of the system (iii) use pairwise neural networks to improve the ranking performance.

Using the LSA model, we were able to obtain high values for the mAP, the accuracy, and the precision thus outperforming the winning system for this task. Using the pointwise approach for the BiGRU model, we were bound to use overlapping terms as features to get comparable performance to the pairwise approaches which confirms our initial hypothesis that pairwise methods perform better than pointwise methods.

In our future work, we plan to (i) devote our effort to leveraging the syntactic structure of sentences by using kernel trees to compute the similarity measures between constituency and dependency trees (ii) combine machine translation evaluation metrics such as Meteor, BLEU, TER with lexical features which would be of great value to harness the lexical similarity (iii) experiment with graph-based models using a graph presentation for the question and the related pairs of question answers combined with knowledge bases as a huge proportion of terms in the medical domain are written in English or transliterated from English (iv) use augmentation as a technique to tackle overfitting and unbalanced datasets (v) use automatic summarization and topic detection to alleviate the problem of questions and answers length by extracting the most relevant part of the text (vi) use the tri-letter root of words in parallel with the conventional word-based embeddings as most words in Arabic derive from a tri-letter verb.

CHAPTER 3

Community Question Answering using Neural Networks

3.1 Introduction

CQA purpose is to provide users pertinent answers to their questions by identifying, among a set (sometimes a thread) of question-answer pairs, questions that are similar to the original question which is a challenging task. The challenge is hard due to the use of different wordings which we tackle by using a *knowledge-based approach* using medical terminology extracted from Wikipedia and Dbpedia.

In other words, *CQA* asks, given a new question, denoted the original question, and a set of related question-answer pairs, each associated with one correct answer, to re-rank the question-answer pairs according to their relevance with respect to the original question.

When dealing with *CQA*, we should overcome the following challenges: i) the high density of medical terms, ii) the occurrence of English terms embedded within the Arabic texts, that could complicate the linguistic process of Arabic texts, iii) the relatively high overlapping of terms between the query texts and the texts of query/answer pairs and iv) the high rate of typos stemming from the nature of biomedical questions.

Our contributions are summarized as follows:

- We learn domain-specific distributional word embeddings for Arabic that are able to outperform existing pre-trained vectors.
- We combine lexical and a convolutional neural network to predict the relevance.
- We compare the performance using different pre-trained embeddings.
- We compare the performance using a cross-lingual approach.

- We compare the performance using state of the art models for sentence encoding.

After this introduction, the organization of the rest of the chapter is as follows: Section 3.2 presents with some details our method. In Section 3.3 our experimental setting is described. The results of such experiments are presented in Section 3.4. Finally, our conclusions are presented in Section 3.5.

3.2 Our Method

The task consists of answering the initial query using the thread as the space of possible answers. We simplify this setting considering the thread as an unordered set of question-answer pairs directly linked to the initial question. The task can be approached as classification, regression, or ranking problem. In our case we select the set of possible candidates, we assign a relevance score to each candidate and we rank them accordingly.

Fig.2.3 illustrates the architecture of our system where the input files for training, validation, and test datasets, each containing a number of original questions and related pairs of question/answer have been obtained from Semeval challenge as described in section 3.3. In order to face the evaluation of the cross-lingual experiments, all the Arabic texts (original question and question/answer pairs) were translated into English using Google Translate[1]. English texts were pre-processed, as Arabic ones, using the Stanford CoreNLP pipeline. After being pre-processed, tokenized, and augmented with part of speech tags, the output JSON files are combined with n-gram cosine measures to constitute the inputs of our neural network described thoroughly in 3.2.5. The pipeline was constructed using a slightly modified Stanford CoreNLP powered by Arabic language models and multi-threading to improve performance. The pipeline is applied to the inputs to extract tokens which are combined with lexical features, embedded, and trained using a neural network to compute the relevance of each pair related to the original question.

The embedding layer is powered by pre-trained word embeddings, its weights are frozen during the training of the model. At the output, we get the logits thereafter used as the confidence to rank the pair.

3.2.1 Word Embeddings

Our guess, confirmed in our experiments, see Section 3.4, was that learning our own embeddings from adequate sources could be better than using pre-trained generic embeddings. So we learned our embeddings as follows: A set of medical articles were collected from webteb, an Arabic medical website publishing articles for diverse topics such as general health, body parts, drugs, treatment, diseases, tests as reported in Table 3.1. The scraped html pages

[1]https://cloud.google.com/translate

Table 3.1: Webteb Articles

Section	Articles
Body Parts	87
Diseases	653
Drugs	699
Treatments	173
Tests	213
Vaccines	32
Vitamins	24
Recipes	480
Articles	4629
Q&A	3998
Others	783
Total	**10988**

Table 3.2: Arabic Text Normalization

Buckwalter	Isolated	End	Middle	Beginning
b	ب	ـب	ـبـ	بـ
t	ت	ـت	ـتـ	تـ
v	ث	ـث	ـثـ	ثـ
j	ج	ـج	ـجـ	جـ

are run through a pipeline where the result is used to train an unsupervised network in order to obtain pre-trained word embeddings. The process of learning these embeddings is presented in the following subsections.

3.2.1.1 Data Cleansing

The obtained HTML articles are cleaned in order to remove HTML tags and ads content then merged into a plain text file.

3.2.1.2 Text Normalization

Normalization is critical for Arabic as many symbols have various forms depending on context (beginning, middle or end) thus a semantic similarity analysis could fail if the text is not correctly normalized as different alphabet sequences are considered different though they refer to the same word. Table 3.2 illustrates examples of Arabic letter forms and their Buckwalter transliteration.

Other normalization rules were used for punctuation, digits, and diacritics.

Table 3.3: FastText Hyperparameters

Hyperparameter	Value
Learning Rate	0.1
Epochs	5
Dimensions	100
Vocabulary	78908
Type	Skipgram
Context Window	5
Negative Samples	5

3.2.1.3 Distributional Embeddings

To train word embeddings, we used FastText as described by [125] and [145] using the hyperparameters shown in Table 3.3.

Fig.2.1 and Fig.2.2 visualize the obtained embeddings using two different projections where the former is the Principal Component Analysis denoted PCA trying to capture as much of the data variability in a few dimensions as possible, we used three components for this purpose to project the obtained vectors in 3D. A word was selected Sodium/AlSwdywm/الصوديوم in order to display its neighbors which are its inflection forms, its translation, other similar minerals or drugs containing Sodium as raw material. The latter is the tSNE projection denoting T-distributed stochastic neighbor embedding which is a nonlinear non-deterministic algorithm that tries to preserve local neighborhoods. Similar words cluster in the same region of the projection where the same word "Sodium" was used as an example. In both visualizations, the mark size is proportional to the semantic similarity, with large balls for similar words and tiny circles for less similar ones. The corresponding term in English is added after a pipe for each Arabic word. We have also used publicly available embeddings, common crawl, and Wikipedia [2] which are domain-general vectors, and pyysalo medical domain vectors [3].

3.2.2 Preprocessing

The same normalization rules applied in Section 3.2.1.2 for Webteb corpus are applied for the training, validation, and test datasets.

3.2.3 Preprocessing Pipeline

We used Stanford CoreNLP[4] [146] to run our datasets through a pipeline in order to tokenize the text and get a POS tag for each token. The output file is a JSON file where the keys

[2]https://fasttext.cc/docs/en/crawl-vectors.html
[3]http://bio.nlplab.org/
[4]https://stanfordnlp.github.io/CoreNLP/

Table 3.4: JSON Format

Key	Description
qid	Question ID
qaid	Pair ID
question	Question Tokens List
qaquestion	Pair Question Tokens List
qaanswer	Pair Answer Tokens List
qpos	Question POS List
qaqpos	Pair Question POS List
qaapos	Pair Answer POS List
qarel	Pair Relevance
qaconf	Pair Confidence

with their descriptions are depicted in Table 3.4.

The pipeline is composed of sentence splitting, tokenization, and POS tagging steps where tokenization is done without removing stop words.

3.2.4 Features Engineering

3.2.4.1 Text Tokenization

The features used for representing the candidates to be ranked are described in this section.

Pair tokens list (question, qaquestion) are converted to numeric IDs after being padded with a start token <s> and an end token </s> while out of vocabulary tokens denoted OOV are replaced by <unk>. The size of the tokens is dynamic so no restriction on sentence length is imposed, batches are sorted by question length.

3.2.4.2 Ngram Cosine

We have used various lexical similarity measures such as Levenshtein, Jaro-Winkler, Longest Common Subsequence, Sorensen-Dice coefficient but Ngram cosine has given the most reliable results. Ngram cosine is a similarity measure used to determine if two words or sequences are lexically similar based on the n-grams extracted from the two sequences. For our model, we have chosen 3-gram.

3.2.5 Our Model

Our model, denoted KCNN, as illustrated in Fig.3.1, is a multi-input and multi-layers neural model. The inputs are sequences of tokens for the Question denoted qaquestion and n-gram cosine lexical similarity measure. The two first inputs are embedded using pre-trained word embeddings if the token belongs to the vocabulary and randomly initialized otherwise, the embedding layer weights are frozen during training. The embedding layer is followed by a

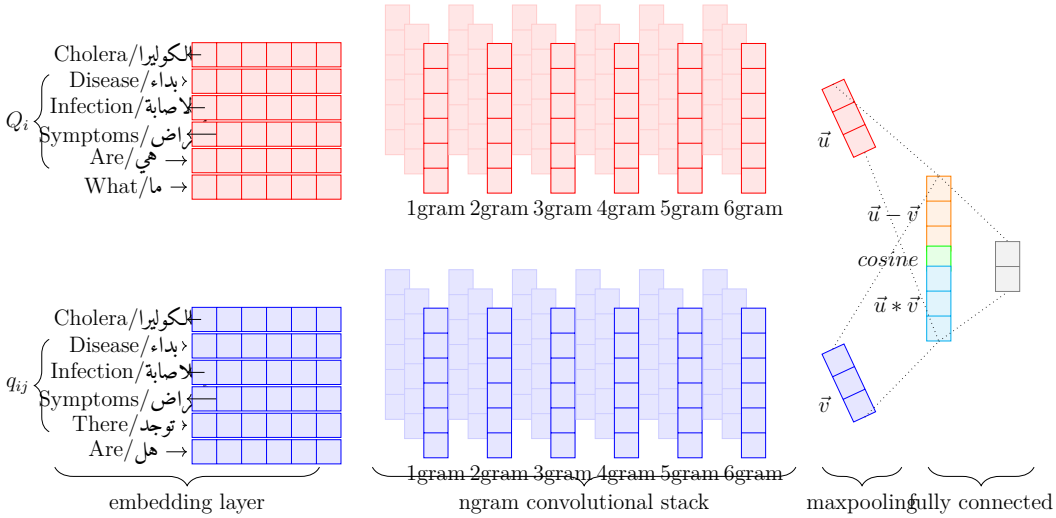


Figure 3.1: Neural Network composed of the input layer where both the question Q_i and the pair question q_{ij} are embedded using pre-trained embeddings, followed by a stack of convolutional networks then a fully connected layer with a softmax.

Table 3.5: K-CNN Hyperparameters

Hyperparameter	**Value**
Epochs	50
Batch Size	128
Filters	64
Kernels	k
Dropout	0.7
Learning Rate	0.01
Fully Connected Units	300

stack of six CNN for the two first inputs, each CNN purpose is to detect a k-gram pattern where $k \in \{1, 2..6\}$. CNN hyperparameters are indicated in Table 3.5.

Each convolution layer is followed by a RELU activation function, then, a max-pooling layer. We denote $\vec{u}$ the output for question $\mathbf{Q_i}$ and $\vec{v}$ the output related to qaquestion $\mathbf{q_{ij}}$. We denote $\vec{\mu}$ the output of the merged layer concatenating the following vectors:

- The absolute value of $\vec{u} - \vec{v}$.
- The dot product of $\vec{u}$ and $\vec{v}$.
- The lexical feature denoted $\vec{l}$ obtained by applying normalization, linear, and a RELU function to the third input which is n-gram cosine.

$$\vec{\mu} = |\vec{u} - \vec{v}| \frown \vec{u} * \vec{v} \frown \vec{l} \tag{3.1}$$

given that $\frown$ is the concatenation operator. We apply a fully connected layer to the output from the previous one which is a sequence of dropout, normalization, linear, relu,

dropout, normalization, linear, and finally a softmax in order to obtain the logits of each output class: 1 if relevant and 0 otherwise.

3.2.6 Cross-lingual Approach

We use Google Translate API to translate the original questions and question-answer pairs, then we apply the standard Stanford CoreNLP API for splitting and tokenization. We use the same model as for Arabic. We power the embeddings layer using publicly available word vectors to embed the tokens including Pyysalo PubMed and PMC [5] and publicly available for both FastText and Glove.

3.2.7 Sentence Encoders

Sentence encoders are used to transform sentences into vectors that encompass their semantic meaning. They are used for many tasks where sentence semantic matching is critical such as machine comprehension, answer selection, and language inference. We use for that matter state of the art sentence encoders:

- Bilateral Multi-Perspective Matching for Natural Language Sentences denoted BiMPM [147] matching two sentences using a pair of bidirectional LSTMs.
- Enhanced LSTM for Natural Language Inference denoted ESIM [148] based on chain LSTMs.

3.3 Experimental Setup

The system presented in Section 3.2 has been applied to the datasets described in Section 3.3.1. We present in the following subsections details on the different aspects of the experiments and their evaluation.

3.3.1 Dataset Description

To have a fair evaluation of our approach we attended the Semeval 2016 and 2017 contests, specifically the Task 3, Subtask D: Arabic Community Question Answering and we used the datasets in the research described here. Semeval cQA threads are simpler than the general ones described in Section 3.1 and depicted in Fig. 1.1. Compare this figure with the corresponding to Semeval in Fig. 1.2. In Semeval scenario threads are simple: An initial query is followed by parallel Question/Answer pairs, so paths consist of only three nodes. So the topology of the threads is practically useless. No information is provided about the

[5] http://bio.nlplab.org/

community members, so member's profiles and behavior are unknown and so cannot be used as features. For the general case of disposing of such information [149] discusses the use of non-textual features for the task.

SemEval 2017 Task 3 purpose is to help automate this process to find good answers to new questions thus shrinking the time and efforts needed.

Subtask D targets medical fora where users ask questions in Arabic related to the medical domain. Three datasets are available, the training and validation dataset where each original question is related to a maximum of 30 question-answer pairs, and the test dataset where each original question is related to a maximum of 10 pairs. The count of original questions and pairs contained within each dataset is illustrated in Table 1.1.

3.3.2 Training Phase

The training phase is performed on a Tesla K80 GPU of 13 GB RAM using the training dataset split into batches of size 128, Cross Entropy Loss denoted $\mathcal{L}$ as the loss function:

$$\mathcal{L} = -(ylog(\hat{y} + (1-y)log(1-\hat{y}))) \tag{3.2}$$

where y is the relevance of the pair and $\hat{y}$ is the predicted value obtained as logits from our model.

The training objective is to minimize the loss by using backpropagation, Adam, [144], as the optimizer, and [150] as the weight initialization scheme.

Figure 3.2 illustrates the evolution of the accuracy respectively for the training and validation sets using various stacks of n-grams where the x-axis represents iterations which are combining epochs and batches varying from 0 to 12,000 while the y-axis represents the loss for the former and the accuracy for the latter.

3.3.3 Validation Phase

To fine-tune the network hyperparameters, the model is evaluated on the validation set for each 200 iterations. The hyperparameters depicted in Fig. 3.5 are chosen in a way to get the optimal accuracy. The linear layer applying an affine transformation to the inputs was interleaved by dropout layers where p=0.7 to combat overfitting. $k \in \{1..6\}$ was chosen over $k \in \{1..5\}$, $k \in \{1..4\}$ and $k \in \{1..3\}$ as it has produced low values for the loss and high values for the accuracy as illustrated in Fig.3.2 respectively. Batch normalization was used to help the model converge quickly.

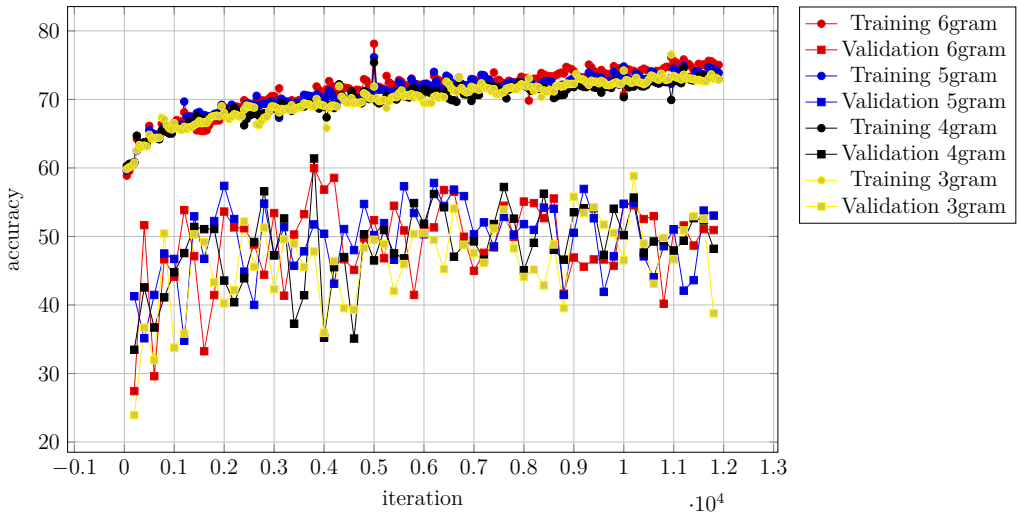


Figure 3.2: Training Accuracy for different stacks of kernels. The optimal values are obtained using up to 6gram $k \in 1, 2, .., 6$ where the accuracy plotted using red has the maximum values.

3.4 Results & Discussion

The evaluation metrics reported in Table 3.6 are regrouped into four groups, the first group is the official results of the SemEval 2017 task 3 subtask D task which targets an Arabic Medical dataset.

The second group represents the results obtained using the model presented in this chapter. KNN+COSINE+FastText outperformed the state of the art model with regards to accuracy by almost 3 points and competitive performance for the MAP. These results were obtained using our pre-trained vectors and a frozen embeddings layer. KCNN+COSINE has the same architecture except that the embedding layer weights are learned during training, the performance of this model has dropped which justifies the benefit of using pre-trained embeddings. The intuition behind using this architecture could be justified by the high degree of inflection characterizing the Arabic language. Besides its high performance compared to sequence architectures such as RNNs, LSTMs and, GRUs, a stack of CNNs can detect occurrences of k-grams where k varies from 3 to 6 which was able to reduce the effect of inflection.

Using publicly available pre-trained vectors contributes to the performance of the model but using our embeddings leads to better results which could be explained by two facts. On the one hand, training our embeddings reduces the out of vocabulary terms by applying the same splitting and tokenization rules. On the other hand, using a specific-domain corpus for training contribute to noise reduction and performance optimization.

The third group depicts the cross-lingual approach results. Using pyysalo medical pre-trained vectors improves drastically the performance.

The last group corresponds to the state of the art encoders performance. These models

have been demonstrated to be efficient for sentence matching and entailment but lead to weak results compared to the proposed model, this could be justified by the characteristics of Arabic as these models were used initially for English.

Table 3.6: SemEval2017 Task3 SubtaskD Results

Model	MAP	AvgRec	MRR	P	R	F1	Acc
GW-QA [151]	61.16	85.43	66.85	55.63	55.63	64.75	66.92
UPC-USMBA [43]	57.73	81.76	62.88	63.41	33.00	43.41	66.24
QU-BIGIR [152]	56.69	81.89	61.83	41.59	70.16	52.22	49.64
KCNN+COSINE+FastText (Arabic)	60.12	83.98	65.80	62.53	55.25	58.66	69.45
KCNN+COSINE (Arabic)	54.22	79.13	59.35	54.89	23.93	32.93	62.41
KCNN+COSINE+FastText (Wikipedia&CommonCrawl)	57.83	82.01	63.66	57.24	56.16	56.69	66.34
KCNN+COSINE+FastText (English)	57.34	81.79	62.68	55.76	63.96	59.58	65.95
KCNN+COSINE+GloVe (English)	58.09	82.37	63.64	55.12	67.28	60.60	65.67
KCNN+COSINE+Pyysalo-PubMed-PMC (English)	58.19	82.25	63.51	59.09	58.29	58.68	67.79
KCNN+COSINE+Pyysalo-PubMed (English)	58.79	82.79	64.34	61.54	49.92	55.12	68.11
ESIM	56.34	81.01	60.98	53.69	71.56	61.35	64.63
BiMPM	47.23	72.32	50.96	39.23	1.00	56.36	39.23

3.5 Conclusion

We have demonstrated that using pre-trained word embeddings, a stack of CNN neural networks to detect up to 6-gram and a 3-gram cosine lexical feature, significantly improve the accuracy, the MAP was also improved since our official contribution by two and a half points and could be further improved by applying the same architecture in a pairwise setting to rank combinations of question/answer pairs related to the same original question to deduct their ranking.
To sum up:

- The proposed model is a stack of convolutional neural networks combined with lexical features that leads to state-of-the-art results.

- Our pre-trained vectors learned on an Arabic medical domain corpus contributes to the performance of the model.
- Using a cross-lingual approach combined with pyysalo pre-trained vectors leads to satisfactory results.
- Using state of the art sentence encoders fails to achieve competitive performance.

To extend our work, we intend to:

- Use a knowledge base for medical vocabulary to expand both the original question and the corresponding pair of question-answer.
- Performing a dependency parsing of Arabic texts and using the dependencies as an alternative to n-grams for building the convolutional layers.
- Use a graph approach to harness the syntactic structure of sentences.
- Use machine translation evaluation metrics as features to measure the similarity between the original question and the question-answer pair.
- Apply extractive summarization to sentences to reduce the features space dimensionality.
- Due to the rich morphology of Arabic words, using a channel of character embeddings as a complement to word embeddings.
- A similar approach could be a channel of root (triliteral) embeddings, taking profit of one characteristic of the Arabic language: most content words are derived from a triliter verbal root. So, the root of a word can be considered as a code of its meaning.

Using distributional embeddings combined with deep neural networks has improved neatly results, in the next chapter we deep dive into contextualized embeddings and demonstrate how context improves further our previous results.

CHAPTER 4

Community Question Answering based on Contextualized Embeddings

4.1 Introduction

Despite the effectiveness of previous studies, the mean average precision mAP has witnessed steady variations, these low performances could be explained by various factors including 1) the scarcity of resources characterizing the Arabic language 2) usage of English terms implied by the domain of the dataset 3) the difference of length between the original and the pair questions introduced by paraphrasing 4) the unbalanced nature of the dataset 5) misspellings and typographical errors characterizing CQA platforms as a consequence of seeking a quick answer 6) the absence of punctuation.

The motivation of this chapter is to leverage a bunch of contextualized embeddings which has proven effective for a wide range of downstream tasks such as textual entailment, paraphrasing identification, and reading comprehension by fine tunning them for the CQA task. Although contextualized embeddings are trained on huge corpora of text, they can harness the context of each word leading to better performances.

The main contributions of this work are six-fold:

- We leverage contextualized language representation models.
- We use a cross-lingual approach to tackle a cQA task in Arabic as most resources are readily available in English.
- We demonstrate the impact of sentence maximum length on performance.

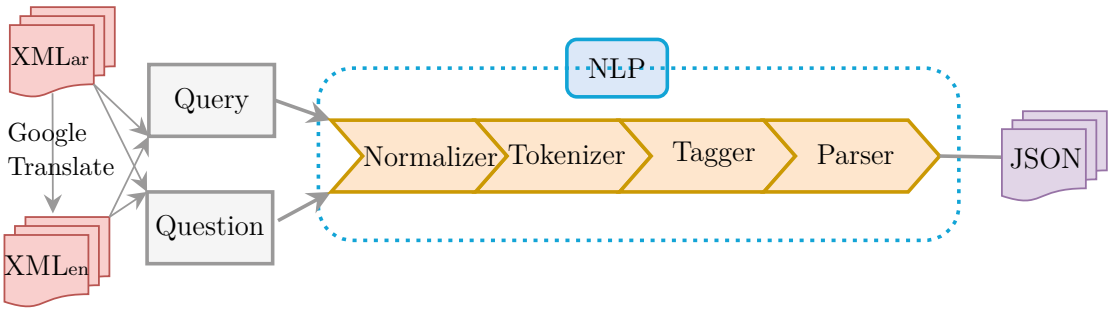


Figure 4.1: The preprocessing step of our model: the input XML files are parsed to extract the original question and the question-answer pair thereafter processed using a pipeline composed of a normalizer, a tokenizer, a tagger and a parser.

- We benchmark four contextualized language representations on two community question answering tasks.
- We demonstrate that the performance of contextualized embeddings is superior to context-free embeddings.
- Extensive experiments have been conducted to evaluate the performance of our model, these experiments show a significant improvement over the state-of-the-art models for CQA-MD [153].

The rest of this chapter is organized as follows: In Section 4.2, we present the followed methodology. The experimental setup is introduced in Section 4.3 and the obtained results are discussed in Section 4.4. Finally, the conclusions are presented in Section 4.5.

4.2 Our Method

We propose a model combining two neural networks, the first to obtain sentence representations while the other is used as a classifier. Our model as depicted in Fig.4.2 is constituted of three components 1) a preprocessing step using a natural language processing pipeline as depicted in Fig.4.1 2) an embeddings layer fed by a context-free or a contextualized embeddings model 3) a classifier fed by embeddings obtained by mapping the input dataset to the embedding layer.

4.2.1 Pre-processing

As depicted in Fig.4.1, we use Google Translate API to translate our XML datasets from Arabic to English, this is highly motivated by the number of linguistic resources available in English. We use Stanford CoreNLP for normalization and tokenization and store results in the JSON lines format.

4.2.2 Sentence Representation

We use BERT, XLNet, RoBERTa, and XLM to encode the original question and the question-answer pair. The first three transformers are applied to the original version of our dataset while the last one is applied to the translated version. The input is constituted of three components as shown in Fig.4.3:

1. input IDs denoted $\mathcal{S}$ are sequences of tokens offsets in the vocabulary file, surrogate tokens are added to the vocabulary, [PAD] to denote padding which is used when the original question or the question-answer pair lengths is less than the chosen maximum length, [SEP] to denote the separator between the original question and the question-answer pair while [CLS] is used by the encoder for classification tasks to denote the final hidden state.

2. attention masks denoted $\mathcal{A}$ used to distinguish between true tokens and surrogate tokens such as those used for padding.

3. input type IDs denoted $\mathcal{T}$ are used to delimit tokens belonging to the original question from those belonging to the question-answer pair.

Paddings are used to adjust short sequences to the maximum length while truncating is used to reduce the length of sequences to the maximum length. The maximum length is chosen based on performance of the model when applied to the evaluation dataset.

4.2.3 Model Implementation

The classifier is a multi-layer neural network pre-trained on huge volume of text. The network is composed of up to 24 layers, thousands of hidden nodes, and up to 16 heads, we report in Table.4.1 the hyperparameters used for each model which lead to the best results. The weights are further fine-tuned for the task to optimize the logarithm of the likelihood for the class C.

The output layer of the classifier is a score used for both tasks i) ranking the question-answer pairs related to the original question ii) determining the relevance of the question-answer regarding the original question using 0.5 as a threshold. In this thesis, we use contextualized embeddings to get the original question and the question-answer pair representations. We first apply a pipeline of pre-processing steps to clean text, an encoding step to get the original, and the question-answer pair representations which are used to get a similarity score of each question-answer pair regarding a specific original question.

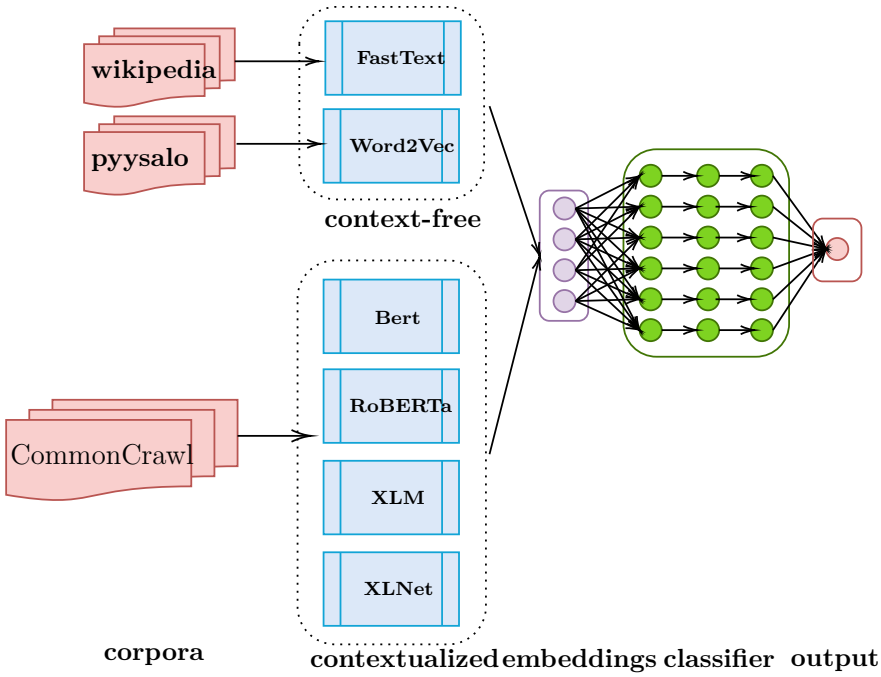


Figure 4.2: The model architecture: Context-free embeddings and contextualized embeddings are fed into a neural network classifier to output a score for each question/answer pairs.

Table 4.1: Model Hyperparameters

Model	Layers	Nodes	Heads	Weights(M)	LR	Epochs	Batch Size
BERT	12	768	12	110	2e-5	3	8
XLNet	24	1024	16	340	2e-5	10	16
XLM	12	2048	16	NA	2e-5	10	16
RoBERTa	24	1024	16	355	3e-5	10	8

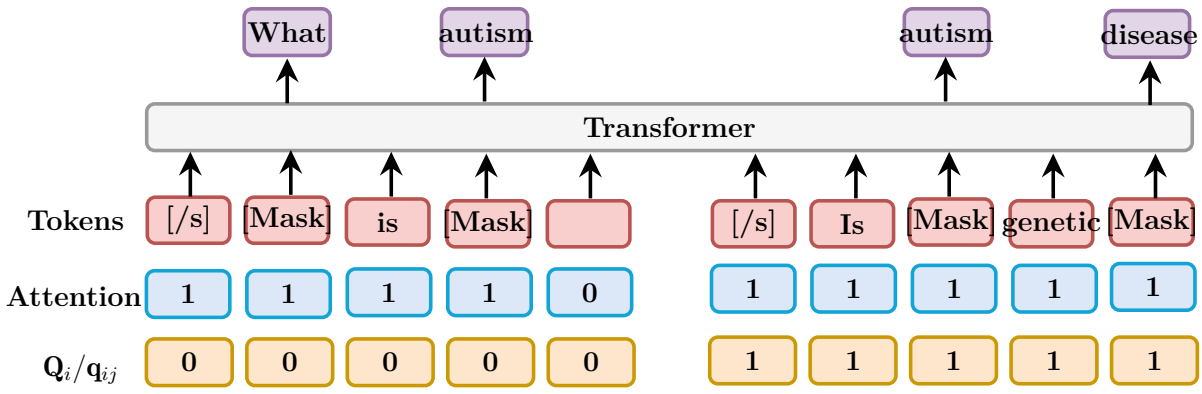


Figure 4.3: The fine-tuning step of our model: the transformer is fed with three inputs, the token type, the attention flag and the token id. The transformer is trained to recognize masked out tokens.

Table 4.2: Semeval 2016 & 2017 CQA-MD datesets

Category	Train	Dev	Test 2016	Test 2017
Questions	1,031	250	250	1,400
QA Pairs	30,411	7,384	7,369	12,581
Direct	917	70	65	891
Related	17,412	1,446	1,353	4,054
Irrelevant	12,082	5,868	5,951	7,655

4.3 Experimental Setup

4.3.1 Datasets description

We conduct our experiments on two widely used datasets for Arabic community question answering for biomedical domain CQA-MD, SemEval-2016 [154] and SemEval-2017 [112] Task 3 Subtask D. These datasets were extracted from fora of Arabic biomedical domain such as Altibbi and Webteb. The choice of these datasets is guided by different factors i) scarcity of benchmark datasets for Arabic ii) the dataset annotation quality controlled using hidden sets iii) biomedical cQA is of paramount interest as most Arabic users favor anonymity while asking taboo questions. The task purpose is to match an original question as submitted by a user to a set of question-answer pairs retrieved from the history of the fora as a potential answer to the user question. Table 1 depicts an example of an original question, a potential pair, a relevant pair, and an irrelevant pair. Table 4.2 depicts the count of original questions, pairs, direct, related, and irrelevant instances for CQA-MD.

4.3.2 Baseline Models

- **SLS** uses a bag of vectors composed of lexical, topic modeling, and deep learning approaches including convolutional and long short term memory neural networks with pre-trained embeddings to tackle community question answering.
- **ConvKN** combines tree kernels, functions used to measure the similarity between syntactic trees, with convolutional neural networks, lexical similarity, and thread-specific features.
- **RDI_team** combines three modules including a TF-IDF based module, a language modeling module, and a wikipedia based module, a score is obtained from the weighted sum of each module applied to the original and the question-answer pair.
- **QU-IR** uses word embeddings combined with a learning to rank model using gradient boosted trees and random forests.

- **UPC_USMBA** uses knowledge bases extracted from dbpedia to feed a logistic regression model for SemEval-2016 and topic modeling for SemEval-2017.
- **GW_QA** uses weighted textual matrix factorization and a set of lexical features bases on string length and surface-level matching to rank pairs w.r.t the original question.
- **QU_BIGIR** uses lexical similarity combined with word embeddings to learn an SVM classifier.

4.3.3 Sequence length

Fig.4.5 and Fig.4.6 depict the effect of the chosen sequence length on the ranking metrics (mAP, AvgRec, and MRR) and the classification metrics (Accuracy, Precision, Recall, and F1-measure) respectively. The figures show that the optimal question and answer length for contextualized embeddings is 220 henceforth used for all experiments.

4.3.4 Obtained Results

Table 4.3 and Table 4.4 depict results obtained by applying different embeddings and architectures on the CQA-MD for 2016 and 2017 tasks respectively followed by results obtained during the contest. Table 1 contains the results using i) fasttext trained on Arabic Wikipedia ii) fasttext trained on Arabic Common Crawl iii) pyysalo obtained by training word2vec on English PubMed and PMC texts iii) BERT-Base, Multilingual Cased finetuned for CQA-MD tasks iv) XLNet fine-tuned for CQA-MD tasks v) XLM fine-tuned for CQA-MD tasks. Fig.4.4 shows the training loss while finetuning the contextualized embeddings (BERT, XLNet, XLM, and RoBERTa) using 1200 iterations.

4.3.4.1 Ranking Results

For Semeval 2016 we were able to outperform the winning model by almost 2 points for the mAP, the official metric. This performance was achieved using RoBERTa as a sentence representation. Using BERT has also outperformed the winning model by 1 point. This latter has also lead to better AvgRec and MRR. Both models have also lead to superior performances for Semeval 2017 where mAP was improved by 1 point.

4.3.4.2 Classification Results

Using RoBERTa, we were able to outperform the official results by 1 point and more than 14 points regarding the F1 measure for both editions while obtaining an improvement of almost two points for the accuracy using Pyysalo and BERT.

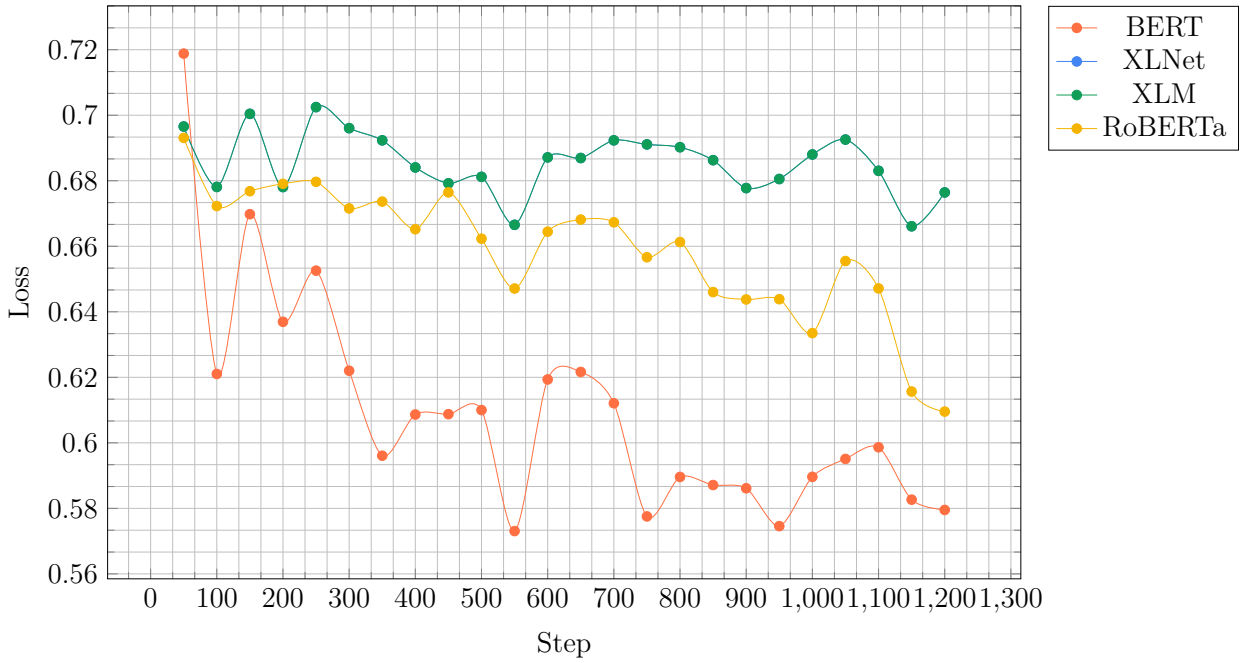


Figure 4.4: Training loss value change with respect to the iteration steps.

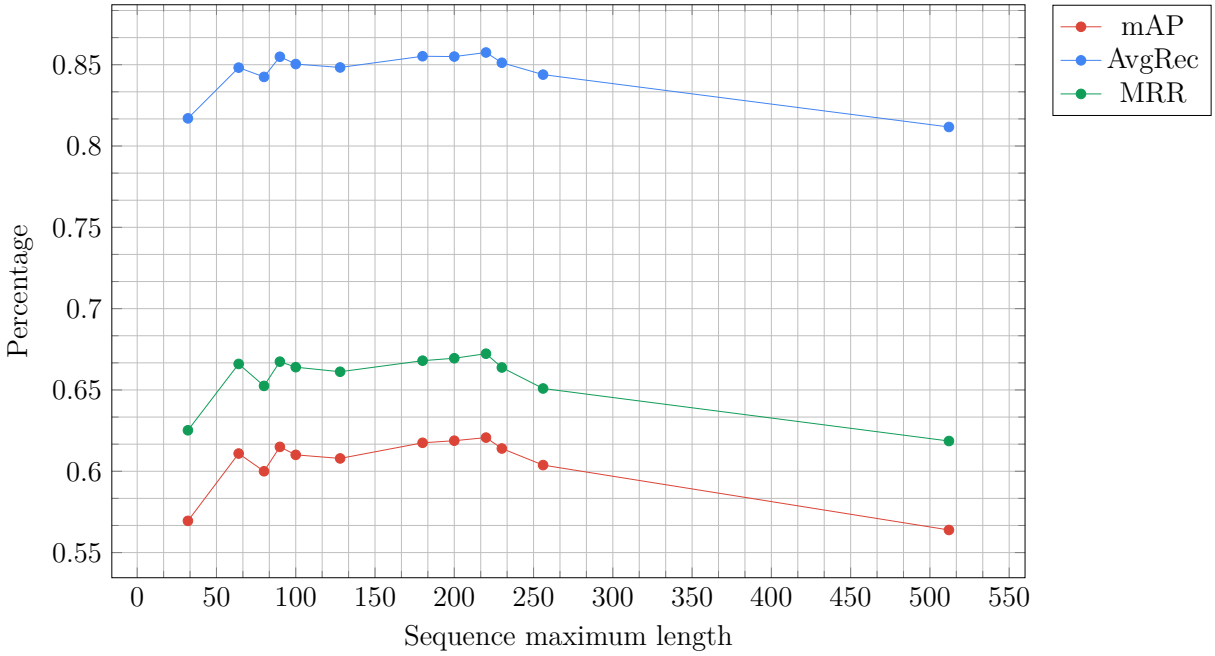


Figure 4.5: The effect of sequence length on ranking metrics

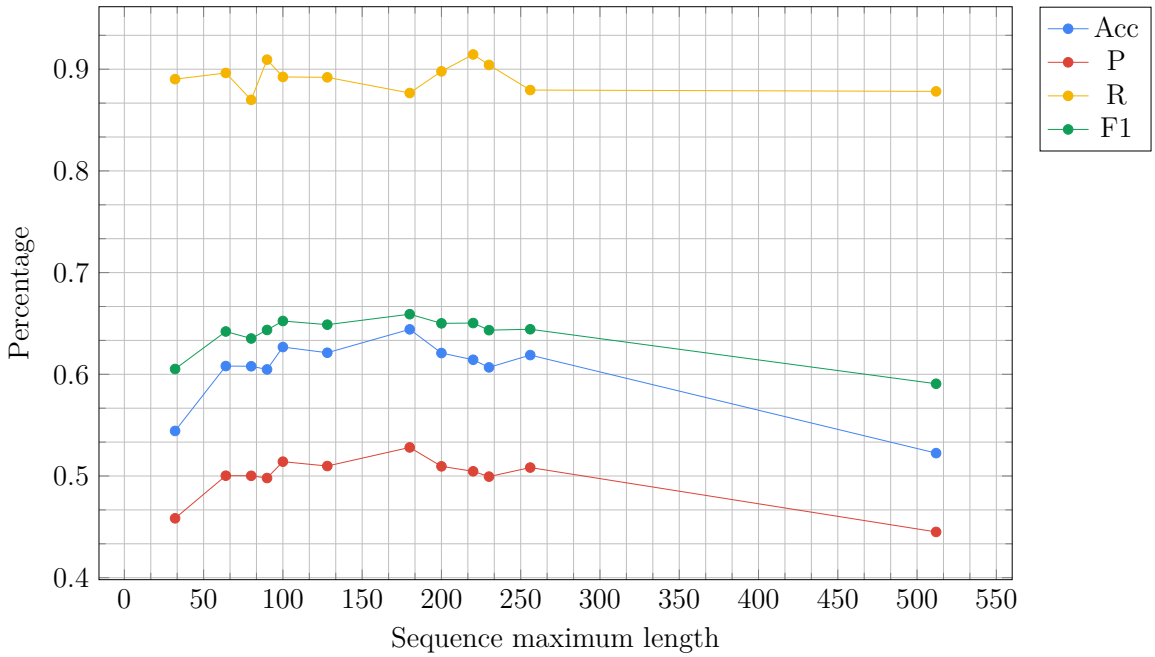


Figure 4.6: The effect of sequence length on classification metrics

4.4 Results & Discussion

Using contextualized embeddings, BERT for instance, we were able to improve the mAP, the official score for CQA-MD by almost one point for the 2017 test set and two points for the 2016 test set. These performances could be explained by their ability to take into account the context during the training and finetuning of the model, corpora used for pre-training, and the number of hidden layers and nodes used. XLM bad results could be explained by the language of the model used, better results could be obtained by pre-training XLM on Arabic and English medical domain corpora. Using context-free embeddings pre-trained on general domain corpora leads to inferior results compared to those pre-trained on specific domain corpora as the latter reduces the number of out-of-vocabulary terms.

The weak performance of the system regarding the mAP and Accuracy could be explained by many factors i) the skewness of the dataset as the majority of question-answer pairs are irrelevant to the original question ii) the length of the original question and the question-answer pair is non-proportional iii) the presence of typos in text fragments leading to many out-of-vocabulary terms, these factors could be overcome by i) pre-training the contextualized models on biomedical corpora instead of using pre-trained models ii) extracting

[-6]http://knowitall.cs.washington.edu/paralex/
[-5]https://github.com/abachaa/MedQuAD
[-4]http://alt.qcri.org/semeval2016/task3/
[-3]https://github.com/abachaa/RQE_Data_AMIA2016
[-2]https://www.kaggle.com/quora/question-pairs-dataset
[-1]https://askubuntu.com/
[0]https://www.kaggle.com/quora/question-pairs-dataset

Table 4.3: Semeval 2016 ranking and classification results

		Ranking			Classification			
	Submission	MAP	AvgRec	MRR	P	R	F1	Acc
1	Wikipedia	43.73	46.93	50.00	26.73	76.80	39.66	55.03
2	Pyysalo	43.05	47.67	49.64	27.80	81.66	41.48	55.65
3	BERT	46.83	52.41	53.83	30.37	65.30	41.46	64.51
4	XLNet	45.05	51.37	51.52	25.33	96.12	40.09	44.71
5	XLM	46.83	52.41	53.83	30.37	65.30	41.46	64.51
6	RoBERTa	47.67	51.74	51.75	27.56	93.51	42.57	51.45
7	SLS-primary	45.83	51.01	53.66	34.45	52.33	41.55	71.67
8	ConvKN-primary	45.50	50.13	52.55	28.55	64.53	39.58	62.10
9	RDI_team-primary	43.80	47.45	49.21	19.24	100.00	32.27	19.24
10	QU-IR-primary	38.63	44.10	46.27	25.50	45.13	32.59	64.07
11	UPC_USMBA-primary	29.09	30.04	34.04	20.14	51.69	28.99	51.27

Table 4.4: Semeval 2017 ranking and classification results

		Ranking			Classification			
	Submission	MAP	AvgRec	MRR	P	R	F1	Acc
1	Wikipedia	56.82	81.21	62.73	58.28	49.55	53.56	66.29
2	Pyysalo	58.79	82.79	64.34	61.54	49.92	55.12	68.11
3	BERT	62.07	85.75	67.23	59.33	55.57	57.39	67.63
4	XLNet	59.46	84.02	64.81	51.54	83.63	63.77	62.72
5	XLM	49.52	74.86	53.83	40.29	76.26	52.72	46.34
6	RoBERTa	61.19	84.96	66.26	54.83	85.80	66.90	66.70
7	GW_QA-primary	61.16	85.43	66.85	0.00	0.00	0.00	60.77
8	UPC-USMBA-primary	57.73	81.76	62.88	63.41	33.00	43.41	66.24
9	QU_BIGIR-primary	56.69	81.89	61.83	41.59	70.16	52.22	49.64

key sentences from the original question and the question-answer pair iii) using knowledge bases to annotate text before proceeding to similarity computation.

4.5 Conclusion

Community question answering systems are springing up due to their capacity to address health information needs conveniently. CQA for the medical domain helps users seek information to diagnose diseases by describing symptoms and requesting information about drugs and medical tests.

Motivated by these facts, we addressed in the present work the problem of community question answering applied to Arabic medical domain datasets by leveraging the power of state of the art contextualized embeddings. Our method is harnessing contextualized embeddings to compute the similarity between the original question and the question-answer pair in order to rank the pairs related to a specific question on one side and to classify pairs as relevant or irrelevant on the other side.

Using contextualized embeddings as sentence representations leads to better results as they are context-aware and able to measure similarity at the semantic level bypassing syntactic errors or paraphrasing usage.

Our model has proven successful for both cQA tasks Semeval 2016 and Semeval 2017. We were able to improve the former by 2% and the latter by 1% regarding the mAP which is the official metric.

While using contextualized-embeddings has lead to satisfactory results, these embeddings neglect the hierarchical structure of the text, in the next chapter, we present our model CoNN combining constituency trees with neural models to improve further our results.

CHAPTER 5

Tree-Based Approach to Community Question Answering

5.1 Introduction

The main goal of cQA platforms is to determine for a specific question denoted the original question, the most relevant questions amid a set of questions answered beforehand in this platform ranked by decreasing relevance, Fig.1.1 shows an example of a cQA thread.

cQA is often cast as a two-fold problem: classification and re-ranking. The former could be framed as given an original question and a question-answer pair, the problem lies in classifying the relationship between them into one of the three classes: 'direct', 'relevant' or 'irrelevant' While the latter could be framed as given a new question and a set of question-answer pairs, re-rank the correct answers related to the original question. Specifically, we use syntactic analyses to accurately assess the similarity between the original question and the question-answer pair and harness recursive neural networks to boost the performance of the platform.

Earlier efforts have focused on applying engineered features as input to machine learning models such as support vector machines (SVM) and random forests. These features include frequency features (tf-idf, BM25) lexical similarity measures (Jaccard and Sorensen coefficients), machine translation based scores such as BLEU.

Despite their weak performances, they are usually time-consuming. Today, the focus has shifted to neural-based approaches as they can model any Borel-measurable function without going through the labor-intensive feature engineering step. The major downside of neural networks is the scarcity of annotated data, a drawback usually exacerbated for languages such as Arabic suffering scarcity of resources and analyzers.

Notwithstanding the effectiveness of previous studies, the potential of graph-based approaches has been seldom explored despite the hierarchical tree structure of documents.

This could be alleviated by using constituency or dependency trees as input to shallow or deep machine learning models.

The motivation of this thesis is two folds, rank question-answer pairs related to an original question and classify a question-answer pair as relevant or irrelevant to the original question. We leverage syntactic analyses and recursive neural networks to rank question-answer pairs related to an original question i) we use constituency trees generated for the original question and question-answer pair as input to a recursive neural network based on a shift-reduce approach. Using syntactic parse trees enables the model to learn combinations of subtrees that are able to predict the similarity between the original question and the question-answer pair while using the shift/reduce approach boil down to more convenient batched operations leading to increased performance.

ii) we apply subset tree SST kernel and partial tree PT kernel as described in [118]

on dependency trees generated from the question and the question-answer pair to compute a score which is later combined with lexical similarity measure and fed to a ranking SVM model. iii) A combined approach using constituency neural networks for ranking and subset tree kernel for classification. The results of these approaches are reported for a widely used dataset for Arabic biomedical community question answering Semeval CQA-MD leading to state-of-the-art results for the official metric.

To sum up, the main contributions of this work are three-fold:

- We apply two widely used tree kernels.
- We train a neural network on constituency trees.
- We report the performance on a widely used cQA dataset for the Arabic biomedical domain using a combined approach leading to superior results compared to state-of-the-art models.

The remainder of this thesis is structured as follows. Section 1.5 reviews and discusses related work. In Section 5.2, we fully describe the followed approaches. The experimental setup is depicted in Section 5.3. Section 5.4 shows the evaluation and discusses the obtained results while Section 5.5 makes the conclusions and discusses the future work.

5.2 Our Method

Community Question Answering could be defined as, given a question denoted the original question and a set of threads, each composed of a question and an answer, rank the threads based on their similarity related to the original question. cQA are usually subdivided into two subtasks:

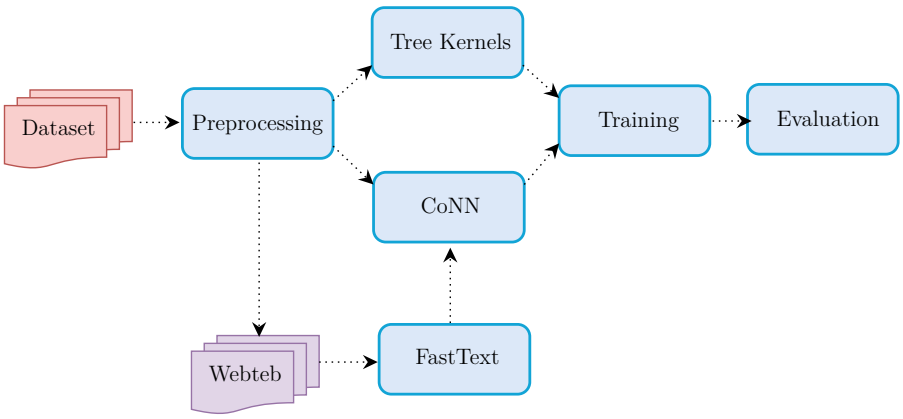


Figure 5.1: The architecture main components are the preprocessing step, tree kernels and the constituency neural network fed by embeddings generated from the webteb corpus.

- Classification: classify each thread as relevant, irrelevant, or perfect match related to the original question.
- Ranking: rank the threads based on their similarity related to the original question.

More formally, a cQA task could be defined as a tuple $(\mathcal{Q}, \mathcal{D}, \mathcal{Y})$ where $\mathcal{Q}$ is a set of original questions, $\mathcal{D}$ a set of question-answer pairs, and $\mathcal{Y}$ a set of labels.

$$\mathcal{Q} = \{q_i\}_{i=1}^{n} \tag{5.1}$$

$$\mathcal{D}_i = \{(Q_{ij}, A_{ij})\}_{j=1}^{m_i} \tag{5.2}$$

$$\mathcal{Y}_i \in \{Relevant, Irrelevant, Direct\}^1. \tag{5.3}$$

Where:

- n is the number of original questions.
- m_i is the number of pairs related to the query q_i

In order to tackle cQA for the Arabic Biomedical domain, we use the architecture shown in Fig.5.1. We use a pre-processing module for sentence splitting, tokenization, part-of-speech, and parsing. The preprocessed dataset is ingested by two models, one using two of the most used tree kernels while the other is a neural network powered by an embedding layer where the embeddings are frozen during the training phase and are obtained applying the FastText algorithm on a collection of Arabic biomedical documents which are preprocessed beforehand using the same steps as for the task dataset. The hyperparameters and the

[1]We employ frequently the abbreviate forms R, I and D

model weights are fine-tuned during the training phase while the output file is produced during the evaluation phase.

5.2.1 Preprocessing

The pre-processing step is of paramount importance as community question answering tasks are characterized by the occurrence of many typos and grammatical errors.

The question and question-answer pair are pre-processed using normalization, sentence splitting, tokenization, part-of-speech tagging, constituency parsing, and dependency trees. The obtained constituency trees are then processed to generate equivalent binary trees. These parse trees are converted to SHIFT/REDUCE sequences as input to the model, their symmetric aspect is more convenient for batching.

The normalization step objective is to obtain the canonical form of text by getting rid of extraneous characters, ligatures that are used by some users to emphasize some words, diacritics which are irrelevant for this task and transform different forms of some letters to a unique form such as forms implied by position in a word or those implied by syntactic rules. Fig. 4.1 depicts the different steps of the NLP pipeline.

5.2.2 Constituency Neural Network

Our Constituency Neural Network (CoNN) neural model could be divided into five components, the input layer, the embeddings layer, the RNN layer, the MLP layer, the tracker layer, and the output layer. The binary trees obtained from original questions and those obtained from question-answer pairs are grouped into batches and fed into a neural model. The embeddings layer use pre-trained embeddings that are frozen during the training of our model. We use a shift/reduce strategy to simplify batched operations using a buffer and a stack. The buffer contains the tokens of the input sentence and the stack is initially empty. During a shift operation, a token is moved from the buffer to the stack while the reduce operation combines the top two elements of the stack $\vec{c_1}$ and $\vec{c_2}$ into a single element $\vec{p}$.

$$\vec{p} = \rho(W[\vec{c_1}, \vec{c_2}]) \tag{5.4}$$

The tracker component is a memory which combines the topmost elements of the stack with the buffer $\vec{b_1}$ taking into account the previous memory $\vec{m_{t-1}}$.

$$\vec{m_t} = Tracker(\vec{m_{t-1}}, \vec{c_1}, \vec{c_2}, \vec{b_1}) \tag{5.5}$$

A fully-connected layer or mlp layer is then used to output the score used for classification and ranking. The architecture of this model is depicted in Fig.5.2 while the hyperparameters used are shown in Table 5.1.

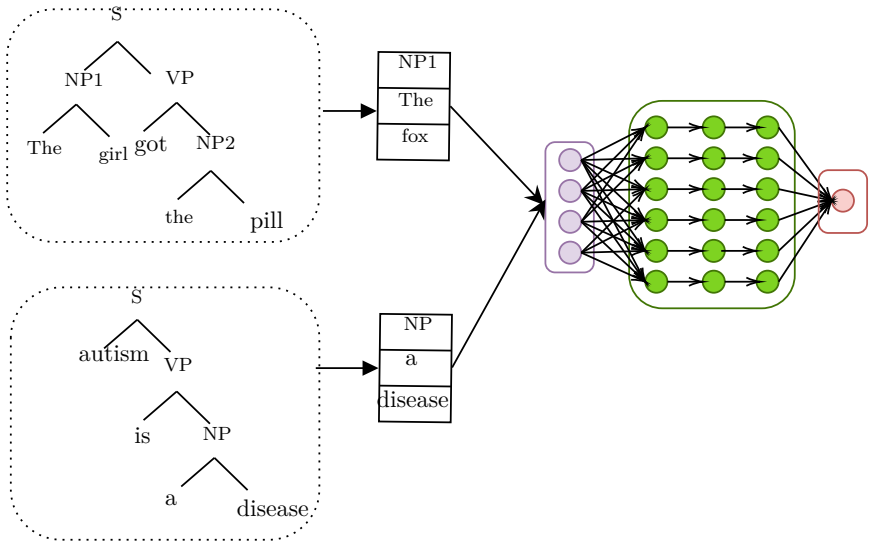


Figure 5.2: Constituency Neural Network is a multilayer network where binary trees are fed as stacks, an embedding layer is used to transform tokens to vectors and a classifier is used to output the similarity between the original question and the question-answer pair.

Table 5.1: CoNN hyperparameters values and descriptions

Parameter	Value	Description
epochs	50	number of iterations user for training
batch_size	128	number of examples per iteration
d_embed	300	dimension of embeddings layer
n_embed	78907	vocabulary size for the embeddings layer
d_proj	100	dimension of projection layer
d_hidden	100	dimension of hidden layer
n_hidden	1	number of hidden layers
lr	2e-3	learning rate
embed_dropout	0.2	the dropout used for the embedding layer
rnn_dropout	0.2	the dropout used for the rnn component
mlp_dropout	0.07	the dropout used for the mlp component
d_mlp	600	dimension of the mlp component
n_mlp_layers	3	number of layers of the mlp component
d_tracker	64	dimension of the tracker component
d_out	2	dimension of the output layer

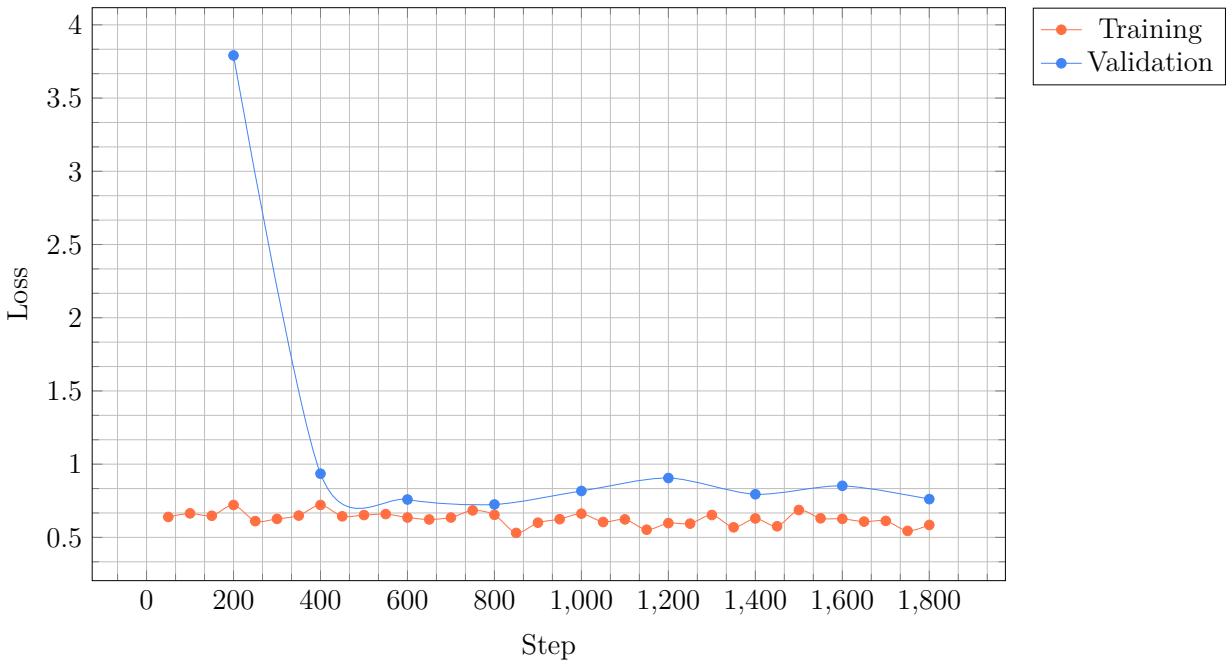


Figure 5.3: Training loss value change with respect to the iteration steps.

5.2.3 Tree Kernels

Kernel trees are functions measuring the structural similarity between trees.

$$\phi_{tk}(q,\rho).\phi_{tk}(q_i,\rho_i) = TK(t(q,\rho),t(q_i,\rho_i)) + TK(t(\rho,q),t(,\rho_i,q_i)) \tag{5.6}$$

where $\mathcal{Q}$ and $\mathcal{A}$ are the set of question-answer pairs, q the original question and $\rho \in \mathcal{Q}, \alpha \in \mathcal{A}$.

We use subset tree kernel (SST) and partial tree kernel (PTK) as described by [155]. The former are subtrees for which all the children of a node belong to the tree or none of them while the latter is the set of partial matching between subtrees.

Fig.5.4 depicts the syntactic trees for the original question and the question extracted from a question-answering pair.

q: هل اضراس العقل تسبب التهاب اللوزتين؟ (Do the molars of the mind cause tonsillitis?)

ρ: ما هو علاج التهاب اللوزتين؟ (What is the treatment of tonsillitis?)

The features obtained from the subset kernel trees or partial kernel trees are then combined with various lexical features and fed into an SVM^{rank} model to get a ranking score for each question-answer pair related to the corresponding original question.

5.3 Experimental Setup

In this section, we detail the experiments that we performed for the different approaches, we describe each dataset, followed by the training setup and implementation details.

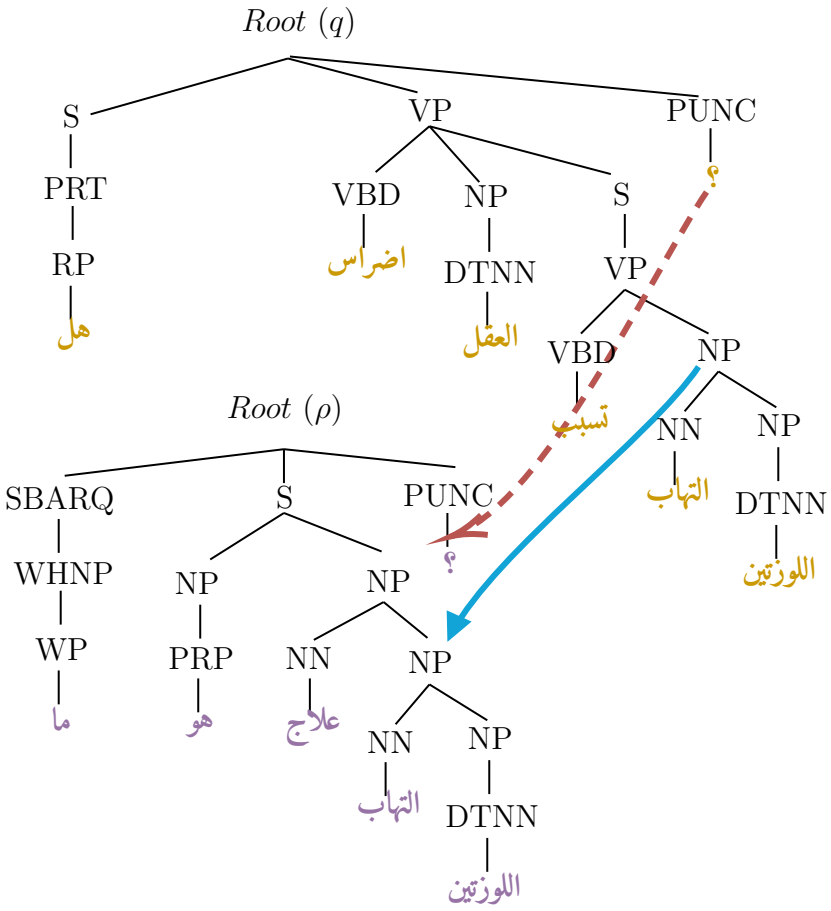


Figure 5.4: Kernel trees for an original question (q) and a pair question (ρ).

Table 5.2: Semeval 2016 CQA-MD dateset instances count by category and split.

Category	Train	Dev	Test 2016
Questions	1,031	250	250
QA Pairs	30,411	7,384	7,369
Direct	917	70	65
Related	17,412	1,446	1,353
Irrelevant	12,082	5,868	5,951

5.3.1 Datasets Description

We use Semeval 2016 CQA-MD [153] which is a community question answering extracted from Arabic biomedical websites such as webteb[2] and altibbi[3] for training, validation, and test. Table 5.2 depicts the number of original questions and question-answer pairs for each split while Table 5.3 depicts an excerpt from the SemEval 2016 CQA-MD dataset.

5.3.2 Training setup

For textual-based embeddings, we have used FastText [156] trained on an Arabic medical domain dataset using 100 dimensions, the dataset composed of 10,899 articles, was collected from biomedical websites containing articles about diseases, drugs, diagnostic and body parts applying the same pre-processing steps as for the task dataset. Table 5.4 dissects the corpus size for each category.

For neural architectures, we trained the model on a Tesla T40 GPU while for shallow architectures, we trained the model on a CPU. The training hyperparameters are depicted in Fig.4.1.

Fig.5.3 shows the evolution of the loss function after 1,800 iteration steps for the training and validation sets confirming the ability of the model to predict the relevance of a question-answer pair regarding the original question. The shape of the curves shows robustness to overfitting.

5.3.3 Implementation details

5.3.3.1 Recursive Neural Network

For the recursive approach, we have pre-processed the text using Stanford CoreNLP through a segmentation, part-of-speech, constituency parsing pipeline and used Pytorch to train the network by feeding the trees after binarization.

[2]https://www.webteb.com
[3]https://www.altibbi.com/

Table 5.3: SemEval 2016 Task 3 Subtask D Training Excerpt

qid	qaid	question	qaquestion	qaanswer	qarel	qaconf
200521	20332	Who gets tooth decay ما سبب تكون حب الشباب	Tooth decay in children السلام عليكم احب اسال عن اسباب ضهور حب الشباب و اهي الكلات او الشياء التي تمنع ضهور حب الشباب وشكرا	Children's teeth are exposed to tooth decay ... ظهور حب الشباب يتعلق بحدوث تغيرات عمرية...	D	1.0
200521	20853	Who gets tooth decay ما سبب تكون حب الشباب	I have a problem of caries in the two front teeth ... حب الشباب....اسبابه وعلاجه وهل يمكن للدرجه الحرارة العاليه ان تؤثر على حب الشباب سلبا	Must follow the scientific specificities by the dentist ... هذه الحبوب حب شباب ظهور حب الشباب يتعلق بحدوث تغيرات...	R	1.0
200521	53749	Who gets tooth decay ما سبب تكون حب الشباب	I have a problem of caries in the two front teeth ... غسول للتخلص من حب الشباب	Must follow the scientific specificities by the dentist ... يوجد انواع كثيرة جيدة بالصيدليات	I	0.69630

5.3.3.2 Tree Kernels

We have used a UIMA pipeline to annotate the dataset and Farasa [122] for parsing. We applied SVM^{rank} to rank the questions related to the original question.

5.4 Results & Discussion

Table 5.5 reports the results obtained using the previously mentioned approaches followed by the results obtained during the contest. The first two rows contain the results for the tree kernel approach, the third row reports the results obtained using a recursive neural network applied to constituency trees. The fourth row is the results for the model combining subset

Table 5.4: Webteb corpus size by category

Category	Count
Body Parts	87
Diseases	653
Drugs	699
Treatments	173
Tests	213
Vaccines	32
Vitamins	24
Recipes	480
Articles	4629
QA	3998
Others	783

tree kernel and constituency neural network. The former leads to better results for the classification while the latter conducts to better results for ranking. The subsequent rows report results obtained during the official contest for the five participating teams. The results report the ranking metrics mAP, AvgRe, and MRR and the classification metrics P, R, F1, and Acc for each model.

Table 5.5: Performance comparison of our tree-based methods with SemEval 2016 results

	Submission	MAP	AvgRec	MRR	P	R	F1	Acc
1	**Subset Tree Kernel**	38.73	**40.68**	**43.96**	**28.78**	**34.63**	**31.43**	**70.93**
2	**Partial Tree Kernel**	33.98	**36.93**	**38.62**	**23.11**	**37.31**	**28.54**	**64.05**
3	**Constituency Neural Network**	**48.50**	**74.03**	**53.05**	**41.96**	**12.05**	**18.73**	**58.95**
4	**CoNN + STK**	**48.50**	**74.03**	**53.05**	**28.78**	**34.63**	**31.43**	**70.93**
5	**SLS-primary**	45.83	**51.01**	**53.66**	**34.45**	**52.33**	**41.55**	**71.67**
6	**ConvKN-primary**	45.50	**50.13**	**52.55**	**28.55**	**64.53**	**39.58**	**62.10**
7	**RDI_team-primary**	43.80	**47.45**	**49.21**	**19.24**	**100.00**	**32.27**	**19.24**
8	**QU-IR-primary**	38.63	**44.10**	**46.27**	**25.50**	**45.13**	**32.59**	**64.07**
9	**UPC_USMBA-primary**	29.09	**30.04**	**34.04**	**20.14**	**51.69**	**28.99**	**51.27**

While applying tree kernels has led to weak performance compared to those obtained using constituency neural network, we notice a neat performance of subset tree kernel, this could be explained by the nature of the dataset containing large sentences and the fact of usage of paraphrasing in cQA. Using a constituency neural network has produced state-of-the-art results for semeval 2016 CQA-MD task confirming our hypothesis. Harnessing the tree structure of the text has lead to a better value for the mean average precision which is the official metric for the task, combining this model with the subset tree kernel, we were able to improve the classification metrics. Using the shift/reduce approach, we were able to feed the model with batched input while harnessing the recursive nature of the neural

model in a linear time.

5.5 Conclusion

Community Question Answering systems are witnessing increasing popularity recently. Much of this interest stems from their capacity to represent a hub for communities to ask questions, seek information, and share knowledge around various topics in a trustworthy and timely manner. Content in these online fora is characterized by a high amount of redundancy, the presence of misspelled words, and usage of paraphrasing which leads to low accuracies. Motivated by these aspects, in this thesis we addressed the problem of cQA for an Arabic medical domain forum by harnessing the graph structure of the text by i) applying two of widely used tree kernels ii) training a recursive neural network on constituency trees iii) reporting the performance on widely used cQA datasets for Arabic biomedical domain.

General Conclusion

Community question answering systems are springing up due to their capacity to address health information needs conveniently. CQA for the medical domain helps users seek information to diagnose diseases by describing symptoms and requesting information about drugs and medical tests. Motivated by these facts, we focused in the present work examining and enhancing community question answering for the Arabic biomedical domain, where we tackled a wide range of issues and challenges such as usage of Latin words for drugs and illnesses, the presence of typos, and the imbalanced nature of datasets. This improvement targeted different components of the cQA pipeline including i) question-answers classification ii) answer selection iii) passage retrieval and iv) question-answers ranking and re-ranking. We proposed a bunch of methods to enable users to find accurate answers to their questions. The experimental results on two widely used datasets have proven effective leading to state-of-the-art results. In this chapter, we present the main conclusions derived from our contributions and perspectives for future work.

Contributions

Our proposed methods are based on machine learning and deep learning applied to tokens and syntactic trees extracted from the original and the question-answer pairs.

Topic Modeling and Pairwise Approaches

We addressed the problem of cQA for the Arabic biomedical domain by harnessing the connection between the original question and the question-answer pairs. We first customize the LSA model to project the original question and the question-answer pairs into a multi-dimensional space of 100 components using the aforementioned algorithm. Besides, we used stemming to alleviate the impact of noise and misspelled words on the performance of the system. We also proofed that harnessing pairwise neural networks improve the ranking performance. Using the LSA model, we were able to obtain high values for the mAP, the accuracy, and the precision thus outperforming the winning system for this task. Using the pointwise approach combined with various neural network models fed with overlapping terms

between the question and the question-answer pairs, we successfully reached comparable performance compared to the pairwise approaches which confirm our initial hypothesis that pairwise methods perform better than pointwise methods.

Combining a stacked CNN and Lexical Features

The Arabic language is suffering from a scarcity of resources which is furthermore widened for specific domain tasks. Our objective is to bridge this gap by training embeddings for Arabic biomedical documents. Our pre-trained vectors learned on an Arabic medical domain corpus contributes significantly to the performance of the model. We fed the obtained embeddings to a stack of convolutional neural networks to detect up to 6-gram and a 3-gram cosine lexical feature leading to state-of-the-art performance. We applied other widely used embeddings for general and specific domains and demonstrated that our embeddings lead to better results. On the other side, using state-of-the-art encoders for related tasks show that our architecture is more convenient for community question answering.

Contextualized Embeddings

Contextualized embeddings have proven effective for a wide spectrum of downstream natural language processing tasks. Motivated by these facts, we addressed in the present work the problem of community question answering applied to Arabic medical domain datasets by leveraging the power of state of the art contextualized embeddings. We compute the similarity between the original question and the question/answer pairs for both ranking and classification. We also demonstrate how contextualized embeddings as sentence representations lead to better results compared to context-free embeddings. This is partially owing to their context-aware traits providing the measured similarity at the semantic level bypassing syntactic errors or paraphrasing usage leading to improved results of up to 2% regarding the mAP, the official metric.

Tree-Based Approaches

we addressed the issue of cQA for an Arabic biomedical domain by harnessing the graph structure of text applying two widely used tree kernels fed into SVM^{rank}. We also train a recursive neural network on constituency trees. We finally report the performance on a cQA dataset for the Arabic biomedical domain. The former approach has proven effective for classification while the latter has led to state-of-the-art results for ranking.

Future Work

We plan to extend this work by pre-training the same algorithms on domain-specific corpora to improve the coverage, thus minimizing the out-of-vocabulary ratio. We are also working on combining a graph-based approach and contextualized embeddings to harness the structure of the text using a knowledge base.

To extend our work, we intend to:

- Use a knowledge base for medical vocabulary to expand both the original question and the corresponding pair of question-answer.
- Performing a dependency parsing of Arabic texts and using the dependencies as an alternative to n-grams for building the convolutional layers.
- Use a graph approach to harness the syntactic structure of sentences.
- Use machine translation evaluation metrics as features to measure the similarity between the original question and the question-answer pair.
- Apply extractive summarization to sentences to reduce the features space dimensionality.

Due to the rich morphology of Arabic words, using a channel of character embeddings as a complement to word embeddings. A similar approach could be a channel of root (triliteral) embeddings, taking profit of one characteristic of the Arabic language: most content words are derived from a triliter verbal root. So, the root of a word can be considered as a code of its meaning.

Publications

International Journals

Yassine El Adlouni, Horacio Rodrìguez, Mohammed Meknassi, Said Ouatik El Alaoui, and Noureddine Ennahnahi. A multiapproach to community question answering. Expert Syst. Appl., 137:432–442, 2019. doi: 10.1016/j.eswa.2019.07.024.

Yassine El Adlouni, Horacio Rodrìguez, Mohammed Meknassi, Said Ouatik El Alaoui, and Noureddine Ennahnahi. Arabic Biomedical Community Question Answering Based on Contextualized Embeddings. Int. J. Intell. Inf. Technol. (Submitted)

Yassine El Adlouni, Horacio Rodrìguez, Mohammed Meknassi, Said Ouatik El Alaoui, and Noureddine Ennahnahi. Arabic Biomedical Community Question Answering using a Stacked Convolutional Neural Network . Journal of King Saud University - Computer and Information Sciences (Minor Revisions)

Yassine El Adlouni, Horacio Rodrìguez, Mohammed Meknassi, Said Ouatik El Alaoui, and Noureddine Ennahnahi. Tree Based Approach to Biomedical Community Question Answering . Natural Language Engineering (Submitted)

Book Chapters

Nabil Alami, **Yassine El Adlouni**, Noureddine En-nahnahi, and Mohammed Meknassi. Using statistical and semantic analysis for arabic text summarization. In Gherabi Noreddine and Janusz Kacprzyk, editors, International Conference on Information Technology and Communication Systems, pages 35–50, Cham, 2018. Springer International Publishing. ISBN 978-3-319-64719-7.

International Conferences & Workshops

Yassine El Adlouni, Imane Lahbari, Horacio Rodrìguez, Mohammed Meknassi, Said Ouatik El Alaoui, and Noureddine Ennahnahi. UPC-USMBA at semeval2017 task 3: Combining multiple approaches for CQA for arabic. In Proceedings of the 11th International

Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017, pages 275–279. Association for Computational Linguistics, 2017. doi: 10.18653/v1/S17-2044.

Yassine El Adlouni, Imane Lahbari, Horacio Rodrìguez, Mohammed Meknassi, Said Ouatik El Alaoui, and Noureddine Ennahnahi. Using domain knowledge and bilingual resources for addressing community question answering for arabic. In 4th IEEE International Colloquium on Information Science and Technology, CiSt 2016, Tangier, Morocco, October 24-26,2016, pages 368–373. IEEE, 2016. doi: 10.1109/CIST.2016.7805073.

Yassine El Adlouni, Imane Lahbari, Horacio Rodrìguez, Mohammed Meknassi, and Said Ouatik El Alaoui. 2016. UPC-USMBA at SemEval-2016 Task 3: UPCUSMBA participation in SemEval 2016 task 3, subtask d: CQA for Arabic. In Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval '16, San Diego, CA.

Bibliography

[1] Bhuwan Dhingra, Kathryn Mazaitis, and William W. Cohen. Quasar: Datasets for question answering by search and reading. *CoRR*, abs/1707.03904, 2017.

[2] P. Rajpurkar, R. Jia, and P. Liang. Know What You Don't Know: Unanswerable Questions for SQuAD. *ArXiv e-prints*, June 2018.

[3] Petr Baudiš and Jan Šedivý. Modeling of the question answering task in the yodaqa system. In Josanne Mothe, Jacques Savoy, Jaap Kamps, Karen Pinel-Sauvagnat, Gareth Jones, Eric San Juan, Linda Capellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 222–228, Cham, 2015. Springer International Publishing.

[4] Alexander H. Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. *CoRR*, abs/1606.03126, 2016.

[5] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *CoRR*, abs/1705.03551, 2017.

[6] Yi Yang, Wen-tau Yih, and Christopher Meek. Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 2013–2018, 2015.

[7] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-Driven Learning for Open Question Answering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013.

[8] A. Ben Abacha and D. Demner-Fushman. A Question-Entailment Approach to Question Answering. *arXiv e-prints*, January 2019.

[9] Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. Semeval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 525–545, 2016.

[10] Asma Ben Abacha and Dina Demner-Fushman. Recognizing question entailment for medical question answering. In *AMIA 2016, American Medical Informatics Association Annual Symposium, Chicago, IL, USA, November 12-16, 2016*, 2016.

[11] D. Ferrucci, E. Nyberg, J. Allan, K. Barker, E. Brown, J. Chu-Carroll, A. Ciccolo, P. Duboue, J. Fan, D. Gondek, E. Hovy, B. Katz, A. Lally, M. McCord, P. Morarescu, W. Murdock, B. Porter, J. Prager, T. Strzalkowski, W. Welty, , and W. Zadrozny. Towards the open advancement of question answer systems. In *IBM Technical Report RC24789, Yorktown Heights, NY.* The Association for Computer Linguistics, 2009.

[12] J. Weston, Bordes, A., Chopra, and T. S. & Mikolov. Towards ai-complete question answering: a set of prerequisite toy tasks. In *http://arxiv.org/abs/1502.05698*, 2015.

[13] Xin Li and Dan Roth. Learning question classifiers. In *19th International Conference on Computational Linguistics, COLING 2002, Howard International House and Academia Sinica, Taipei, Taiwan, August 24 - September 1, 2002*, 2002.

[14] John W Ely, Jerome A Osheroff, Paul N Gorman, Mark H Ebell, M Lee Chambliss, Eric A Pifer, and P Zoe Stavri. A taxonomy of generic clinical questions: classification study. *BMJ*, 321(7258):429–432, 2000.

[15] Sanda M. Harabagiu, V. Finley Lacatusu, and Andrew Hickl. Answering complex questions with random walk models. In *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 220–227, 2006.

[16] Halil Kilicoglu, Marcelo Fiszman, and Dina Demner-Fushman. Interpreting consumer health questions: The role of anaphora and ellipsis. In *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing, BioNLP@ACL 2013, Sofia, Bulgaria, August 8, 2013*, pages 54–62, 2013.

[17] Kirk Roberts, Kate Masterton, Marcelo Fiszman, Halil Kilicoglu, and Dina Demner-Fushman. Annotating question decomposition on complex medical questions. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 2598–2602, 2014.

[18] Tyler B Forbush, Adi Gundlapalli, Miland Palmer, Shuying Shen, Brett South, Guy Divita, Marjorie Carter, Andrew Redd, Jorie Butler, and Matthew Samore. "sitting on pins and needles": Characterization of symptom descriptions in clinical notes". *AMIA Summits on Translational Science proceedings AMIA Summit on Translational Science*, 2013:67–71, 03 2013.

[19] Sven Hartrumpf. Semantic decomposition for question answering. In *ECAI 2008 - 18th European Conference on Artificial Intelligence, Patras, Greece, July 21-25, 2008, Proceedings*, pages 313–317, 2008.

[20] Lou Ann Scarton, Liqin Wang, Halil Kilicoglu, Margaret Jahries, and Guilherme Del Fiol. Expanding vocabularies for complementary and alternative medicine therapies. *I. J. Medical Informatics*, 121:64–74, 2019.

[21] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio. Learning structured embeddings of knowledge bases. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 7-11, 2011*, 2011.

[22] Antoine Bordes, Xavier Glorot, Jason Weston, and Yoshua Bengio. Joint learning of words and meaning representations for open-text semantic parsing. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, Spain, April 21-23, 2012*, pages 127–135, 2012.

[23] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. Irreflexive and hierarchical relations as translations. *CoRR*, abs/1304.7158, 2013.

[24] Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340, 2015.

[25] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *CoRR*, abs/1511.02301, 2015.

[26] Bhuwan Dhingra, Hanxiao Liu, William W. Cohen, and Ruslan Salakhutdinov. Gated-attention readers for text comprehension. *CoRR*, abs/1606.01549, 2016.

[27] Yue Chen, Ana Busic, and Sean P. Meyn. Ergodic theory for controlled markov chains with stationary inputs. *CoRR*, abs/1604.04013, 2016.

[28] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. *CoRR*, abs/1608.07905, 2016.

[29] Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. Text understanding with the attention sum reader network. *CoRR*, abs/1603.01547, 2016.

[30] Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. Attention-over-attention neural networks for reading comprehension. *CoRR*, abs/1607.04423, 2016.

[31] Alessandro Sordoni, Phillip Bachman, and Yoshua Bengio. Iterative alternating neural attention for machine reading. *CoRR*, abs/1606.02245, 2016.

[32] Adam Trischler, Zheng Ye, Xingdi Yuan, and Kaheer Suleman. Natural language comprehension with the epireader. *CoRR*, abs/1606.02270, 2016.

[33] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *CoRR*, abs/1611.01603, 2016.

[34] O. Vinyals, M. Fortunato, and N. Jaitly. Pointer Networks. *ArXiv e-prints*, June 2015.

[35] Matthew S. Simpson and Dina Demner-Fushman. Biomedical text mining: A survey of recent progress. In *Mining Text Data*, 2012.

[36] Hong Yu and Carl Sable. Being erlang shen : Identifying answerable questions. In /, 2005.

[37] Pierre Zweigenbaum. Knowledge and reasoning for medical question-answering. In *Proceedings of the 2009 Workshop on Knowledge and Reasoning for Answering Questions*, KRAQ '09, pages 1–2, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[38] Dina Demner-Fushman and Jimmy J. Lin. Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 33(1):63–103, 2007.

[39] Dina Demner-Fushman, Barbara Few, Susan E. Hauser, and George R. Thoma. Research paper: Automatically identifying health outcome information in MEDLINE records. *JAMIA*, 13(1):52–60, 2006.

[40] YongGang Cao, Feifan Liu, Pippa Simpson, Lamont Antieau, Andrew Bennett, James J. Cimino, John Ely, and Hong Yu. Askhermes: An online question answering system for complex clinical questions. *Journal of Biomedical Informatics*, 44(2):277 – 288, 2011.

[41] Sofia J. Athenikos and Hyoil Han. Biomedical question answering: A survey. *Computer Methods and Programs in Biomedicine*, 99(1):1–24, 2010.

[42] Kai Zhang, Wei Wu, Haocheng Wu, Zhoujun Li, and Ming Zhou. Question retrieval with high quality answers in community question answering. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 371–380, New York, NY, USA, 2014. ACM.

[43] Yassine El Adlouni, Imane Lahbari, Horacio Rodríguez, Mohammed Meknassi, Said Ouatik El Alaoui, and Noureddine Ennahnahi. UPC-USMBA at semeval-2017 task 3: Combining multiple approaches for CQA for arabic. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 275–279, 2017.

[44] Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft. Retrieval models for question and answer archives. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008*, pages 475–482, 2008.

[45] Guangyou Zhou, Yang Liu, Fang Liu, Daojian Zeng, and Jun Zhao. Improving question retrieval in community question answering using world knowledge. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 2239–2245, 2013.

[46] Li Cai, Guangyou Zhou, Kang Liu, and Jun Zhao. Learning the latent topics for question retrieval in community QA. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011, Chiang Mai, Thailand, November 8-13, 2011*, pages 273–281, 2011.

[47] Zongcheng Ji, Fei Xu, Bin Wang, and Ben He. Question-answer topic model for question retrieval in community question answering. In *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*, pages 2471–2474, 2012.

[48] Cícero Nogueira dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. Learning hybrid representations to retrieve semantically equivalent questions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 2: Short Papers*, pages 694–699, 2015.

[49] Rada Mihalcea and Dragomir Radev. *Graph-based Natural Language Processing and Information Retrieval*. Cambridge University Press, 2011.

[50] Hongyuan Zha and Xiang Ji. Correlating multilingual documents via bipartite graph modeling. In /, pages 443–444, 01 2002.

[51] Hongyuan Zha and Xiang Ji. Correlating multilingual documents via bipartite graph modeling. In *SIGIR 2002: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland*, pages 443–444, 2002.

[52] S. S. Sonawane and P. A. Kulkarni. Article: Graph based representation and analysis of text document: A survey of techniques. *International Journal of Computer Applications*, 96(19):1–8, June 2014. Full text available.

[53] Andrei Alexandrescu and Katrin Kirchhoff. Data-driven graph construction for semi-supervised graph-based learning in NLP. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, pages 204–211, 2007.

[54] Xiaojin Zhu. *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University, 2005.

[55] Jari Björne, Juho Heimonen, Filip Ginter, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. Extracting complex biological events with rich graph-based feature sets. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, BioNLP@HLT-NAACL 2009 - Shared Task, Boulder, Colorado, USA, June 5, 2009*, pages 10–18, 2009.

[56] Amit Mishra and Sanjay Kumar Jain. A survey on question answering systems with classification. *J. King Saud Univ. Comput. Inf. Sci.*, 28(3):345–361, July 2016.

[57] Junlong Zhang and Yu Gui Luo. Degree centrality, betweenness centrality, and closeness centrality in social network. In /, 2017.

[58] Jon M. Kleinberg. Hubs, authorities, and communities. *ACM Comput. Surv.*, 31(4es):5, 1999.

[59] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[60] Taher H. Haveliwala. Topic-sensitive pagerank. In *Proceedings of the Eleventh International World Wide Web Conference, WWW 2002, May 7-11, 2002, Honolulu, Hawaii, USA*, pages 517–526, 2002.

[61] Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*, pages 33–41, 2009.

[62] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *CoRR*, abs/1109.2128, 2011.

[63] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 404–411, 2004.

[64] D.C. Lay. *Linear Algebra and Its Applications*. Addison-Wesley, 2012.

[65] Danai Koutra, Aaditya Ramdas, Ankur Parikh, and Jing Xiang. Algorithms for graph similarity and subgraph matching, 2011.

[66] Yuanyuan Tian, Richard C. McEachin, Carlos Santos, David J. States, and Jignesh M. Patel. SAGA: a subgraph matching tool for biological graphs. *Bioinformatics*, 23(2):232–239, 2007.

[67] Haiyan Hu, Xifeng Yan, Yu Huang, Jiawei Han, and Xianghong Jasmine Zhou. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21(suppl_1):i213–i221, 06 2005.

[68] Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, August 12-15, 2007*, pages 687–696, 2007.

[69] Mohamed Touzani. *Alignement des ontologies OWL-Lite*. PhD thesis, Montreal University, 2005.

[70] Danai Koutra, Tai-You Ke, U. Kang, Duen Horng Chau, Hsing-Kuo Kenneth Pao, and Christos Faloutsos. Unifying guilt-by-association approaches: Theorems and fast algorithms. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011, Proceedings, Part II*, pages 245–260, 2011.

[71] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 855–864, 2016.

[72] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: large-scale information network embedding. *CoRR*, abs/1503.03578, 2015.

[73] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, February 26 - March 1, 2002*, pages 117–128, 2002.

[74] Ronald Read and Derek G. Corneil. The graph isomorphism disease. *Journal of Graph Theory*, 1:339 – 363, 10 2006.

[75] Yunsheng Bai, Hao Ding, Song Bian, Ting Chen, Yizhou Sun, and Wei Wang. Graph edit distance computation via graph neural networks. *CoRR*, abs/1808.05689, 2018.

[76] Horst Bunke and Kim Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recogn. Lett.*, 19(3-4):255–259, March 1998.

[77] Nataša Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 01 2007.

[78] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and Applications*, 13:113–129, 2008.

[79] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80, 2009.

[80] David Duvenaud, Dougal Maclaurin, Jorge Aguilera-Iparraguirre, Rafael Gómez-Bombarelli, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P. Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2224–2232, 2015.

[81] Alessandro Lusci, Gianluca Pollastri, and Pierre Baldi. Deep architectures and deep learning in chemoinformatics: The prediction of aqueous solubility for drug-like molecules. *Journal of Chemical Information and Modeling*, 53(7):1563–1575, 2013.

[82] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: online learning of social representations. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, pages 701–710, 2014.

[83] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard S. Zemel. Gated graph sequence neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[84] Mathias Niepert and Alberto García-Durán. Towards a spectrum of graph convolutional networks. In *2018 IEEE Data Science Workshop, DSW 2018, Lausanne, Switzerland, June 4-6, 2018*, pages 244–248, 2018.

[85] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. Learning convolutional neural networks for graphs. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2014–2023, 2016.

[86] Dario Garcia-Gasulla, Armand Vilalta, Ferran Parés, Jonathan Moreno, Eduard Ayguadé, Jesús Labarta, Ulises Cortés, and Toyotaro Suzumura. Building graph representations of deep vector embeddings. *CoRR*, abs/1707.07465, 2017.

[87] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 1112–1119, 2014.

[88] Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 926–934, 2013.

[89] Rodolphe Jenatton, Nicolas Le Roux, Antoine Bordes, and Guillaume Obozinski. A latent factor model for highly multi-relational data. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.*, pages 3176–3184, 2012.

[90] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[91] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *CoRR*, abs/1410.3916, 2014.

[92] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014.

[93] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051, 2017.

[94] Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. Deep learning for answer sentence selection. *CoRR*, abs/1412.1632, 2014.

[95] Antoine Bordes, Sumit Chopra, and Jason Weston. Question answering with subgraph embeddings. *CoRR*, abs/1406.3676, 2014.

[96] Antoine Bordes, Jason Weston, and Nicolas Usunier. Open question answering with weakly supervised embedding models. *CoRR*, abs/1404.4326, 2014.

[97] Yuanfang Wang Junjie Ke and Fei Xia. Question answering system with bi-directional attention flow, 2017.

[98] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *CoRR*, abs/1611.01604, 2016.

[99] Xipeng Qiu and Xuanjing Huang. Convolutional neural tensor network architecture for community-based question answering. In *IJCAI*, 2015.

[100] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247. Association for Computational Linguistics, 2014.

[101] Quoc Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196, 2014.

[102] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[103] B Chiu, G Crichton, A Korhonen, and S Pyysalo. How to train good word embeddings for biomedical nlp. *Proceedings of the 15th Workshop on Biomedical NLP*, 2016.

[104] Tomas Mikolov, Quoc V Le, , and Ilya Sutskever. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168, 2013*, 2013.

[105] Mikel Artetxe, Gorka Labaka, , and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual nvariance. *Proceedings of EMNLP, 2016*, 2016.

[106] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[107] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[108] Jeremy Howard and Sebastian Ruder. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146, 2018.

[109] Alec Radford. Improving language understanding by generative pre-training. In /, 2018.

[110] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[111] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. Gaithersburg, MD: NIST, January 1995.

[112] Preslav Nakov, Doris Hoogeveen, Lluís Màrquez, Alessandro Moschitti, Hamdy Mubarak, Timothy Baldwin, and Karin Verspoor. SemEval-2017 task 3: Community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 27–48, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[113] Salvatore Romeo, Giovanni Da San Martino, Alberto Barrón-Cedeño, Alessandro Moschitti, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Mitra Mohtarami, and James Glass. Neural attention for learning to rank questions in community question answering. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1734–1745, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[114] Marc Franco-Salvador, Sudipta Kar, Thamar Solorio, and Paolo Rosso. UH-PRHLT at SemEval-2016 task 3: Combining lexical and semantic-based features for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 814–821, San Diego, California, June 2016. Association for Computational Linguistics.

[115] Shafiq R. Joty, Lluís Màrquez, and Preslav Nakov. Joint multitask learning for community question answering using task-specific embeddings. *CoRR*, abs/1809.08928, 2018.

[116] Francisco Guzmán, Lluís Màrquez, and Preslav Nakov. Machine translation evaluation meets community question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 2: Short Papers*, 2016.

[117] Yonatan Belinkov, Alberto Barrón-Cedeño, and Hamdy Mubarak. Answer selection in arabic community question answering: A feature-rich approach. In *ANLP@ACL*, 2015.

[118] Alessandro Moschitti. Efficient convolution kernels for dependency and constituent syntactic trees. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, pages 318–329, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[119] Alberto Barrón-Cedeño, Giovanni Da San Martino, Salvatore Romeo, and Alessandro Moschitti. Selecting sentences versus selecting tree constituents for automatic question ranking. In *COLING*, 2016.

[120] Salvatore Romeo, Giovanni Da San Martino, Alberto Barrón-Cedeño, Alessandro Moschitti, Yonatan Belinkov, Wei-Ning Hsu, Yu Zhang, Mitra Mohtarami, and James R. Glass. Neural attention for learning to rank questions in community question answering. In *COLING*, 2016.

[121] Salvatore Romeo, Giovanni Da San Martino, Yonatan Belinkov, Alberto Barrón-Cedeño, Mohamed Eldesouki, Kareem Darwish, Hamdy Mubarak, James Glass, and Alessandro Moschitti. Language processing and learning models for community question answering in arabic. *Information Processing & Management*, 56(2):274 – 290, 2019. Advance Arabic Natural Language Processing (ANLP) and its Applications.

[122] Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California, June 2016. Association for Computational Linguistics.

[123] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in*

Neural Information Processing Systems 26, pages 3111–3119. Curran Associates, Inc., 2013.

[124] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.

[125] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *CoRR*, abs/1607.04606, 2016.

[126] Guangyou Zhou, Yang Liu, Fang Liu, Daojian Zeng, and Jun Zhao. Improving question retrieval in community question answering using world knowledge. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2239–2245. AAAI Press, 2013.

[127] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 697–706, New York, NY, USA, 2007. ACM.

[128] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, 193:217–250, December 2012.

[129] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.

[130] Min Yang, Wenting Tu, Qiang Qu, Wei Zhou, Qiao Liu, and Jia Zhu. Advanced community question answering by leveraging external knowledge and multi-task learning. *Knowledge-Based Systems*, 171:106–119, May 2019.

[131] Denis Lukovnikov, Asja Fischer, Jens Lehmann, and Sören Auer. Neural network-based question answering over knowledge graphs on word and character level. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1211–1220, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[132] Yassine El Adlouni, Horacio Rodríguez, Mohammed Meknassi, Said Ouatik El Alaoui, and Noureddine En-nahnahi. A multi-approach to community question answering. *Expert Systems with Applications*, 137:432–442, December 2019.

[133] Xiao Yang, Madian Khabsa, Miaosen Wang, Wei Wang, Madian Khabsa, Ahmed Awadallah, Daniel Kifer, and C. Lee Giles. Recurrent convolutional neural network

for answer selection in community question answering. *Neurocomputing*, 274:8–18, January 2018.

[134] Lei Sang, Min Xu, ShengSheng Qian, and Xindong Wu. Multi-modal multi-view Bayesian semantic embedding for community question answering. *Neurocomputing*, 334:44–58, March 2019.

[135] Salvatore Romeo, Giovanni Da San Martino, Yonatan Belinkov, Alberto Barrón-Cedeño, Mohamed Eldesouki, Kareem Darwish, Hamdy Mubarak, James Glass, and Alessandro Moschitti. Language processing and learning models for community question answering in Arabic. *Information Processing & Management*, 56(2):274–290, March 2019.

[136] Fang Wang, Wei Wu, Zhoujun Li, and Ming Zhou. Named entity disambiguation for questions in community question answering. *Knowledge-Based Systems*, 126:68–77, June 2017.

[137] Guangyou Zhou, Yin Zhou, Tingting He, and Wensheng Wu. Learning semantic representation with neural networks for community question answering retrieval. *Knowledge-Based Systems*, 93:75–83, February 2016.

[138] Zhifeng Hao, Biao Wu, Wen Wen, and Ruichu Cai. A subgraph-representation-based method for answering complex questions over knowledge bases. *Neural Networks*, 119:57–65, November 2019.

[139] Wei Wu, Xu Sun, and Houfeng Wang. Question condensing networks for answer selection in community question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1746–1755, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[140] Xiao Yang, Madian Khabsa, Miaosen Wang, Wei Wang, Madian Khabsa, Ahmed Awadallah, Daniel Kifer, and C. Lee Giles. Adversarial Training for Community Question Answer Selection Based on Multi-scale Matching. *arXiv:1804.08058 [cs]*, April 2018.

[141] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[142] Kazem Taghva, Rania Elkhoury, and Jeffrey S. Coombs. Arabic stemming without A root dictionary. In *International Symposium on Information Technology: Coding and Computing (ITCC 2005), Volume 1, 4-6 April 2005, Las Vegas, Nevada, USA*, pages 152–157. IEEE Computer Society, 2005.

[143] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[144] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[145] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *CoRR*, abs/1607.01759, 2016.

[146] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[147] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. *CoRR*, abs/1702.03814, 2017.

[148] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, and Hui Jiang. Enhancing and combining sequential and tree LSTM for natural language inference. *CoRR*, abs/1609.06038, 2016.

[149] Jiwoon Jeon, W. Bruce Croft, Joon Ho Lee, and Soyeon Park. A framework to predict the quality of answers with non-textual features. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 228–235, 2006.

[150] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[151] Nada AlMarwani and Mona T. Diab. Gw_qa at semeval-2017 task 3: Question answer re-ranking on arabic fora. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 344–348, 2017.

[152] Marwan Torki, Maram Hasanain, and Tamer Elsayed. QU-BIGIR at semeval 2017 task 3: Using similarity features for arabic community question answering forums. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 360–364, 2017.

[153] Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 525–545, San Diego, California, June 2016. Association for Computational Linguistics.

[154] Preslav Nakov, Lluís Màrquez i Villodre, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, James R. Glass, and Bilal Randeree. Semeval-2016 task 3: Community question answering. In *SemEval@NAACL-HLT*, 2016.

[155] Alessandro Moschitti. Making tree kernels practical for natural language learning. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy, April 2006. Association for Computational Linguistics.

[156] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification, 2016.

APPENDIX A

Résumé Détaillé

A.1 Introduction

Les systèmes de question-réponse communautaires (cQA) connaissent récemment un intérêt croissant avec l'avènement des forums tels que Yahoo !Answers, Quora, Reddit et StackExchange. Une grande partie de cet intérêt découle de leur capacité à devenir une plateforme permettant à la communauté de poser des questions, rechercher des informations et partager des connaissances sur divers sujets de manière fiable et opportune. Outre leur objectif évident d'obtenir une réponse à une question, les systèmes cQA devraient être en mesure d'aider les utilisateurs à vérifier l'existence de questions similaires avant de soumettre sa requête, ce qui pourrait permettre un gain de temps considérable, éviter d'attendre la réponse des experts, minimiser les questions sans réponse et celles en double contribuant ainsi à l'efficacité de la plateforme. Le contenu de ces forums est caractérisé par une grande redondance, la présence des mots mal orthographiés et l'utilisation de paraphrases qui conduisent à une faible précision. En outre, leurs performances sont fortement influencées par la nature déséquilibrée de l'ensemble de données, car peu d'instances sont pertinentes pour la requête de l'utilisateur et l'aspect long des questions et des réponses où le contenu est généralement complété par des salutations et d'autres constructions qui ne sont pas en relation avec la question. La pénurie des ressources pour certaines langues a également un impact sur les systèmes cQA car, une analyse sémantique du texte nécessite généralement des ressources telles que les dictionnaires, les thésaurus, et les bases de connaissances.

Les systèmes question-réponse communautaires cQA peuvent être définis comme, étant donné une question dénotée la question originale et un fil de discussion sous forme d'un ensemble de paires de question-réponse, classer les paires en se basant sur leur similarité avec la question originale. Les tâches cQA sont généralement subdivisées en deux sous-tâches:

- Classification : classer chaque fil de discussion comme étant non pertinent, pertinent ou direct par rapport à la question originale
- Classement : classer les fils de discussion en fonction de leur similarité avec la question originale

Plus formellement, une tâche cQA pourrait être définie comme un triple $(\mathcal{Q}, \mathcal{D}, \mathcal{Y})$ où $\mathcal{Q}$ est un ensemble de questions originales, $\mathcal{D}$ un ensemble de paires question-réponse et $\mathcal{Y}$ un ensemble d'étiquettes.

$$\mathcal{Q} = \{q_i\}_{i=1}^{n} \tag{A.1}$$

$$\mathcal{D}_i = \{(Q_{ij}, A_{ij})\}_{j=1}^{m_i} \tag{A.2}$$

$$\mathcal{Y}_i \in \{Pertinent, Nonpertinent, Direct \tag{A.3}$$

Où:

- n est le nombre des questions originales.
- m_i est le nombre de paires liées à la requête q_i

A.1.1 Motivations

Les systèmes de question-réponse communautaires sont de plus en plus utilisés de nos jours pour aider les utilisateurs à trouver plus facilement des informations. Certaines sont destinées à des domaines ouverts tels que Quora, Yahoo !Answers et Reddit, tandis que d'autres sont destinées à des domaines spécifiques tels que StackExchange qui est un réseau multilingue de forums pour divers domaines notamment la technologie, la science et les affaires.

En ce qui concerne les systèmes de question-réponse (QA), les utilisateurs posent des questions courtes alors que les réponses sont généralement longues, sauf dans de rares cas où le répondant redirige vers des questions similaires où la question est une interrogation totale avec une réponse oui/non ou factoïde. Cela rend laborieuse la tâche de similarité lexicale, syntaxique et sémantique.

Les questions contiennent fréquemment des mots mal orthographiés introduits par les utilisateurs lors de la saisie de leurs questions ce qui engendre un grand nombre de termes hors vocabulaire qui entravent les phases de l'entraînement et de l'apprentissage du modèle.

Dans les forums de domaines spécifiques, les questions et réponses contiennent généralement des termes scientifiques, des abréviations nécessitant l'utilisation de bases de connaissances et des nomenclatures toponymiques afin de produire des modèles plus précis.

Ce travail est motivé par les faits précédents, l'approche proposée vise à construire un modèle capable de déterminer la relation entre une paire de question-réponse et de reclasser un ensemble de questions en fonction de leur similitude par rapport à une question originale.

A.1.2 Défis

La performance des cQA arabes est fortement influencée par la nature déséquilibrée des jeux de données cibles, la longueur des questions et réponses, les sujets techniques abordés dans ces forums, la rareté des ressources linguistiques caractérisant l'arabe et l'usage des mots latins translittérés et bruyants, ces problèmes produisent de faibles valeurs de mAP et de précision par rapport aux cQA en anglais. Contrairement à d'autres plateformes cQA impliquant des utilisateurs du même milieu, les plateformes cQA biomédicaux rassemblent deux types d'utilisateurs, les utilisateurs avertis en matière de santé, tels que les médecins et les utilisateurs à la recherche d'informations. Les premiers utilisent un langage plus formel, l'arabe standard moderne (MSA) par exemple, tandis que les derniers utilisent le dialecte arabe en fonction de leur emplacement. Cet écart entre le vocabulaire de la question et le vocabulaire de la réponse se traduit par un grand nombre de mots hors vocabulaire (OOV) entravant la classification ainsi que le classement et les capacités de reclassement de la plateforme. Cet écart est encore élargi par un grand nombre de dialectes pour l'arabe comptant plus de 30 dialectes. Les utilisateurs des cQA utilisent généralement de manière flagrante les signes de ponctuation lorsqu'ils se précipitent pour soumettre leur requête, ce qui influence toutes les phases ultérieures du pipeline du prétraitement, telles que la segmentation en phrases, l'analyse morphologique, l'analyse syntaxique des constituants et l'analyse syntaxique des dépendances.

A.1.2.1 Présence de fautes de frappe

Les mots mal orthographiés se produisent souvent si l'utilisateur tape particulièrement vite ou en raison d'un manque de connaissances. Certaines fautes de frappe courantes sont : des lettres en double, des lettres manquantes et des mots concaténés. Cela introduit un nombre élevé de termes OOV et par conséquent un impact sur les performances de la plateforme.

A.1.2.2 Lettres non normalisées

Les lettres sous-jacentes composant deux mots en Arabe peuvent différents même si les mots semblent visuellement identiques. Ce problème provient du fait qu'une lettre en Arable a trois formes différentes selon sa position dans le mot ce qui influence les étapes majeures de l'analyse lexicale, syntaxique et sémantique.

A.1.2.3 Utilisation des mots latins ou translittérés

Pour des raisons liées au manque de termes techniques correspondants en arabe, les utilisateurs, en particulier les plus avertis, ont tendance à utiliser des termes en latin ou translittérés, la variance étant élevée pour ces derniers.

A.1.2.4 Jeux de données déséquilibrés

La plupart des jeux de données de référence pour les QA et cQA sont déséquilibrés, ce qui découle du fait que peu de questions sont pertinentes par rapport à la question originale. Ce problème peut d'avantage entraver les performances des modèles à base de l'apprentissage automatique (ML) et peut être résolu en utilisant des technique comme l'augmentation, l'échantillonnage ou en utilisant des algorithmes plus adaptés.

A.1.2.5 Ressources rares

La langue Arabe, comme d'autres langues, souffre de pénurie des ressources, à la fois linguistiques comme les thésaurus, les ontologies, les dictionnaires et les corpus annotés et techniques comme les outils de traitement.

A.1.3 Brève introduction de l'approche

Les systèmes cQA peuvent être abordés sous différents angles en tant que classification ou classement. Nous avons appliqué diverses approches afin de produire un modèle précis capable de déterminer la relation entre la question originale et les paires de question-réponse. Une analyse approfondie doit prendre en considération la similarité lexicale, syntaxique et sémantique.

Une approche à base de modélisation des thèmes ou sujets (Topic Modeling) est une approche à la fois simple mais puissante capable de mesurer la similarité sémantique entre une question originale et un ensemble de paires de question-réponse, le score de similarité est utilisé pour le classement ainsi que pour la classification en moyennant un seuil pour. Nous avons utilisé TfIdf comme caractéristique et diverses techniques de factorisation matricielle telles que l'allocation latente de Dirichlet dénotée LDA, l'analyse sémantique latente dénotée LSA et la factorisation matricielle non négative dénotée NMF.

Les approches d'apprentissage profond se sont avérées assez efficaces pour des tâches telles que la reconnaissance des entités nommées (NER), la reconnaissance de l'implication textuelle (RTE) et la traduction automatique (MT). Nous avons appliqué diverses architectures neuronales qui ont conduit à des résultats médiocres. Suivant l'intuition que les n-grams sont puissants pour les langages hautement flexionnels, nous avons combiné un réseau de neurones convolutif empilé avec des caractéristiques lexicales pour améliorer les

performances de notre système. Nous avons également utilisé des plongements de mots entraînés sur un corpus Arabe du domaine biomédical pour améliorer d'avantage nos résultats.

Nous appliquons en outre une approche multilingue de l'Arabe vers l'Anglais pour vérifier les résultats. Nous utilisons le même modèle « Réseau de neurones convolutif empilé » avec des plongements pré-entraînés produits par pyysalo, ces plongements ont été entraînés sur PMC et PubMed avec des n-grams allant jusqu'à 7-gram. Les résultats obtenus sont proches de ceux obtenus avec nos propres plongements. Nous améliorons d'avantage ces plongements en prenant en considération le contexte via l'utilisation des plongements contextualisés.

L'utilisation des reformulations lors de la soumission des requêtes caractérise les systèmes cQA, nous appliquons une approche basée sur les arbres pour tirer parti de la structure hiérarchique du texte. Deux approches sont utilisées, celle à base des noyaux des arbres et celle à base des réseaux de neurones récursifs.

Pour résumer, les principales contributions sont les suivantes :

A.1.3.1 Modélisation de sujet

Le modèle basé sur LSA a pu surpasser le système état de l'art pour le jeu de données cible suivant une approche non-supervisée, ce système ne nécessite pas de ressources externes et est donc plus pratique et plus efficace pour cette tâche.

A.1.3.2 Apprentissage du classement par paires

L'utilisation de l'approche par paires pour les cQA permet d'améliorer les résultats des modèles d'apprentissage profond. Cette hypothèse a été confirmée en utilisant un ensemble de réseaux de neurones profonds de référence pour d'autres tâches de traitement automatique de langues.

A.1.3.3 Plongements biomédicaux arabes

Nous utilisons après prétraitement, un ensemble de documents biomédicaux en Arabe pour entraîner des plongements distributionnels denses pour des termes de vocabulaire qui sont utilisés par la suite pour alimenter diverses architectures de réseaux de neurones. Cela améliore largement les performances de notre système.

A.1.3.4 Réseau convolutif empilé

Bien que certaines architectures à base des réseaux de neurones produisent des résultats satisfaisants pour les jeux de données de référence pour d'autres langues telles que l'Anglais, l'application des mêmes modèles sur les systèmes cQA engendre des précisions faibles, cela peut être expliqué par la nature bruyante de notre jeu de données. Nous proposons une

architecture neuronale alimentée par les plongements susmentionnés combinés à des caractéristiques lexicales pour calculer la similarité entre la question originale et les paires de question-réponse.

A.1.3.5 Plongements contextualisés

Les plongements contextualisés se sont avérés plus efficaces pour un large éventail de tâches de traitement automatique de langues naturelles telles que le QA, la détection de reformulation ou paraphrase et la reconnaissance de l'implication textuelle RTE. L'ajustement de ces plongements contextualisés pour l'Arabe améliore nettement les performances de notre système.

A.1.3.6 Approche à base des arbres

L'utilisation de la structure aplatie du texte néglige la structure hiérarchique du texte. Nous utilisons deux stratégies afin de prendre en considération cette structure dans le but d'améliorer les performances des systèmes cQA. La première se base sur les noyaux d'arbres qui sont utilisés en entrée des algorithmes d'apprentissage automatique alors que la deuxième utilise les réseaux de neurones récursifs alimentés avec des arbres binaires extraits à l'aide d'une analyse syntaxique.

A.2 Méthodes

A.2.1 Modélisation des sujets

L'application du modèle à base du LSA conduit à des résultats meilleurs que ceux obtenus lors de la compétition SemEval, l'utilisation du tokenizer Stanford CoreNLP améliore d'avantage les résultats. L'utilisation d'autres approches de modélisation de sujets telles que NMF et LDA, par exemple, conduit à des précisions comparables mais des faibles valeurs pour le mAP. L'utilisation de SVM^{rank}, la méthode omniprésente de classement, a produit une haute précision mais un faible mAP, cela pourrait s'expliquer par la nature déséquilibrée de l'ensemble de données et la longueur des questions originales et des paires de question-réponse. Le modèle BiGRU est généralement utilisé pour encoder les phrases, ces dernières sont ensuite transmises à une douche dense pour la classification, une comparaison entre les résultats obtenus en utilisant l'intersection des triplets en entrée sont considérablement meilleurs que ceux obtenus en utilisant leurs concaténations, cela pourrait s'expliquer par le rapport entre la taille du jeu de données et la taille du vecteur des caractéristiques. Les méthodes d'apprentissage profond nécessitent un énorme corpus d'entraînement afin d'apprendre les poids des modèles sans surajustement. La minimisation des entités à l'aide de l'intersection réduit également le temps d'apprentissage. La comparaison des résultats

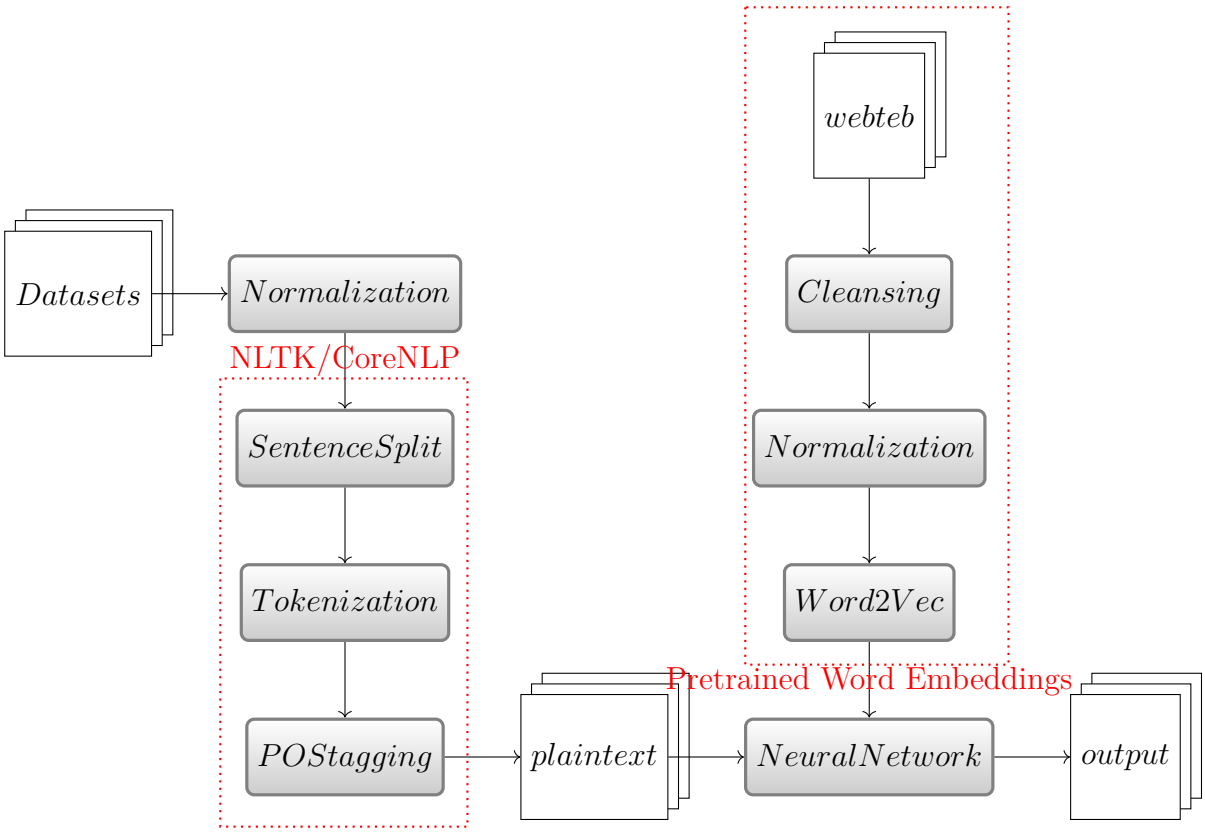


Figure A.1: Architecture globale du système.

obtenus en utilisant les approches par paires et celles par points montre que les premières surpassent les secondes pour toutes les mesures d'évaluation, ce qui confirme notre hypothèse initiale.

L'architecture globale du système est illustrée dans la Fig.A.1 conduisant aux résultats reportés dans la Table A.5.

A.2.2 Réseau de neurones convolutionnel empilé

L'utilisation du modèle KNN+COSINE+FastText combiné à la caractéristique n-gram cosine et FastText a surpassé les modèles état de l'art en termes de précision de près de 3 points et de performances proches pour le mAP. Ces résultats ont été obtenus en utilisant nos plongements pré-entraînés et gelés lors de l'entraînement du modèle. KCNN+COSINE a la même architecture sauf que les pondérations des couches de plongements sont apprises pendant l'entraînement, les performances de ces modèles ont chuté ce qui justifie l'utilisation des plongements pré-entraînés. L'intuition derrière l'utilisation de cette architecture pourrait être justifié par le haut degré d'inflection qui caractérise la langue Arabe. Outre ses performances élevées par rapport aux architectures de séquences telles que RNN, LTM et GRU, une pile de CNN est capable de détecter des occurrences de k-gram où k varie de 3 à 6, ce qui a pu réduire l'effet d'inflexion. Le modèle mis en place est illustré dans Fig.A.2.

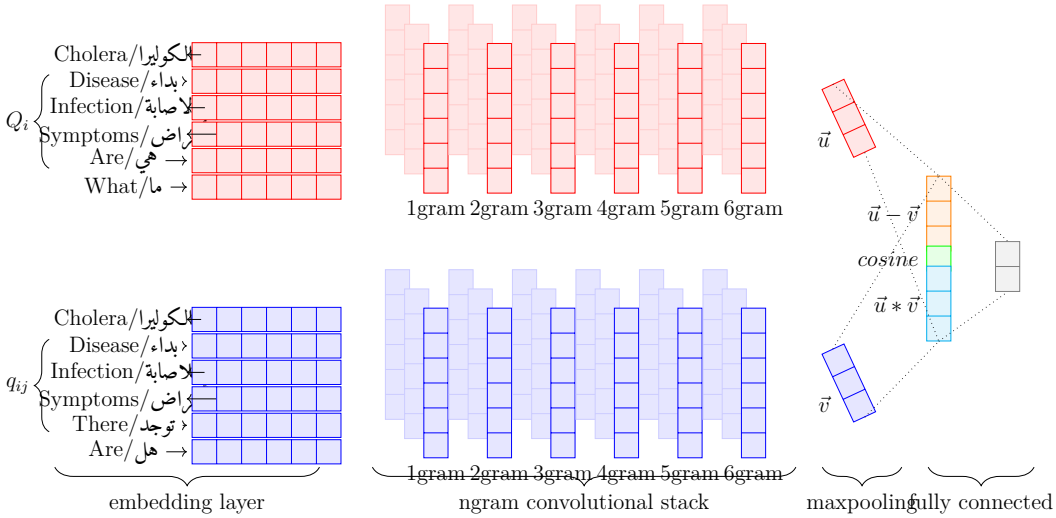


Figure A.2: Modèle Réseaux de Neurones Empilés.

L'utilisation des plongements pré-entraînés accessibles au public contribue aux performances du modèle, mais l'utilisation de nos propres plongements conduit à de meilleurs résultats qui pourraient s'expliquer par deux faits. D'une part, l'entraînement de nos propres plongements réduit le manque de vocabulaire. D'autre part, l'utilisation d'un corpus du domaine spécifique pour l'entraînement contribue à la réduction du bruit et à l'optimisation des performances. L'utilisation d'une approche de traductions automatique combinée avec des plongement pré-entraînés pour le domaine biomédical en Anglais améliore considérablement les performances. Tandis que plusieurs encodeurs ont produit des résultats satisfaisants pour d'autres tâches du traitement automatique du langages naturel, leur utilisation pour les cQA conduit vers des résultats médiocres. Ceci peut être expliqué par les caractéristiques de la langue arabe. Les résultats obtenus sont présentés dans la table Table A.1.

A.2.3 Plongements contextualisés

En utilisant des plongements contextualisés, BERT par exemple, nous avons pu améliorer le mAP, le score officiel du CQA-MD de près d'un point pour l'ensemble de test 2017 comme reporté dans la table A.3 et de deux points pour l'ensemble de test 2016 tel que reporté dans la table A.2. Ces performances pourraient s'expliquer par leur capacité à prendre en compte le contexte lors de l'apprentissage et la mise au point du modèle, les corpus utilisés pour le pré-apprentissage et le nombre de couches et de nœuds cachés utilisés, l'architecture globale de notre système est illustrée dans la figure Fig.A.3. Les mauvais résultats de XLM pourraient être expliqués par la langue du modèle utilisé, de meilleurs résultats pourraient être obtenus en pré-formant XLM sur les corpus du domaine médical arabe et anglais. L'utilisation de plongements sans contexte pré-entraînés sur des corpus de domaines généraux conduit à des résultats inférieurs par rapport à ceux pré-formés sur des corpus de domaines spécifiques, car ce dernier réduit le nombre de termes hors vocabulaire.

Table A.1: Résultats pour le jeux de données SemEval2017 Task3 SubtaskD en utilisant le modèle réseaux de neurones empilés.

Model	MAP	AvgRec	MRR	P	R	F1	Acc
KCNN+COSINE+FastText (Arabic)	60.12	83.98	65.80	62.53	55.25	58.66	69.45
KCNN+COSINE (Arabic)	54.22	79.13	59.35	54.89	23.93	32.93	62.41
KCNN+COSINE+FastText (Wikipedia&CommonCrawl)	57.83	82.01	63.66	57.24	56.16	56.69	66.34
KCNN+COSINE+FastText (English)	57.34	81.79	62.68	55.76	63.96	59.58	65.95
KCNN+COSINE+GloVe (English)	58.09	82.37	63.64	55.12	67.28	60.60	65.67
KCNN+COSINE+Pyysalo-PubMed-PMC (English)	58.19	82.25	63.51	59.09	58.29	58.68	67.79
KCNN+COSINE+Pyysalo-PubMed (English)	58.79	82.79	64.34	61.54	49.92	55.12	68.11
ESIM	56.34	81.01	60.98	53.69	71.56	61.35	64.63
BiMPM	47.23	72.32	50.96	39.23	1.00	56.36	39.23

La faible performance du système en ce qui concerne la mAP et la précision pourrait s'expliquer par de nombreux facteurs i) l'asymétrie de l'ensemble de données car la majorité des paires question / réponse ne sont pas pertinentes pour la question initiale ii) la longueur de la question initiale et la question / les paires de réponses ne sont pas proportionnelles iii) la présence de fautes de frappe dans les fragments de texte menant à de nombreux termes hors vocabulaire, ces facteurs pourraient être surmontés par i) pré-entraînement des modèles contextualisés sur les corpus biomédiaux au lieu d'utiliser des modèles pré-entraînés ii) extraire des phrases clés de la question originale et du couple question / réponse iii) utiliser des bases de connaissances pour annoter le texte avant de procéder au calcul de similarité.

A.2.4 Approche à base des arbres

Alors que l'application des noyaux d'arbres a conduit à de faibles performances par rapport à celles obtenues en utilisant un réseau de neurones récursif, nous remarquons une nette amélioration de performances comme reporté sur la table A.4, ceci peut être expliqué par la nature du jeux de données contenant de longues phrases et l'utilisation de la reformulation en cQA. L'utilisation d'un réseau de neurones récursif produit les meilleurs résultats, ce qui confirme notre hypothèse concernant l'exploitation de la structure hiérarchique du texte. En combinant de ce modèle avec les noyaux d'arbres, on a pu améliorer les métriques de la

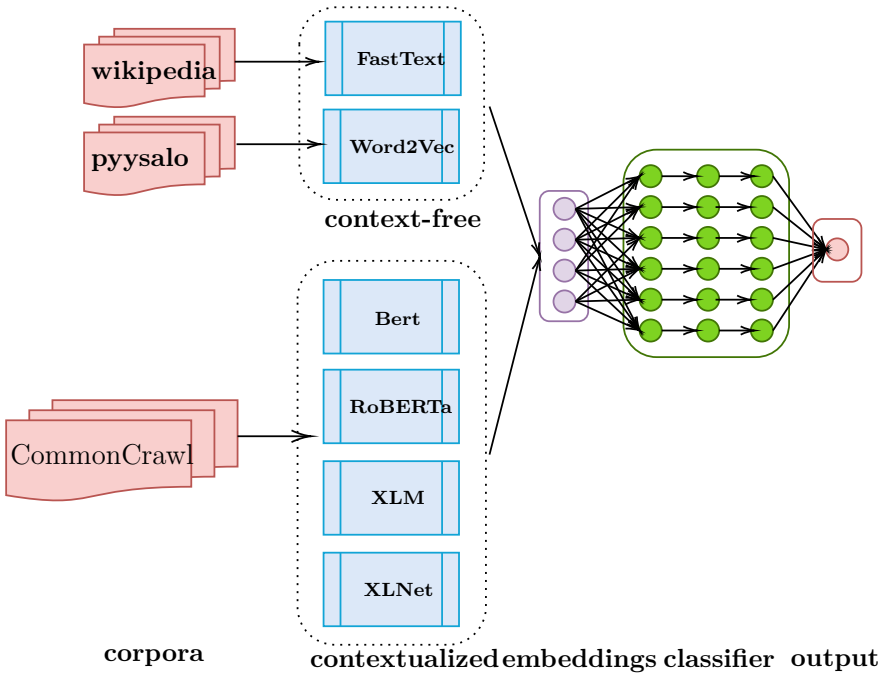


Figure A.3: Architecture du modèle: les plongements sans et avec contexte alimentent le classificateur à base des réseaux de neurones pour mesurer la pertinence pour chaque pair de question-réponse.

		Ranking			Classification			
	Submission	MAP	AvgRec	MRR	P	R	F1	Acc
1	Wikipedia	43.73	46.93	50.00	26.73	76.80	39.66	55.03
2	Pyysalo	43.05	47.67	49.64	27.80	81.66	41.48	55.65
3	BERT	46.83	52.41	53.83	30.37	65.30	41.46	64.51
4	XLNet	45.05	51.37	51.52	25.33	96.12	40.09	44.71
5	XLM	46.83	52.41	53.83	30.37	65.30	41.46	64.51
6	RoBERTa	47.67	51.74	51.75	27.56	93.51	42.57	51.45
7	SLS-primary	45.83	51.01	53.66	34.45	52.33	41.55	71.67
8	ConvKN-primary	45.50	50.13	52.55	28.55	64.53	39.58	62.10
9	RDI_team-primary	43.80	47.45	49.21	19.24	100.00	32.27	19.24
10	QU-IR-primary	38.63	44.10	46.27	25.50	45.13	32.59	64.07
11	UPC_USMBA-primary	29.09	30.04	34.04	20.14	51.69	28.99	51.27

Table A.2: Résultats du classement et de la classification du jeux de données Semeval 2016.

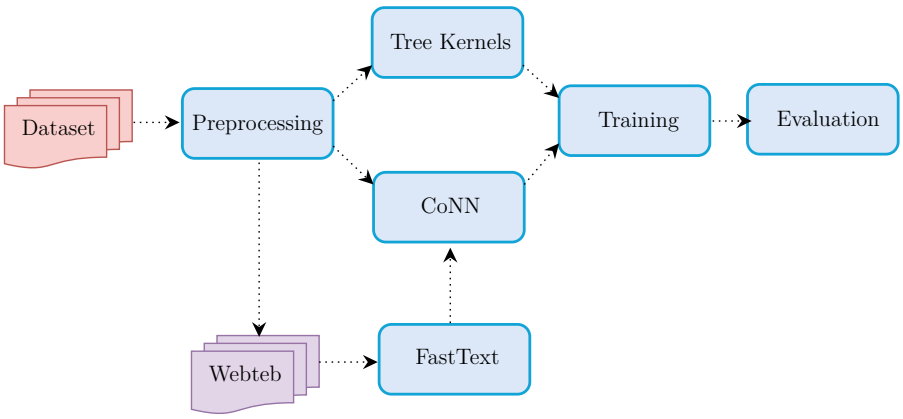


Figure A.4: Les composants principaux de l'architecture sont le pré-traitement, les fonctions arborescentes et le réseau de neurones à bases des arbres de constituants augmenté les plongements générés depuis webteb.

classification ainsi que celles du classement. Moyennant une approche shift/reduce, on était capables d'utiliser en entrée du modèle des batchs tout en exploitant la structure récursive du modèle dans un temps linéaire. La figure Fig.A.4 illustre les différents composants de notre système.

A.3 Conclusion

Les systèmes question-réponse communautaires ne cessent de se développer en raison de leur capacité à répondre convenablement aux besoins d'information dans le domaine biomédical. Ils aident les utilisateurs à rechercher des informations pour diagnostiquer les maladies en décrivant les symptômes, les maladies et les médicaments. Motivé par ces faits, le présent travail a pour objectif l'examen et l'amélioration des systèmes question-réponse communautaires pour les documents biomédicaux en arabe où nous avons abordé un large éventail de problèmes et de défis tels que l'utilisation des mots latins pour les médicaments et les maladies, la présence des fautes de frappe et la nature déséquilibrée des jeux de données. Ces améliorations ciblent différents composants du pipeline y compris la classification des questions-réponses, la sélection des réponses, la récupération des passages et le classement ainsi que le reclassement des pairs de question-réponse. Nous avons proposé un ensemble de méthodes pour permettre aux utilisateurs de trouver des réponses à leur besoin en information. Les résultats expérimentaux sur deux jeux de données largement utilisés dans le cadre des compétitions SemEval se sont avérés efficaces. Dans cette section, nous présentons les principales conclusions tirées de nos contributions et perspectives pour les prochains travaux.

A.3.1 Contributions

Les méthodes proposées sont basées sur l'apprentissage automatique et l'apprentissage profond appliqués aux termes ainsi qu'aux arbres syntaxiques extraits de la question originale et des paires de question-réponse.

A.3.1.1 Modélisation des sujets et approches par paires

Nous avons abordé le problème des systèmes question-réponse communautaires pour le domaine biomédical en exploitant le lien entre la question originale et les paires question-réponse. Nous personnalisons d'abord le modèle LSA pour projeter la question originale et les paires dans un espace multidimensionnel de 100 composants à l'aide de l'algorithme susmentionné. De plus, nous avons utilisé la dérivation pour atténuer l'impact du bruit et des mots présentant des erreurs d'orthographe sur les performances du système. Nous avons également prouvé que l'exploitation des réseaux de neurones par paires améliore considérablement les performances du classement. En utilisant une variante du modèle LSA, nous avons pu obtenir des valeurs élevées pour le principal indicateur « mAP » ainsi que les autres indicateurs. En utilisant l'approche ponctuelle combinée à divers modèles de réseaux de neurones alimentés par les différents termes en intersection entre la question et les paires de question-réponse, nous avons réussi à atteindre des performances comparables à ceux obtenus à l'aide des approches par paires, ce qui confirme notre hypothèse initiale.

A.3.1.2 Combinaison des réseaux de neurones convulotionnels avec les caractéristiques lexicales

La langue arabe souffre d'une pénurie de ressources qui est en outre élargie pour les tâches des domaines spécifiques. Notre objectif est de combler ces lacunes en mettant en place un ensemble de plongements comme représentations pour les différents termes en Arabe pour le domaine biomédical. Ces vecteurs pré-entraînés appris sur un corpus du domaine biomédical pour l'arabe contribuent de manière significative aux performances du modèle. Ces plongements sont ensuite consommés par une pile de réseaux de neurones convolutifs et une fonction lexicale menant à de hautes performances. D'un autre côté, l'utilisation des encodeurs souvent utilisés pour des tâches similaires montre l'efficacité de notre architecture.

A.3.1.3 Plongements contextualisés

Les plongements contextualisés se sont avérés efficaces pour un large éventail de tâches de traitement du langage naturel. Motivés par ces faits, nous avons abordé dans le présent travail le problème des systèmes question-réponse communautaires appliqués au domaine biomédical arabe en tirant parti de la puissance des plongements contextualisés. Nous calculons la similarité entre la question originale et les paires question/réponse pour le

classement et la classification. Nous démontrons également les plongements contextualisés, utilisés comme représentations pour les phrases, conduisent à de meilleurs résultats par rapport aux plongements qui ne prennent pas en considération le contexte. Ceci est en partie dû à leurs traits sensibles au contexte fournissant une similarité mesurée au niveau sémantique en contournant les erreurs syntaxiques et les défis introduits par la reformulation ce qui nous a conduit à des améliorations de résultats allant jusqu'à 2% concernant le mAP, la métrique officielle.

A.3.1.4 Approches à base des Arbres

Nous avons abordé le problème des systèmes de question-réponse communautaires pour le domaine biomédical arabe en exploitant la structure sous forme d'arbres en appliquant deux noyaux d'arbres largement utilisés alimentés par des arbres obtenus à l'aide d'une analyse syntaxique des questions et des paires de question/réponse. Le modèle $SVM^{r}ank$ est ensuite utilisé pour la classification des différentes paires. Nous avons également utilisé un réseau de neurones récursif alimenté par des arbres binaires de constituants pour le classement des paires.

A.3.2 Prochains Travaux

Nous prévoyons d'étendre ce travail en pré-entraînant les mêmes algorithmes des corpus biomédicaux arabes afin d'améliorer la couverture minimisant ainsi le rapport des termes hors vocabulaire. Nous travaillons également afin de combiner une approche à base des graphes et des plongements contextualisés pour exploiter la structure du texte moyennant une base de connaissances. Pour améliorer les résultats obtenus, nous planifions également de :

- Utiliser les métriques d'évaluation des systèmes de traduction automatique comme caractéristiques afin de mesurer la similarité entre la question originale et la paire question-réponse
- Appliquer le résumé automatique extractif aux phrases afin de palier au problème des différences de taille entre la question et la réponse
- Entraîner un ensemble de plongements contextualisés sur des corpus biomédicaux arabes, les plongements existants étant exclusivement entraînés sur des corpus du domaine général

		Ranking			Classification			
	Submission	**MAP**	**AvgRec**	**MRR**	**P**	**R**	**F1**	**Acc**
1	**Wikipedia**	**56.82**	**81.21**	**62.73**	**58.28**	**49.55**	**53.56**	**66.29**
2	**Pyysalo**	**58.79**	**82.79**	**64.34**	**61.54**	**49.92**	**55.12**	**68.11**
3	**BERT**	**62.07**	**85.75**	**67.23**	**59.33**	**55.57**	**57.39**	**67.63**
4	**XLNet**	**59.46**	**84.02**	**64.81**	**51.54**	**83.63**	**63.77**	**62.72**
5	**XLM**	**49.52**	**74.86**	**53.83**	**40.29**	**76.26**	**52.72**	**46.34**
6	**RoBERTa**	**61.19**	**84.96**	**66.26**	**54.83**	**85.80**	**66.90**	**66.70**
7	**GW_QA-primary**	**61.16**	**85.43**	**66.85**	**0.00**	**0.00**	**0.00**	**60.77**
8	**UPC-USMBA-primary**	**57.73**	**81.76**	**62.88**	**63.41**	**33.00**	**43.41**	**66.24**
9	**QU_BIGIR-primary**	**56.69**	**81.89**	**61.83**	**41.59**	**70.16**	**52.22**	**49.64**

Table A.3: Résultats du jeux de données Semeval 2017 pour le classement et la classification.

	Submission	**MAP**	**AvgRec**	**MRR**	**P**	**R**	**F1**	**Acc**
1	**Subset Tree Kernel**	38.73	**40.68**	**43.96**	**28.78**	**34.63**	**31.43**	**70.93**
2	**Partial Tree Kernel**	33.98	**36.93**	**38.62**	**23.11**	**37.31**	**28.54**	**64.05**
3	**Constituency Neural Network**	**48.50**	**74.03**	**53.05**	**41.96**	**12.05**	**18.73**	**58.95**
4	**CoNN + STK**	**48.50**	**74.03**	**53.05**	**28.78**	**34.63**	**31.43**	**70.93**
5	**SLS-primary**	45.83	**51.01**	**53.66**	**34.45**	**52.33**	**41.55**	**71.67**
6	**ConvKN-primary**	45.50	**50.13**	**52.55**	**28.55**	**64.53**	**39.58**	**62.10**
7	**RDI_team-primary**	43.80	**47.45**	**49.21**	**19.24**	**100.00**	**32.27**	**19.24**
8	**QU-IR-primary**	38.63	**44.10**	**46.27**	**25.50**	**45.13**	**32.59**	**64.07**
9	**UPC_USMBA-primary**	29.09	**30.04**	**34.04**	**20.14**	**51.69**	**28.99**	**51.27**

Table A.4: Comparaison des performances entre nos approches à base des arbres et les résultats lors de la compétition SemEval 2016.

Model	**MAP**	**Acc**	**P**	**R**	**F1**
LDA	53.58	63.06	60.94	16.25	25.66
NMF	56.42	62.52	58.95	14.75	23.59
LSA+NLTK	61.31	62.16	83.08	04.47	08.49
LSA+CoreNLP	61.66	62.34	85.87	04.80	09.09
SVM^{rank}	56.22	61.08	100	00.81	06.10
BiGRU-intersection	56.93	59.07	48.58	73.58	58.52
BiGRU-concatenation	48.53	39.23	39.23	100	56.36

Table A.5: Résultats du jeux de données SemEval 2017 en utilisant la modélisation de sujets.

Contributions to the Improvement of Arabic Information Retrieval:
Application to Arabic Biomedical Community Question-Answering

Abstract: Community Question Answering has attracted much attention in recent years due to their ability to stand in between the less verbose search engines offering a keyword-based lookup and the more verbose question answering systems. Besides their obvious purpose of suggesting potential answers to a user question, they enable users to save considerable time by avoiding duplicate questions. Previous community question answering systems were mostly based on lexical and syntactic approaches leading to weak performances as they are unable to identify paraphrasing. This dissertation goes beyond these approaches seeking to improve text comprehension and retrieval by leveraging the semantic dimension of text using learning to rank, neural and knowledge-based models. To this end we propose a bunch of novel methods including topic modeling, distributional embeddings, and knowledge bases. This thesis research starts by exploring the performance of classic approaches to information retrieval based on machine learning, presenting a topic modeling approach to improve text representation and depicting our learning to rank models by showing how pairwise models are more powerful than their counterpart pointwise approaches. Then this dissertation presents our distributional embeddings trained on a biomedical corpus and used to power a novel neural architecture combining lexical features with a stack of convolutional networks. Afterward, we show how contextualized embeddings perform better than context-free embeddings by fine-tuning them on our task. This thesis research concludes with a tree-based approach either using kernel trees combined with a machine learning algorithm or using binary constituency trees as inputs to a recursive neural network.

Keywords— Community Question Answering, Information retrieval, Arabic Natural Language processing, Topic Modeling, Deep Learning, Contextualized Embeddings, Tree-Based Approach

Contributions à l'amélioration de la recherche de l'information :
Application aux systèmes question réponse communautaires biomédicaux Arabes

Résumé: Les systèmes de question réponse communautaires ont suscité beaucoup d'intérêt durant les dernières années vu leur capacité de se positionner comme alternative aux systèmes de recherche d'information basés sur les mots clés et les systèmes de question réponse. Hormis leur but de suggérer une réponse à la question posée par l'utilisateur, ils permettent d'épargner le temps aux utilisateurs tout en optimisant la performance du système à travers l'élimination des doublons. Cette thèse a pour objectif l'amélioration de la représentation, la restitution ainsi que la compréhension du texte en utilisant la modélisation des thèmes, les plongements distributionnels des mots et les bases de connaissances. Les systèmes de question réponses préalables ont été généralement basés sur l'analyse lexicale et l'analyse syntaxique du texte, ce qui engendrait de faibles performances dues à leurs incapacités de détecter les reformulations entre autres. Cette thèse a pour but d'aller au-delà de ces représentations en exploitant la dimension sémantique du texte à travers les modèles d'apprentissage pour la classification et le classement, les réseaux de neurones et les bases de connaissances. Cette thèse commence par explorer la performance des modèles classiques pour la recherche d'information, présenter une approche à base de la modélisation des sujets et démontrer par la suite comment l'approche par pairs est plus performante que celle par points quand il s'agit de l'apprentissage du classement. Ensuite, la thèse présente l'utilité d'un ensemble de plongements de mots générés à partir d'une collection d'articles du domaine biomédical afin de générer des représentations vectorielles denses des mots et qui sont par la suite injectées dans une pile de réseaux de neurones convolutifs. On démontre également l'utilité du contexte pour ces représentations en les ajustant pour notre plateforme. Cette thèse se termine par une présentation de l'approche à base des arbres à travers l'utilisation des fonctions noyaux d'arbres ou les réseaux de neurones récursifs avec comme entrée les arbres binaires de constituants.

Mots clés— Systèmes Réponse aux questions de la communauté, Recherche d'information, Traitement du langage naturel arabe, Modélisation de sujets, Apprentissage en profondeur, Intégration contextualisée, Approche arborescente.