République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieure et la recherche scientifique
Université Saad Dahleb Blida 1
Faculté Des Sciences
Département d'informatique

Projet de fin d'étude

Pour l'obtention du diplome de Master

Domaine : Mathématique et Informatique

Filière : Informatique

Spécialité : Traitement Autoamtique de la Langue

Présenté par : HARIZI Walid & NAIT HAMOUD Yasmina

Sujet :

Conception et réalisation d'un système de questions-réponses comparatif : cas de la langue Arabe

Soutenu le 27/09/2022 devant les membres de jury composé de :

Mme OUAHRANI.L	Université de Blida 1	Présidente
Mme BERREMDANE.D	Université de Blida 1	Examinatrice
Mme TEBBI.H	Université de Blida 1	Promotrice
Mr HAMOUDA SIDHOUM.A	EMP	Encadrant

Année Universitaire
2021/2022

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et le courage pour terminer nos études et élaborer ce modeste travail.

Nous tenons à présenter nos sincères remerciements à notre encadrant **Mr HAMOUDA SIDHOUM Abdellah** pour sa patience, sa disponibilité, ses judicieux conseils, et surtout d'avoir veillé à l'aboutissement de notre projet.

Nous tenons à remercier aussi notre chère promotrice **Mme TEBBI Hanane** pour l'orientation, la confiance, la patience qui ont constitué un rapport considérable sans lequel ce travail n'aurait pas pu être mené au bon port.

Nous remercions particulièrement **Mr MATAOUI Mhamed** et **Mr NAIT HAMOUD Othmane** pour leurs bonnes explications qui nous ont éclairé le chemin de la recherche.

Nous aimerons exprimer notre gratitude à tous nos professeurs de l'université de **SAAD DAHLAB BLIDA** .Nous remercions tout particulièrement **Mme MEZZI Melyara**, Pour avoir enrichi nos connaissances et de nous avoir guidé durant toute ces années, Pour être l'enseignante qui arrive à voir l'avenir dans les yeux de chaque étudiant et qui a réussi à nous inspirer, à nous donner confiance en soi et en l'avenir mais aussi qui a réussi à nous donner l'envie d'apprendre. Merci d'être une enseignante vraiment exceptionnel.

Nous remercions également toute l'équipe pédagogique de **l'École Militaire Polytechnique** et les intervenants professionnels responsable de notre formation, pour avoir assuré ce travail.

Nous tenons a remercier nos parents, nos familles, et amis pour leurs soutien tout au long de ce mémoire.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Dédicaces

***A MA TRES CHERE MERE**, Source inépuisable de tendresse, de patience et de sacrifice. Ta prière et ta Bénédiction m'ont été d'un grand secours tout au long de ma vie. Quoique je puisse dire et écrire, je ne pourrais exprimer ma grande affection et ma profonde reconnaissance. J'espère ne jamais te décevoir, ni trahir ta confiance et tes sacrifices. Puisse Dieu tout puissant, te préserver et t'accorder santé, longue vie et Bonheur.*

***A MON TRES CHER PERE**, De tous les pères, tu es le meilleur.Tu as été et tu seras toujours un exemple pour moi par tes qualités humaines, ta persévérance et perfectionnisme. En témoignage de brut d'années de sacrifices, de sollicitudes,d'encouragement et de prières.*

En ce jour, j'espère réaliser l'un de tes rêves. Aucune dédicace ne saurait exprimer mes respects, ma reconnaissance et mon profond amour. Puisse Dieu vous préserver et vous procurer santé et bonheur.

***A mes sœur Leila et Ikram** qui depuis des années m'encouragent, me comprennent et a toujours étaient à mes côtés ; et **à mon petit frère Youcef**, mon fils adoré à qui je souhaite un avenir radieux plein de réussite*

A mon binôme Walid et toute la famille HARIZI

*Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures, mes aimables amis, collègues d'étude, et frères de cœur, **Silia , Nawal , Baya ,Lamia ,Manel ,Said ,Kader et Nadhir.***

Je vous dédie ce travail, témoignage de ma profonde reconnaissance et mon plus grand attachement.

Yasmine

Dédicaces

Je dédie du profond de mon coeur ce mémoire :

***A MON CHER PERE MOHAMED**,qui a su montrer à ses enfants que le travail est libérateur et qui n'a ménagé aucun effort pour l'aboutissement de ce travail, qu'il en soit récompensé pour ses sacrifices.*

***A MA CHERE MERE HOURIZI ASSIA**,pour ton amour pour moi, pour les sacrifices que tu consens pour rendre tes enfants heureux. tu as enduré beaucoup de peine pour mon bien-être et à ma réussite. Reçois ceci en guise de ma reconnaissance et que Dieu te garde longtemps afin que tu puisses goûter aux arbres que tu as plantés. .*

***A mon frère aîné Nabil**,pour dévouement, pour ton écoute et ton soutien. Reçois ici ma profonde gratitude. A qui je souhaite un avenir radieux plein de réussite*

A mon binôme Yasmine et toute la famille NAIT HAMOUD

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures, mes aimables amis, collègues d'étude, et frères de cœur, ***Ayoub ,Ibrahim ,Said ,Nadhir ,Kader et Baya.***

Walid

Résumé

Dans le cadre de notre projet de master, nous proposons un système de QA comparatif spécifique à la langue arabe basé sur le modèle AraBERT, en utilisant le modèle pré-entraîné BERT pour affiner les tâches NLP ultérieures telles que la QA, le NER et la classification. Notre système se compose de quatre modules : (1) un module de classification des questions, (2) un module d'extraction des éléments de comparaison, (3) un module de RI et de QA, et (4) un module de comparaison. Chaque module du système proposé a entraîné des modèles de réseaux neuronaux à l'aide de différents ensembles de données, notamment Arabic-SQuAD, ARCD, une traduction automatique de HotpotQA et un ensemble de données de questions de comparaison générées automatiquement.
Le module de classification a fait preuve d'une précision de 99%, et les résultats de l'extracteur se sont avérés très positifs avec un taux de précision de 79% .De même les résultats du retriever du module de recherche d'information se sont révélés satisfaisante avec un F1 score de 69.46% . Et puis finalement le module de comparaison qui était bien réussi avec un score de précision très satisfaisant qui s'élève a 92%

Mots clés : Recherche d'information, Traitement automatique du langage, Système de questions-réponses, Questions comparatifs, Langue Arabe.

Abstract

In our master's project, we propose an Arabic-specific comparative QA system based on the AraBERT model, using the BERT pre-trained model to refine subsequent NLP tasks such as QA, NER and classification. Our system consists of four modules : (1) a question classification module, (2) a comparison item extraction module, (3) an IR and QA module, and (4) a comparison module. Each module of the proposed system trained neural network models using different datasets, including Arabic-SQuAD, ARCD, a HotpotQA machine translation, and an automatically generated comparison question dataset.

The classification module demonstrated 99% accuracy, and the extractor results were very positive, with an accuracy of 79%. Also the results of the retriever were very satisfactory with an F1 score of 69%. And then finally the comparison module was very successful with a very satisfactory accuracy score of 92%.

Key words : Information retrieval, Natural language processing, Questions answering systems, Comparative questions, Arabic language.

الملخص

نقترح استخدام نظام أسئلة و أجوبة مقارن خاص باللغة العربية على أساس نموذج AraBERT، باستخدام نموذج BERT مدرب مسبقًا لتحسين مهام المعالجة الآلية للغة مثل أنظمة الأسئلة و الأجوبة و التعرف على الكيانات المسماة والتصنيف. يتكون نظامنا من أربع وحدات: (1) وحدة تصنيف الأسئلة، (2) وحدة استخراج عنصر المقارنة، (3) وحدة البحث عن المعلومات والأجوبة ، و (4) وحدة مقارنة. تم تدريب كل وحدة من نماذج الشبكة العصبية للنظام المقترح باستخدام مجموعات بيانات مختلفة، بما في ذلك Arabic-SQuAD وARCD وبرمجة آلية لـHotpotQA ومجموعة بيانات أسئلة مقارنة تم إنشاؤها أوتوماتيكيا .أظهرت وحدة التصنيف دقة 99 %، وكانت نتائج المستخرج إيجابية للغاية.وكانت نتائج المستخرج إيجابية للغاية بنسبة دقة 79٪. وبالمثل، كما أثبتت نتائج وحدة البحث عن المعلومات أنها مرضية للغاية حيث بلغت نسبة الدقة 69,46%. ثم أخيرًا وحدة المقارنة التي تم إجراؤها بشكل جيد وبدرجة دقة مرضية للغاية تصل إلى 92%.

الكلمات المفتاحية: البحث عن المعلومات, المعالجة الآلية للغة, نظام الأسئلة و الأجوبة, أسئلة المقارنة, اللغة العربية.

Table des matières

Liste des figures

Liste des tableaux

Nomenclature

TAL	**T**raitement **A**utomatique **d**e la **L**angue
RI	**R**echerche **d**'Information
QA	**Q**uestion **A**nswering
PR	**R**écupération de **P**assage
EM	**E**xact **M**atch
QAS	**Q**uestion **A**nswering **S**ystem
NER	**N**amed **E**ntity **R**ecognition
CAM	**C**omparative **A**rgument **M**ining
NLP	**N**atural **L**anguage **P**rocessing
CoQAS	**Co**mparative **Q**uestion **A**nswering **S**ystem

INTRODUCTION GÉNÉRALE

Introduction Générale

L'information sur Internet a considérablement augmenté au cours des dernières décennies. Il est devenu de plus en plus difficile d'obtenir des informations correctes en temps réel. Par conséquent, avoir accès aux données et gérer de grandes quantités de données qui existent déjà est devenu un besoin crucial. Cela a conduit à l'élaboration des systèmes de Questions-Réponses (QAS), un sujet de recherche en informatique qui combine les sciences de la Recherche d'information (IR) et le Traitement Automatique du Langage Naturel (NLP).

Les systèmes de Questions-Réponses sont un type de technologie de recherche d'information qui automatise le processus de fourniture de réponses appropriées aux requêtes posées par les humains en langage naturel en utilisant leur propre terminologie. Les QAS peuvent être considérés comme une extension des moteurs de recherche. La distinction entre ces derniers est que les moteurs de recherche renvoient un groupe de documents pertinents, laissant aux utilisateurs la tâche d'extraire la solution. En opposition, les QAS renvoient une seule réponse, courte et précise aux utilisateurs, permettant de gagner du temps lors de la navigation sur le web.

Par conséquent, la qualité d'un système QA est affectée par le nombre de langues prises en charge, car chaque langue a ses propres caractéristiques qui doivent être prises en compte lors de l'évaluation, de la recherche et de l'extraction de contenu. Cependant, la grande majorité des recherches sur les QAS ont été menées sur l'anglais et d'autres langues basées sur le latin avec des structures et des lexiques similaires. Par conséquent, les recherches sur d'autres langues dotées de systèmes morphologiques et syntaxiques différents, comme l'arabe, sont peu performantes. Malgré le fait que cette langue soit parlée par plus de 300 millions de personnes dans le monde et qu'il existe une quantité importante de contenu arabe sur Internet, la recherche sur la langue arabe est encore limitée et confrontée à des défis, ce qui nécessite le développement de systèmes qui prennent en compte la complexité de la langue arabe. Cela nous conduit à mener une étude comparative sur les systèmes de questions-réponses en arabe en tenant compte des difficultés linguistiques rencontrées, de l'absence de systèmes de QA en arabe en ligne et du grand manque d'ensembles de données sur les QA arabe. Nous traiterons principalement des questions comparatives qui ne peuvent pas être considérées comme des faits, ou celles qui ne peuvent pas être traitées automatiquement par des technologies et méthodes connues dans des systèmes QA simples.

Nous proposons un système qui comprend quatre modules : (1) Classification des questions, (2) Extraction des entités comparées et de l'aspect de la comparaison, (3) Recherche d'informations et réponse aux questions, (4) Comparaison des réponses. Il détecte automatiquement les éléments de comparaison et les caractéristiques clés, ce qui permet aux utilisateurs de découvrir plus rapidement et avec plus de certitude les réponses aux questions comparatives. Dans notre conception, le système repose sur des sources de données qui ont été soigneusement sélectionnées et permettent une comparaison de haute précision. Notre système utilise le modèle AraBERT, qui a été pré-entraîné sur des grands corpus en langue arabe. Nous affinons le modèle AraBERT pour différentes tâches présentes dans chacun des modules constituants le système. Nous utilisant une variété d'ensembles de données dans le domaine de QA en langue Arabe en l'occurence Arabic-SQuAD, Arcd, une traduction automatique de HotpotQA et une partie d'une dataset généré automatiquement.
Les résultats préliminaires sont très encourageants pour le développement futur et la continuité de ce travail.

Le présent mémoire est organisé en quatre chapitres :
Le chapitre 1 : Les concepts de base des systèmes de Questions-Réponses et leurs différentes classifications seront abordés dans ce chapitre. Nous discuterons également les questions comparatives et de leurs défis, toute en motivant ce choix.
Le chapitre 2 : Ce chapitre donne un aperçu de l'état actuel de la recherche sur les ensembles de données Arabes, les systèmes de Questions-Réponses Arabes et les systèmes de Questions-Réponses Comparatives. Il examine les résultats et les défis de chaque étude.
Le chapitre 3 : Ce chapitre fournit l'architecture de notre système QA comparatif proposé, ainsi qu'une description détaillée de chaque composant.
Le chapitre 4 : détaille la réalisation du système proposé et les différents aspects relatifs à son évaluation.

Enfin, le présent mémoire sera conclu par une conclusion générale et quelques perspectives du développement et de recherche future dans cette thématique.

Chapitre I

TRAITEMENT AUTOMATIQUE DU LANGUAGE NATUREL ET LES SYSTÈMES DE QUESTIONS-RÉPONSES

I.1 Introduction

L'augmentation de la quantité d'informations disponibles sur le Web a nécessité la construction des systèmes qui aident à combler le fossé entre les utilisateurs finaux et le contenu du Web avec ses diverses représentations de données. Parmi la grande quantité de données Web fournies, une grande partie est écrite en Arabe pour et par des arabophones. Ce chapitre donne un aperçu général des systèmes QA, en faisant référence aux systèmes de Recherche d'information et au Traitement Automatique du Langage Naturel.

Nous examinons les composants d'un système QA, ainsi que ces diverses classifications et méthodologies d'évaluation. De plus, nous visons à analyser les questions comparatives et à avancer dans notre compréhension des systèmes comparatifs en les définissant et en examinant leurs enjeux. Finalement, nous discuterons de certains des challenges auxquels les chercheurs sont confrontés lorsqu'ils tentent de développer des systèmes de QA Arabe, en raison des caractéristiques linguistiques uniques de cette langue.

I.2 Recherche D'Information (RI)

Calvin Mooers a créé le terme "Recherche d'information" (en Anglais :*Information Retrieval*) en 1950 [Mooers, 1950], et il s'est répandu dans la communauté scientifique quand les ordinateurs ont été introduits pour le traitement des informations en 1961 [Mooers, 1961]. En réponse à la demande d'étude, ces systèmes devaient restaurer le document en entier (le livre, l'article, etc.). Bien que ce soit ce que font les systèmes de RI d'aujourd'hui, de nombreuses approches avancées ont été créées et mises en œuvre pour évaluer, traiter et stocker les sources d'information et restaurer celles qui répondent aux exigences d'un utilisateur particulier. Au fil des ans, le texte, la musique, les photos et les vidéos peuvent être utilisés pour créer du contenu multimédia. En conséquence, les systèmes RI concerne le stockage, l'organisation et la restauration du texte, ainsi que les ressources d'information multimédia [Manning et al., 2018].

Sur la base de l'étude menée par [NIOS, 2020], il est conclu que les systèmes de RI reposent sur des index, des résumés et d'autres outils bibliographiques qui transmettent le contenu des revues et nous permettent de récupérer n'importe quel article en utilisant une variété de méthodes, notamment :

— **Méthode de dictionnaire :** Le contenu est classé par ordre des lettres, comme son nom l'indique, de sorte que certains sujets peuvent être rapidement trouvés. Cependant, il y a toujours des problèmes avec les synonymes, les synonymes indi-

viduels ou multiformes, les termes ou sujets complexes, ainsi que les concepts de plusieurs mots.

— **Affichage des relations sujet :** L'approche est objective, également connue sous le nom d'indexation par sujet, est un moyen de reconnaître et de sélectionner les termes (les mots, les phrases, les phrases, les groupes et les catégories) pour décrire le sujet du document. En d'autres termes, il s'agit de déterminer et de décrire le sujet du document. Il vise à faciliter la recherche d'informations spécifiques en fonction de leur contenu.

I.3 Traitement Automatique du Langage Naturel

Elizabeth D. Liddy a décrit dans son étude [Liddy, 2001] que le TAL (en Anglais : *Natural Language Processing*) est une combinaison de motivation théorique de la technologie de calcul des textes naturels à un ou plusieurs niveaux d'analyse linguistique, qui vise à réaliser le traitement linguistique de type humain pour atteindre diverses tâches ou objectifs. Et d'après Liddy, cette définition peut être complétée en séparant les points clés suivants :

— **Motivation théorique de la technologie de calcul :** De nombreuses approches sont disponibles pour effectuer un certain type d'analyse linguistique.
— **Textes naturels :** Les textes traités peuvent être parlés ou écrits tant qu'ils sont des langues utilisées dans l'interaction humaine.
— **Niveaux d'analyse linguistique :** En raison de l'importance de chaque grade dans différentes catégories, chaque TAL utilise un niveau différent d'analyse linguistique ou une combinaison de niveaux.
— **Traitement linguistique de type humain :** Le TAL est une discipline d'intelligence artificielle (IA) dont le but est d'atteindre des performances de type humain.
— **Diverses tâches ou objectifs :** Il y a de nombreuses façons d'accomplir une tâche particulière. La manière la plus efficace dépend de la situation spécifique.

I.3.1 Applications de TAL

IBM Cloud Education a mis en évidence un large éventail d'applications du Traitement Automatique du Langage Naturel qui est à la fois une poursuite théorique et un défi de mise en œuvre [Education, 2020]. Voici quelques applications les plus courantes :

— **Recherche d'Information (RI) :** Conformément à la demande de l'utilisateur, RI retourne une liste de documents potentiellement liés. Cependant, le NLP n'est utilisé que dans quelques cas.
— **Extraction d'Information (EI) :** IE se réfère à la détection, l'étiquetage et la collecte de données dans un format structuré. Ces extraits peuvent ensuite être utilisés pour diverses tâches, telles que les systèmes QA, visualisation et recherche de données.
— **Système de questions-réponses (QAS) :** Ce Système fournit aux utilisateurs la réponse ou les passages qui fournissent la réponse.
— **Récapitulation de Données :** Peuvent permettre l'intégration d'un texte plus long dans une représentation narrative plus interloquée dans le document original, mais toujours construite d'une manière riche.
— **Traduction Automatique :** Cela inclut la traduction d'une langue naturelle à une autre langue tout en conservant le sens et en produisant un texte fluide. Dans ces systèmes, différents niveaux de TAL ont été utilisés.
— **Systèmes de Dialogue :** Les systèmes de dialogue qui utilisent les niveaux phonétique et lexical de la langue se concentrent sur une application étroitement définie. Comme les Chatbots.

I.4 Système de questions-réponses (QAS)

Majid Latifi a décrit les systèmes questions-réponses (QAS) (en Anglais :*Question Answering System*) non seulement comme des systèmes intelligents qui envoient des documents liés à des questions, mais extraient également des informations pertinentes de ces ressources et fournissent des réponses complètes similaires à celles données par les humains [Latifi, 2018]. C'est un vaste sujet de recherche qui relie les domaines de la recherche d'information, de l'intelligence artificiel, et du traitement automatique du langage naturel.

I.4.1 Composants d'un Système Question-Réponse

Chaque système de QA a sa propre architecture et son propre pipeline, en fonction de l'approche qu'il utilise et de la façon de gérer les dimensions de la tâche QA. Cependant, d'après [Hovy et al., 2000],[Jijkoun et al., 2003], [Kolomiyets and Moens, 2011], il existe une architecture sous-jacente qui peut être détectée dans chaque système. Voici les trois phases principales dans l'architecture d'un système de QA traditionnel :

I.4.1.1 Traitement des questions

Selon [Ezzeldin and Shaheen, 2012], ce module de traitement du question (en Anglais : *Question Processing*)concerne : l'analyse, la classification et la reformulation de la question de l'utilisateur. Il traite la question de l'utilisateur, mot à mot et en langage naturel, et analyse le contenu. Cela se fait par les sous-tâches communes suivantes :

- **Classification des Questions :** En classant la question de l'utilisateur dans un type de question, le type de réponse attendu peut être détecté, ce qui permet d'augmenter la précision de la réponse récupérée [Al Chalabi, 2015].
- **Extraction de Mots-clés :** Le processus d'extraction d'une liste de mots-clés à partir d'une question peut être effectué à l'aide de certaines méthodes telles que :
  - Tokénisation : Décomposition de question en un ensemble de mots significatifs, appelés Tokens.
  - Balisage POS : C'est le processus d'attribution d'une partie du discours (POS) (e.g : nom, verbe, adjectif, etc.) sous forme d'une étiquette à chaque mot de la question.
  - Stemming :, Le processus de réduction des mots dérivés de leur forme racine (ou primitive).
- **Extension de Requête :** Ajout de synonymes de mots-clés à la requête d'origine pour améliorer les performances de récupération des passages.

I.4.1.2 Traitement de Documents

[Hu, 2006] a spécifié que le module de traitement de documents (en Anglais : *Document Processing*) permet de récupérer, filtrer et classer les documents dans lesquels des réponses doivent être recherchées. Les informations disponibles pour répondre à la question lisible par le système incluent le type de question de la première phase, ainsi que la fonction de recherche du système. Ce module nécessite des techniques de RI pour atteindre cet objectif. Les documents doivent avoir un modèle de représentation spécifique pour que le système utilise une technique de recherche dédiée. Cette unité peut être divisée en sous-tâches suivantes :

- **Identification des Documents Pertinents :** Pour chercher les documents les plus similaire à la question en se basant sur l'ensemble des mots-clés ou leurs synonymes.
- **Classement des Documents Pertinents :** Les documents reçus sont ensuite

classés selon leur pertinence par rapport aux question, pour obtenir les N premiers documents qui pourront contenir les passages pertinents.

— **Récupération de Passage :** À partir des N premiers documents classés, le système vise les paragraphes les plus pertinents par rapport aux termes de la requête.

I.4.1.3 Traitement de Réponses

Selon [Tellex, 2003] [Niu, 2007], le module de traitement de réponses (en Anglais : *Answer Processing*) est responsable de la reconnaissance, de la récupération et de la confirmation de la réponse à la question. L'entrée se compose des passages qui ont été découverts par le module précédent. Les candidats à la réponse sont extraits des passages en fonction des informations demandées dans la question. Encore une fois, les techniques de RI sont nécessaires, mais cette fois dans le but de localiser les réponses candidates. Parmi ces réponses candidates, une réponse finale doit être choisie, et cela nécessite un classement. Pour trouver la réponse avec la plus grande probabilité de répondre à la question de l'utilisateur. Ce module suit les étapes suivantes :

— **Reconnaissance d'entités Nommées (NER) :** Les réponses candidates sont recherchées en tant que entités nommées. NER (en Anglais : *Named Entity Recognition*)est le processus d'extraction d'entités nommées et de les étiqueter avec une catégorie prédéfinie, telle que le nom, l'organisation, et le numéro [Shaalan, 2014].

— **Notation et Classement des Réponses :** Les réponses candidates récupérées sont notées et classées pour obtenir la réponse la plus pertinente.

— **Présentation de la Réponse :** Pour les questions factoïdes, la réponse peut être présentée sous la forme d'une entité nommée. D'autres types de questions, telles que celles relatives à la définition, peuvent être présentées sous forme de phrase ou de paragraphe.

I.5 Classification des Systèmes QA

Il existe de nombreux systèmes QA différents, c'est pourquoi des critères ont été mis en place pour aider à les classer. Dans cette section, nous discutons les différentes classification des QAS.

I.5.1 Classification Basée sur le Domaine d'application

Certains utilisateurs ont besoin d'informations sur un sujet général, tandis que d'autres utilisateurs ont besoin d'informations spécifiques sur un domaine d'application particulier. Voici les deux différents cas :

— **Les QAS de Domaine Ouvert :** Les systèmes de questions-réponses de domaine ouvert (en Anglais : *Open Domain QAS*) sont pas limités à un domaine spécifique, et peuvent fournir une réponse courte à une question, adressée en langage naturel. Le Web est une bonne source de données pour ce type de QAS, donc on peut considéré que un système QA basé sur le Web est des systèmes de domaine ouvert [Yogish et al., 2016].

— **Les QAS à Domaine Restreint/fermé :** Dans le système questions-réponses à domaine restreint (en Anglais : *Restricted/Closed Domain QAS*) les questions sont liées à un domaine spécifique. Ce système se compose d'un nombre limité de questions spécifiques à un domaine particulier et peut répondre à un nombre limité de questions. En effet, dans les systèmes QA à domaine fermé, la qualité des réponses est élevée. Ces systèmes sont conçus pour obtenir des réponses à partir de données structurées (telles que les ontologies), de données non structurées (texte libre) et de données semi-structurées (telles que du texte annoté XML). L'un des systèmes QA à domaine fermé est Green's BASEBALL, qui donne des réponses aux questions sur les données de Base-ball d'une saison [Kaur and Gupta, 2013].

I.5.2 Classification Basée sur les Types de Question

Une étude de Moldovan et al.[Moldovan et al., 2003] menée sur l'état de l'art des QAS met en évidence que la tâche de générer des réponses aux questions des utilisateurs est directement liée au type de questions posées. Cela signifie que la classification des questions affecte directement les réponses. Les résultats montrent que 36,4% des erreurs sont dues à une mauvaise classification des questions posées dans le QAS.

Les questions sont classés comme factoïdes, non-factoïdes et hybrides. Les questions non factoïdes sont des questions qui relèvent de l'une des catégories suivantes : liste, hypothétique, causale, confirmation, validation et questions complexes. Les questions hybrides sont une combinaison de questions factoïdes et non-factoïdes.

— **Questions de Type Factoïde :** Ces questions sont faciles à répondre et sont

basées sur des faits qui nécessitent des réponses en une phrase ou dans des phrases courtes. Les types de réponse pour les questions de type factoïde sont généralement nommés "entités" [Khillare et al., 2014], il s'agit généralement d'éléments d'information. Ces questions ne nécessitent pas de traitement compliqué du langage naturel pour obtenir des réponses. Par conséquent, leur identification et leur classification est l'un des défis de la recherche dans le système QA.

— **Questions de Type Liste :** Les questions de type liste nécessitent une liste de faits ou d'entités comme réponse. Les techniques qui peuvent être utilisées dans les questions de type factoïde peuvent également bien fonctionner pour les questions de type liste [Wu et al., 2015].

— **Questions de Type Hypothétique :** Les questions hypothétiques demandent des informations sur des événements hypothétiques, sans préciser ce qui en résultera. La fiabilité et la précision de ces questions varient selon les utilisateurs et le contexte. Le type de réponse attendu est réparti ce type de question, et pour cette raison la précision est faible [Mishra and Jain, 2016].

— **Questions Causales :** Les réponses aux questions causales ne sont pas des entités nommées en tant que questions de type factoïde. Ils ont besoin des descriptions d'une entité, qui peuvent aller de phrases à des paragraphes à un document entier [Pechsiri and Piriyakul, 2016].

— **Questions de Validation :** Les questions de validation nécessitent des réponses par oui ou par non. Pour générer des réponses, ce système nécessite un mécanisme d'inférence, une base de connaissances et un raisonnement de bon sens [Tanwar et al., 2014].

— **Questions Complexes :** Les réponses aux questions complexes nécessitent souvent de déduire et de synthétiser des informations à partir de plusieurs documents afin d'obtenir plusieurs éléments d'information comme réponse. Selon [Basuki and Purwarianti, 2016], la question complexe se compose de plusieurs questions qui cherchent des réponses à partir de plusieurs documents.

I.5.3 Classification basée sur les types de sources de données

[Mishra and Jain, 2016] classent les QAS sur ces 3 types de sources de données :

— **Source de données structurée :** Dans les documents structurés, les données sont structurées en ensembles sémantiques (entités). Les entités similaires sont regroupées en relations. Les entités qui sont dans la même relation ont les mêmes attributs. Un schéma est une description de toutes les entités d'une unité. La mise en page des données a un format spécifique. La correspondance de la requête avec la source de données structurées est exacte et le langage de requête correspondant est artificiel.

— **Source de données Semi-Structurée :** Dans une source de données semi-structurée, il n'y a pas de distinction claire entre les données stockées et le schéma qui les définit.

— **Source de Données Non-Structurée :** Les données peuvent être de toute nature. Les données ne sont pas organisées de manière significative. Il n'y a pas de règles définies sur la façon dont les données doivent être organisées dans cette source de données. Les QAS traitent des documents non structurés nécessite l'utilisation de technologies de Traitement du Langage Naturel et de la RI afin de trouver des réponses.

I.5.4 Classification Basée sur les Caractéristiques des Sources de Données

[Mishra and Jain, 2016] ont également classé QAS en fonction des caractéristiques des sources de données, notamment :

— **Taille de la Source :** La tâche de trouver des réponses dans les documents est liée à la taille de la source et au nombre de documents. Les grandes collections de documents présentent à la fois des avantages et des inconvénients lorsqu'il s'agit de trouver les bonnes réponses. Deuxièmement, plus il y a de documents avec des réponses différentes ; plus la justification de l'exactitude de ces réponses est élevée.

— **Langue :** Si le document est multilingue, la tâche de générer une réponse est difficile car différentes langues suivent une syntaxe et des règles différentes. Il n'y a pas de règle unique qui peut être utilisée pour comprendre toutes les langues

naturelles.

— **Hétérogénéité :** De nombreuses informations sont stockées sur différents sites et dans différents formats. Il n'existe pas de modèle unique capable de modéliser différents types de sources de données. En conséquence, les systèmes ont du mal à gérer des sources de données hétérogènes [Mishra and Jain, 2016].

— **Genre :** Le langage utilisé dans les sources de données peut être formel ou informel. Le langage informel est difficile à comprendre pour les systèmes informatiques car il ne suit aucune syntaxe ou règle formelle. Ces sources de données sont difficiles à traiter car un arbre d'analyse incorrect est généré. Par conséquent, la récupération des réponses est une tâche difficile.

— **Media :** La plupart des recherches effectuées dans le domaine de QA consultent des collections de documents textuels. Cependant, la récupération de réponses sous forme de multimédia, c'est-à-dire d'audio, de vidéo et de son, est une tâche difficile. [Indurkhya and Damerau, 2010].

I.6 Évaluation d'un Système QA

[Roberts and Gaizauskas, 2004] considèrent qu'une réponse est correcte lorsqu'il s'agit d'une réponse pertinente à la question (pertinence), et qu'il existe suffisamment de preuves démontrant que le document d'où provient la réponse contient bien l'information pertinente (justification). En conséquence, pour chaque question de l'ensemble de test donné, nous pouvons extraire les métriques suivantes :

a : nombre de résultats pertinents récupérés.

b : nombre de résultats non pertinents récupérés.

c : nombre de résultats pertinents, qui n'ont pas été récupérés.

Selon Roberts [Roberts and Gaizauskas, 2004] deux des mesures de base que nous pouvons calculer sur la base des métriques a, b et c sont :

— **Précision :** est le nombre de passages/réponses récupérés qui sont effectivement pertinents à la requête ou à la notion mathématique : I.1

$$Prcision = \frac{a}{a+b} \tag{I.1}$$

— **Rappel :** renvoie la mesure dans laquelle tous les passages/réponses pertinents sont récupérés : I.2

$$r = \frac{a}{a+c} \tag{I.2}$$

Il existe de nombreuses métriques différentes pour évaluer les ensembles de données Question-Réponse, y compris la "correspondance exacte" (*Exact Matching (EM)* en Anglais), K-Précision et le score "F1". Ces scores sont basés sur les paires de questions et de réponses qui ont été données. Le score maximum possible sur une question est le total des scores de toutes les réponses correctes.

— **F-mesure :** (en Anglais : *F-measure*) Le score F1 est une mesure courante utilisée dans les problèmes de classification et est largement utilisé en QA. Il convient que nous prenions également soin de l'exactitude et du rappel. Dans ce cas, la prédiction est basée sur les mots individuels de la prédiction, par rapport aux mots de la vraie réponse. Le score F1, ou précision, est basé sur le rapport du nombre de mots répartis entre la prédiction et la vérité. Le rappel est basé sur le rapport du nombre de mots répartis entre la prédiction et la vérité. [**?**]
I.3

$$F1 = 2\frac{recall * precision}{precision + recall} \tag{I.3}$$

— **Correspondance Exacte :** La métrique de la correspondance exacte est simple (en Anglais : *Exact-match*). Il suffit de compter le nombre de mots dans chaque paragraphe. Pour chaque paire question/réponse, si la prédiction du modèle correspond à l'une des vraies réponses, le modèle est considéré comme ayant réalisé 1 point de précision ; sinon, la précision du modèle est considérée comme nulle. Un changement de personnage unique compte comme une perte, avec un score de 0. Lors de l'évaluation par rapport à un exemple négatif, si le modèle prédit du texte, il reçoit automatiquement un 0 pour cet exemple [Labs, 2020].

— **Gain Cumulé Actualisé Normalisé :** (en Anglais : *Normalized Discounted Cumulative Gain*) mesure l'utilité ou le gain d'un passage/réponse en fonction de sa position dans la liste des résultats. Une réponse très pertinente qui se trouve en haut de la liste des résultats obtient un score plus élevé tandis qu'une réponse très pertinente qui se trouve plus bas dans la liste est pénalisée [Wang et al., 2007].

— **K-Precision :** Renvoie le nombre de passages/réponses pertinents parmi les k premiers passages/réponses récupéré [Foundation, 2022].

I.7 Les Systèmes Comparatifs de questions-réponses

Une recherche effectuée sur la compréhension des Systèmes Comparatifs de questions-réponses impliquait que la comparaison des possibilités alternatives est une technique courante pour faire des sélections éclairées [Bondarenko et al., 2022]. Cet article cite deux études, dont la première a révélé que 80% des Américains préféreraient rechercher des décisions importantes, telles que location ou de l'achat en ligne plutôt que de consulter leurs proches [Turner and Rainie, 2020]. La deuxième étude révélé qu'au moins 3% des questions soumises aux moteurs de recherche sont des questions comparatives, qui demandent des comparaisons factuelles ainsi que des opinions et des arguments [Bondarenko et al., 2020].

Quelques questions comparatives peuvent être résolues directement à partir d'une base de connaissances, tandis que d'autres nécessitent des preuves combinées de différents passages de texte. Le moteur de recherche doit s'assurer que la question d'entrée est réellement comparative. Afin de mieux comprendre et répondre à ces questions, diverses classifications ont été développées, entraînées et évaluées.

I.7.1 Classifications dans les QAS Comparatifs

Lorsqu'un moteur de recherche doit décider de basculer ou non la présentation des réponses vers une interface de comparaison, il doit s'assurer que la question posée est bien comparative. Ainsi, la classification et l'identification des questions comparatives sont essentielles pour trouver une réponse dans des QAS comparative.

Une étude de Bondarenko [Bondarenko et al., 2020] a été menée pour analyser les questions comparatives des journaux Yandex pendant un l'année 2012. Dans la recherche plusieurs techniques de classification différentes ont été combinées, dont l'objectif est d'atteindre une précision parfaite avec le rappel le plus élevé possible en utilisant les règles lexicales et syntaxiques de la classification.

— **Classification Basée sur des Règles :** Le système utilise 10 règles soigneusement élaborées pour répondre à environ 80% des questions étiquetées comme telles. Une question classée comme comparative peut être comparée à d'autres questions. Nous n'incluons pas toutes les règles ici ; mais par exemple, la règle 1 de l'ensemble classe une question comme comparative si elle contient un adjectif ou un adverbe comparatif.

— **Classification Basée sur les Fonctionnalités :** Pour augmenter encore le rappel, les classificateurs basés sur les caractéristiques et appliqués après les règles : régression logistique, naïve Bayes, SVM et forêts aléatoires. Les classificateurs sont formés et affinés dans une validation croisée à 10 reprises sur les questions de l'ensemble de données complet que les règles ne classent pas comme comparatives. comme comparatives. Ces classificateurs sont destinés à identifier les questions comparatives les plus "difficiles".

— **Classification Neuronale :** Un classificateur neuronal sur les questions non classées comme comparatives par les règles ou celles restant après la régression logistique. Un réseau neuronal profond (DNN) a été utilisé en combinaison avec la régression logistique pour obtenir de meilleurs résultats que toute configuration de régression logistique réalisée précédemment.

I.7.2 Classifications des Termes des Questions Comparatifs

Pour mieux comprendre une question comparative, [Bondarenko et al., 2022] a développé des classifieurs qui identifient les termes importants.

— **Classification Multi-Classes des Tokens :** Identifie les aspects des questions et classe ces termes, y compris l'intérêt et les prédicats, dans l'une des trois catégories (Objets, aspects et prédicats) ou plus.

— **La classification des Tokens par Classe :** Elle vise à distinguer les questions directes des questions indirectes, ou uniquement les étiquettes d'aspect pour les questions qui contiennent réellement des aspects, seraient plus efficaces.

I.7.3 Les Défis des Systèmes QA Comparatives

Une recherche effectuée par Lim et al. [Lim et al., 2009] a identifié les principaux défis pour le développement d'un QAS comparative et évaluative. Voici les défis qui ont été présentés :

I.7.3.1 L'Analyse des questions et la semantique des comparatifs

L'analyseur de questions doit identifier les expressions comparatives dans la question et les décomposer en composants significatifs, notamment :

— **Identifier le type de comparaison :** Le type de comparaison peut être déterminé par les insignes et les propriétés des objets par : les limitations de la capacité de se déplacer par rapport aux propriétés au sein du même objet, le degré de comparaison entre la même propriété entre différents objets, ou différentes caractéristiques comme volonté.

— **Détermination du sens sémantique et conversion en mesures quantifiables :** Les propriétés impliquées dans la comparaison sont intégrées dans la sémantique des mots interrogatifs et éventuellement dans le contexte qui accompagne la question.

— **Déterminer des limites, des plages et des valeurs relatives selon l'objet :** C'est un grand défi pour les questions de comparaison. Les propriétés de prédicat peuvent être non spécifiées ou polysémiques et ne gagneraient en contexte que lorsqu'elles seraient associées à l'objet. Il est nécessaire de définir des échelles comparatives le long des propriétés de base afin que ces valeurs soient ordonnées.

— **Traitement des superlatifs et autres formes de quantification liées aux comparaisons :** L'exploration de l'assouplissement de l'évaluation de plusieurs propriétés avant de déterminer les meilleurs résultats et d'évaluer le superlatif de chacune des propriétés afin d'identifier les propriétés que l'objet n'a pas rencontrées.

I.7.3.2 Détermination de la Réponse

Seule une évaluation correcte aura lieu lorsque le prédicat est décomposé en propriétés. Premièrement, le défi principal est de transformer les concepts de la requête en ceux du schéma conceptuel de la base de données. Ensuite, il faut rechercher sur le Web des données pertinentes, extraire des mots-clés de la question puis récupérer les résultats du moteur de recherche à travers celle-ci, via des grammaires locales. Associées à des propriétés, des valeurs pertinentes peuvent être extraites.

I.7.3.3 Génération de Réponse

Un composant générateur de réponses devrait faire partie d'un système QA comparatif et évaluatif puisque la réponse n'est pas extraite du texte source mais générée à partir de comparaisons numériques et textuelles des critères. Il est donc nécessaire de passer par la génération de phrases élaborées, impliquant des expressions comparatives. Dans le cas

où la réponse n'est pas directe, il est également nécessaire de mettre au point des formes adaptées de coopérativité, en fournissant à l'utilisateur des informations appropriées sous forme d'explications, d'élaborations, d'exemples (de propriétés) et d'autres informations pertinentes.

I.8 Les Systèmes QA en langue Arabe

I.8.1 La langue Arabe

De nombreux points de vue, la langue Arabe a été proéminente. Au début, elle était l'une des langues les plus utilisées au cours du «*Golden Age*», en particulier dans les domaines des mathématiques, de la médecine, de l'astronomie et de la chimie. En outre, la langue Arabe a une signification religieuse importante dans l'Islam car le Coran, l'un des quatre livres sacrés, a été révélé en Arabe. Ce qui signifie que plus de 1,2 milliard de musulmans prient quotidiennement en Arabe. Enfin, la langue Arabe est une langue officielle dans 25 pays, avec plus de 300 millions de personnes qui la parlent [Cheddadi, 2014]. Par conséquent, nous pouvons affirmer avec confiance que les facteurs ci-dessus contribuent à expliquer pourquoi cette langue intéresse la communauté du TALN.

I.8.2 Challenges de la langue Arabe

La langue Arabe a une syntaxe complexe avec des caractéristiques uniques difficiles à gérer par un ordinateur. Cette abondance a conduit à un certain nombre de défis qui ont dû être traités différemment par les chercheurs. Cette section traite certains de ces défis :

— **Non-existence de Majuscules :** La reconnaissance d'entités nommées est le processus principal pour répondre aux questions factoïdes et est utilisée pour détecter la reconnaissance d'entités nommées telles que les noms propres, les lieux et les noms de personnes [Shaalan, 2014]. Ce processus est beaucoup plus facile dans les langues latines, dont les entités nommées sont toujours en majuscules, qu'en Arabe, qui n'a pas de majuscules et ne peut pas utiliser les majuscules comme caractéristique orthographique pour distinguer les noms propres Arabes des autres formes de mots telles que les adjectifs et les noms communs [Al-Shargi and Rambow, 2015], [Bakari et al., 2016].

— **Voyelles Facultatives Courtes :** Les voyelles courtes sont écrites avec des signes diacritiques en Arabe. Ces signes diacritiques ont un impact significatif sur le sens

du mot Arabe et sur sa forme orthographique. La version Arabe moderne est disponible sans signes diacritiques et peut être comprise par les Arabes, mais elle peut provoquer une ambiguïté orthographique [Al-Shargi and Rambow, 2015].

— **L'Ordre Libre des Mots :** La langue Arabe a une façon particulière d'ordonner les phrases ayant le même sens. Par exemple, une phrase peut changer de sujet-verbe-objet à verbe-sujet-objet. Cela conduit à d'autres implications qui nécessitent une analyse plus approfondie de la question [Al-Shargi and Rambow, 2015], [Ezzeldin and Shaheen, 2012].

— **Nature Dérivationnelle :** la langue Arabe est une langue fortement fléchie dont le vocabulaire est construit à partir d'un petit nombre de racines fondamentales. Chaque racine se compose généralement de trois à quatre lettres. Chaque racine peut être ajoutée avec un préfixe, un infixe ou un suffixe pour former des mots. Les affixes sont ajoutés à la racine selon 120 modèles. Lemme = racine + modèle, et un mot = racine + affixes (préfixe, infixe ou suffixe). La grande variation de tout cela entraîne une morphologie systématique mais compliquée. Cela conduit à des données clairsemées qui nécessitent un pré-traitement ou un grand ensemble de formation [Ezzeldin and Shaheen, 2012].

— **Manque de Ressources Linguistiques :** Il existe un besoin de corpus avec différents types de questions et de documents, afin d'entraîner et de tester les systèmes de questions-réponses. Ces corpus doivent être reconnus par la communauté des chercheurs. Il existe peu de ressources linguistiques disponibles pour la langue Arabe, ce qui signifie qu'il n'en est encore qu'à ses balbutiements. Ainsi, les chercheurs créent toujours leurs propres corpus, qui doivent être annotés et vérifiés manuellement [Al-Shargi and Rambow, 2015].

Il est également important de noter que les ressources en Arabe sont limitées, en particulier celles qui sont librement disponibles et utilisées comme référence. En plus des ressources, les outils Arabes sont également limités et la plupart d'entre eux ne sont pas disponibles gratuitement.

I.9 Conclusion

Ce chapitre a fourni une explication des systèmes QA, en particulier le QA Arabe. La présentation a commencé par définir les divers domaines combinés de la recherche d'in-

formations et du Traitement Automatique du Langage Naturel utilisés dans les systèmes QA et par la présentation de leurs approches et diverses classifications.

Nous avons également examiné en détail les systèmes QA comparatifs, mettant en évidence les concepts clés. À la fin, nous avons décrit certains des défis de la langue Arabe et leur impact qui devraient être pris en compte lors de la création des systèmes QA Arabe.

Le chapitre suivant, nous décrirons et discuterons brièvement les systèmes QA développés pour la langue Arabe. Nous présenterons les différents ensembles de données Arabes et analyserons les questions et les réponses qu'ils contiennent. À la fin, nous citerons également quelques systèmes comparatifs développés.

Chapitre II

ETAT DE L'ART

II.1 Introduction

La tâche de questions-réponses en langue Arabe n'a pas été largement étudiée au cours de la dernière décennie par rapport aux autres langues. Le but de ce chapitre est d'une part de fournir l'état de l'art actuel sur les QAS développés pour la langue Arabe toute en introduisant les différents datasets utilisés. D'autre part, nous approfondissons notre recherche sur les systèmes QA comparatifs proposés, en décrivant leurs approches et techniques utilisées pour traiter ce type de questions.

II.2 Datasets des Systèmes de Questions-Réponses Arabe

La langue Arabe est l'une des langues les plus parlées dans le monde, principalement au Moyen-Orient et en Afrique du Nord. Malgré le grand nombre de locuteurs, la recherche sur le NLP Arabe est encore entravée par un manque de ressources linguistiques telles que des corpus, des lexiques, des API et des outils d'analyse. Quelques ensembles de données en Arabe ont été proposés pour le système QA.

Dans la partie suivante, Nous allons étudié les ensembles de données existants dans des enquêtes récemment publiées [Alwaneen et al., 2021], [Biltawi et al., 2021]. Nous mettons en évidence leur structure, leur source de données et le type de questions ciblées.

— **l'ARCD (Arabic Reading Comprehension Dataset) :** Présenté par Mozannar [Mozannar et al., 2019], qui se compose de 1 395 questions factoïdes rassemblés à partir d'articles de Wikipédia par des crowdworkers. L'auteur a présenté **Arabic-SQuAD** dans le même papier. Il s'agit d'une traduction automatique de SQuAD 1.1 (Stanford Question Answering Dataset) [Rajpurkar et al., 2016] dont il y a 48 000 tuples de paragraphe-question-réponse .

— **TALAA-AFAQ :** Un acronyme proposé par Aouichat et Guessoum [Aouichat and Guessoum, 2017] pour un corpus composé de 2002 paires question-réponse qui ont été triées manuellement en quatre catégories. Il y a 688 paires de questions-réponses collectées à partir de divers sources, y compris les questions CLEF et TREC en Arabe, Internet et divers textes où les patterns de réponse sont connus.

— **DAWQAS :** Un dataset de QA Arabe introduit par Ismail et Homsi [Ismail and Homsi, 2018]. Dont 3205 paires question-réponse ont été extraites de sites Web publics en Arabe.

— **AQAD :** Fourni par Atef [Atef et al., 2020], un dataset qui contient plus de 17 000 questions et réponses. Les passages de l'ensemble de données ont été collectés à partir d'articles de Wikipédia en Arabe et qui correspondent à ceux utilisés dans l'ensemble de données SQuAD 2.0 [Rajpurkar et al., 2018].

Les ensembles de données susmentionnés sont tous monolingues ; ils contiennent des paragraphes, des questions et des réponses uniquement en langue arabe. Les ensembles de données suivants sont multilingues, y compris une partie en langue Arabe

— **TyDiQA :** Un dataset par Clark [Clark et al., 2020] qui couvre 11 variétés de langues différentes dont il exist 204 000 paires question-réponse. Les données sont collectées directement dans chaque langue sans utiliser de technologies de traduction, contrairement à WQuAD.

— **MKQA :** Un quiz de connaissances multilingue avec 10 000 paires de questions et réponses en 26 langues lancé par Longpre [Longpre et al., 2020]. Les questions ont été extraites de la l'ensemble de données *Google Natural Questions* [Kwiatkowski et al., 2019].

— **MLQA :** Une référence multilingue pour l'évaluation des QAS Extractive.Le jeu de données contient 12 000 instances en anglais et 5 000 dans chacune des autres langues.

— **XQuAD :** est un ensemble de données de référence pour évaluer les performances des QAS multilingues. Il a été publié par Artetxe et al. [Artetxe et al., 2019]. L'ensemble de données comprend 240 paragraphes et 1190 paires de questions-réponses traduites à partir de l'ensemble de données SQuAD 1.1 en dix langues par un outil de traduction professionnel.

Le tableau II.1 ci-dessous résume les ensembles de données discutés précédemment. Dans cette analyse, nous avons exclu les ensembles de données comportant moins de 1000 questions .

Dataset	Référence	Année	Taille	Types de Question	Monolingue/ Multilingue	Traduite de
TALAA-AFAQ	[Aouichat and Guessoum, 2017]	2017	2002	Factoid questions	Monolingual	/
DAWQAS	[Ismail and Homsi, 2018]	2018	3205	Non Factoid (why questions)	Monolingual	/
ARCD	[Mozannar et al., 2019]	2019	1395	Factoid questions	Monolingual	/
Arabic-SQuAD	[Mozannar et al., 2019]	2019	48,344	Factoid questions	Monolingual	SQuAD 1.1
AQAD	[Atef et al., 2020]	2020	17911	Factoid questions	Monolingual	SQuAD 2.0
TyDiQA	[Clark et al., 2020]	2020	15,421	Factoid questions	Multilingual	/
XQuAD	[Artetxe et al., 2019]	2020	1190	Factoid questions	Multilingual	SQuAD 1.1
MLQA	[Lewis et al., 2019]	2020	5000	Factoid questions	Multilingual	/
MKQA	[Longpre et al., 2020]	2021	10,000	Factoid questions	Multilingual	GNQD

Table II.1 — Datasets des Systèmes de Question-Réponse Arabe.

II.3 Systèmes de Questions-Réponse Arabe

Dans la section suivante, nous étudierons les systèmes de Questions-Réponses développés pour la langue Arabe. Cependant, ces systèmes sont limités en terme de type des questions traitées et domaine d'application. Nous citons des exemples des premiers systèmes développés ainsi que d'autres plus récents avec des techniques plus avancées.

— **AQAS :** Le premier système de questions-réponses pour la langue Arabe [Mohammed et al., 1993]. Un système de QA qui extrait les réponses uniquement à partir de données structurées ; à ce jour, aucun résultat d'évaluation n'a été publié pour le système AQAS.

— **QARAB :** Le deuxième système QA pour la langue Arabe proposé en 2002 a été développé par Hammo [Hammo et al., 2002]. Le logiciel utilise des techniques de recherche d'informations (IR) et de traitement du langage naturel (NLP) pour extraire des informations. Ce système a atteint une précision de 97,3% et également un rappel de 97,3%. L'évaluation a été réalisée par quatre arabophones natifs qui ont présenté 113 questions au système et ont jugé eux-mêmes l'exactitude des réponses.

— **ArabiQA :** Benajiba a proposé un système pour répondre aux questions factoïdes [Benajiba et al., 2007]. Il a utilisé le *Java Information Retrieval System (JIRS)*, un système de récupération de passage, pour trouver les passages pertinents. Il a utilisé également un système Arabe de reconnaissance d'entités nommées (NER) appelé ANERsys pour identifier et classer les entités nommées dans les passages récupérés.. L'ensemble de tests comprend 200 questions et 11 000 documents de la version Arabe de Wikipedia. Ils ont atteint une précision de 83.3%.

— **QASAL :** [Brini et al., 2009] Un système qui se compose de trois parties. Dans la première étape, les questions sont analysées pour identifier le type de réponse, les mots-clés et le sujet. Dans le second, les passages pertinents sont récupérés. Dans le dernier, les réponses étaient affichées. Ce système est mis en œuvre à l'aide d'un moteur linguistique qui comprend des dictionnaires et des grammaires à large couverture, et une analyse de corpus en temps réel.

— **DefArabicQA :** introduit par Trigu en 2010 [Trigui et al., 2010]. Ce système se compose de quatre composants principaux : analyse des questions, récupération des passages, extraction des définitions et classification des définitions. Deux expériences ont été menées, et les résultats ont été évalués manuellement par un arabophone natif. Le système a donné cinq réponses. La première expérience a été menée en utilisant le moteur de recherche Google en tant que ressource Web, et la deuxième expérience impliquait d'utiliser le moteur de recherche Google et Wikipedia Arabe en tant que ressources. La deuxième expérience a été plus réussie que la première.

— **QArabPro :** Ce système gère toutes sortes de questions, y compris (comment et pourquoi) [Akour et al., 2011]. Bien que la précision globale de ces deux types de questions soit faible, 62% pour pourquoi et 69% pour combien par rapport aux autres types de questions, cela a été considéré comme une étape importante et une amélioration par rapport à la version actuelle des questions et réponses pour la langue Arabe. Les modules de Reconnaissance d'entités nommées (en Anglais : *Named Entity Recognizers (NER)*) du système sont essentiels à l'expansion des requêtes et à l'extraction des mots-clés pertinents, et constituent un élément crucial de tout lexique linguistique. L'acquisition automatique de connaissances sémantiques devrait améliorer les performances du système dans son ensemble. Plus précisément, l'amélioration des performances du système sera utile pour répondre aux questions "comment et pourquoi".

— **IDRAAQ :** Ce système est basé sur des niveaux et des structures basés sur des mots-clés qui consistent en un processus d'expansion de requête basé sur des relations sémantiques dans l'ontologie **WordNet** Arabes et un système de récupération de passage basé sur un modèle de densité de N-distance [Abouenour et al., 2012].

L'analyse des performances du système IDRAAQ a permis d'identifier les questions pour lesquelles le système n'est pas en mesure de valider la bonne réponse. Les expériences réalisées l'ont montré en ce qui concerne la précision et la mise à l'échelle c@1.

— **ALQASIM :** Un système qui se concentre sur la sélection et la validation des réponses [Ezzeldin and Shaheen, 2012]. Cette étude expérimentale a été menée en conjonction avec la tâche principale de QA4MRE@CLEF 2013. ALQASIM utilise une nouvelle technique qui analyse la lecture des documents de test au lieu des questions, ce qui conduit à une performance prometteuse de 0,31 précision et 0,36 C@1, sans utiliser les collections de fond de l'ensemble de test.

— **AQuASys :** Un système QA Arabe autonome pour les questions factoïdes [Bekhti and Al-Harbi, 2013]. Il utilise largement les techniques NLP pour analyser les questions et récupérer les réponses à partir d'un corpus Arabe développé par les auteurs. Les réponses ont été notées et présentées selon leur pertinence.

— **Al-Bayan :** Un système QA spécialisé pour le Coran développé par abdelnasser et al. [Abdelnasser et al., 2014]. Le système peut répondre à une question en Arabe sur le Coran, en récupérant les versets les plus pertinents du Coran et de ses livres d'interprétation (Tafsir). Les résultats d'évaluation sur un ensemble de données collectées montrent que le système global peut atteindre une précision de 85% en utilisant les 3 premiers résultats.

— **JAWEB :** Une application Web facilement accessible faite par Kurdi [Kurdi et al., 2014]. Il a été démontré que l'application donne des réponses correctes avec un rappel de 100% et une précision moyenne de 80%. Par rapport au système de QA basé sur le Web appelé *ask.com*, JAWEB a fourni un meilleur taux de rappel. Ces résultats prometteurs prouvent clairement que JAWEB a un grand potentiel en tant que plateforme de questions-réponses et est essentielle pour les internautes arabophones du monde entier.

— **NArQAS :** un QAS conçu par bakari et al. [Bakari et al., 2014]. Une approche qui permet d'analyser un texte donné, dans un domaine ouvert, et de générer des représentations logiques. Cette démarche s'appuie sur les prescriptions techniques

du RTE. Évidemment, les données reposent sur une transformation logique afin de répondre à la question. Ensuite, l'objectif général est de fournir une reconnaissance des implications textuelles entre les prédicats logiques. En conséquence, ceux qui ont des implications sont considérés plus tard dans les réponses du candidat à la question de l'utilisateur Dans tous les cas, l'un de ces candidats est la bonne réponse.

— **EWAQ :** Un système qui vise à améliorer la classification des passages pertinents récupérés par les moteurs de recherche en fonction du degré de similitude d'implication entre les questions pourquoi et le passage récupéré [Al-Khawaldeh, 2019]. EWAS teste un ensemble de 250 questions pourquoi et obtient leurs réponses correctes manuellement. Il a atteint une précision de 66,19% pour Google, 63,27% pour ASK et la précision minimale est de 61,48% pour Yahoo.

— **IQAS :** Ce système est conçu pour traiter des informations temporelles impliquant plusieurs formes d'inférence en Arabe [Neji et al., 2016]. Une méthodologie pour calculer l'inférence temporelle qui peut aider à améliorer la reconnaissance des réponses exactes aux questions sur le temps.

— **AlQuAnS :** Ce système intègre de nombreux algorithmes précédemment couronnés de succès et utilise une nouvelle approche pour l'extraction des réponses qui n'a été utilisée par aucun système QA Arabe auparavant. Le module d'extraction de réponses utilise un système NER dont il a été démontré qu'il produit de meilleurs résultats. Les résultats montrent qu'avec plus de données, cela produirait beaucoup plus de types de modèles, ce qui fournirait une sortie acceptable.

— **SOQAL :** Un système de Questions-Réponses en arabe pour les questions factuelles du domaine ouvert utilisant Wikipédia comme source de connaissances [Mozannar et al., 2019]. L'approche de SOQAL consistait en un récupérateur de documents utilisant TF-IDF hiérarchique et un lecteur de documents utilisant BERT.e système a obtenu un score F1 de 61,3 et une correspondance des phrases de 90,0% sur l'ensemble de données de compréhension de lecture en arabe (ARCD) et un score F1 de 27,6 sur une version à domaine ouvert de l'ARCD. Il a également démontré l'efficacité de l'utilisation de données traduites comme ressource de training pour QA.

— **ASHLAK :** Ce système présente une méthode de question réponse à appliquer sur le Hadith afin de fournir une réponse informative correspondant à la requête de l'utilisateur [Abdelnasser et al., 2014]. Il utilise trois méthodes principales : (1) Éviter d'extraire un passage dont la similarité avec la requête est élevée mais dont le sens est différent. (2) Calculer la similarité sémantique et syntaxique de la phrase à la phrase et de la phrase à la requête. (3) Développer des mots à la fois dans la requête et dans les phrases pour résoudre le problème fondamental de l'inadéquation des termes entre les phrases et la requête de l'utilisateur. Les résultats expérimentaux montrent que la méthode proposée est capable d'améliorer les performances par rapport aux méthodes existantes sur les jeux de données Hadith.

Le tableau II.2 résume les systèmes QA de la langue Arabe discutés précédemment et leur classification.

Le Système QA	Année	Référence	Domaine	Types de Question	Source de Données	Résultats
QARAB	2002	[Hammo et al., 2002]	Open	Factoïde	Non-Structuré	**R : 97.3** **P : 97.3** **MRR : 86**
ArabiQA	2007	[Benajiba et al., 2007]	Open	Factoïde	Non-Structuré	**P : 83**
QASAL	2009	[Brini et al., 2009]	NA	Factoïde	Non-Structuré	NA
DefArabicQA	2010	[Trigui et al., 2010]	Open	Non-Factoïde	Non-Structuré	**MRR : 81** **AQ : 64**
QArabPro	2011	[Akour et al., 2011]	Open	Hybride	Non-Structuré	**R : 86** **P : 93** **F1 : 89**
IDRAAQ	2012	[Abouenour et al., 2012]	Restreint	Hybride	Non-Structuré	**A : 13** **C@1 : 21**
ALQASIM	2013	[Ezzeldin et al., 2013]	Restreint	Hybride	Non-Structuré	**A : 31** **C@1 : 36**
AQuASys	2013	[Bekhti and Al-Harbi, 2013]	NA	Factoïde	NA	**R : 97.5** **P : 66.25** **F1 : 78.89**
Al-Bayan	2014	[Abdelnasser et al., 2014]	Restreint	Factoïde	Non-Structuré	**A : 85.4**
JAWEB	2014	[Kurdi et al., 2014]	Open	Factoïde	Non-Structuré	**R : 100** **P : 80**
NArQAS	2014	[Bakari et al., 2014]	Open	Factoïde	Non-Structuré	**A : 89** **C@1 : 89**
EWAQ	2015	[Al-Khawaldeh, 2019]	Restreint	Non-Factoïde	Non-Structuré	**A : 68.53**
IQAS	2016	[Neji et al., 2016]	Open	Factoïde	Non-Structuré	NA
AlQuAnS	2017	[Nabil et al., 2017]	Open	Factoïde	Non-Structuré	**A : 22.20** **MRR : 8.16** **AQ : 47.66**
SOQAL	2019	[Ahmed et al., 2017]	Open	Factoïde	Non-Structuré	**F1 : 42.5** **EM : 20.7**
ASHLAK	2020	[Abdi et al., 2020]	Restreint	Non-Factoïde	Non-Structuré	**R : 63.87** **P : 83.47** **F1 : 72.37**

Table II.2 — Classification des systèmes QA Arabe.

Selon une enquête sur les enjeux des systèmes QA en langue Arabe [Ezzeldin and Shaheen, 2012], la majorité des recherches sur le sujet se concentrent sur la réponse aux questions du domaine ouvert, alors que relativement peu de tentatives ont été faites pour aborder la réponse aux questions du domaine restreint. Ils ont souligné la nécessité d'accorder une importance accrue à la sémantique dans les QAS Arabe, où la majorité des recherches se sont concentrées sur les questions morphologiques et syntaxiques. En outre, le raisonnement basé sur l'ontologie peut aider à interpréter les

questions humaines, à identifier la terminologie essentielle et les liens entre elles, et à déterminer les réponses pertinentes en cas de désaccord. Sur la base de leurs recherches, ils sont arrivés à la conclusion que l'utilisation de la méthode syntaxique rend difficile l'obtention de ces avantages.

II.4 Les Systèmes QA Comparatifs

Répondre à des questions comparatives nécessite un examen attentif de phrases similaires, ainsi qu'un traitement sophistiqué. Cette section présente les résultats de recherche sur les QAS comparatifs ainsi que leurs techniques utilisées.

— **Un classificateur de Questions Comparatives soumises à Yandex** Dans l'étude menée par Bondarenko et al. [Bondarenko et al., 2020], ils ont annoté un échantillon aléatoire de 50 000 questions. 20,8% sont comparatifs, et ils ont développé un classificateur axé sur la précision en combinant des règles lexicales et grammaticales artisanales avec des méthodes basées sur les caractéristiques et neurales. Cette méthode atteint un rappel de 0,6 avec une précision idéale de 1,0. Pour cette étude, trois techniques de classification différentes ont été combinées dans un classifieur d'ensemble : les règles lexico-syntaxiques artisanales, les classifieurs traditionnels basés sur les caractéristiques et les réseaux de neurones. Ensuite, un modèle neuronal (CNN [Zeng et al., 2015], LSTM avec abandon itératif [Gal and Ghahramani, 2016], réseau de capsules avec routage dynamique [Sabour et al., 2017] et BERT [Devlin et al., 2018]). Ces questions ont été formées et testées pour avoir un taux de précision supérieur à l'ensemble de règles de précision parfaites. Cependant, seule environ la moitié des questions non factoïdes sur la plateforme de questions-réponses de la communauté russe *Otvety* peuvent recevoir une réponse correcte. Ainsi, se fier uniquement au graphe de connaissances du moteur et aux plateformes de questions-réponses en ligne ne suffira pas.

— **Système QA Comparatives Non-Factoïdes :** Cette étude qui analyse les questions et réponses comparatives [Bondarenko et al., 2022] suit l'idée que Bondarenko [Bondarenko et al., 2020] a appliquée aux questions russes pour l'identification des questions comparatives, pour développer des approches très précises pour :
  — Distinguer les questions comparatives des autres questions.
  — Classer les questions comparatives en questions factuelles et subjectives.
  — Reconnaître les objets, les aspects et les prédicats de comparaison d'une question, et pour les questions comparatives non factuelles.

— Détecter la position dans d'éventuels fragments de texte de réponse envers les objets de la question.

Ils analysé des questions comparatives et leurs réponses à partir de 31 000 ensembles de données de questions, en les marquant comme comparatives ou non. Les 3 500 questions annotées comme comparatives sont en outre étiquetées avec des comparaisons (objets, aspects et prédicats), ce qui les rend plus intelligibles. Pour 950 questions, les réponses du forum en ligne ont été collectées, puis le poste a été étiqueté en fonction d'objets de comparaison communs.

Les trois classifieurs mentionnés précédemment dans l'étude [Bondarenko et al., 2020] de ont été combinés en cascade pour identifier quelles questions sont directes ou indirectes. Ils Identifient les objets de comparaison, les aspects, les prédicats, et décide si une question comparative est factuelle ou demande des opinions/arguments.

Pour augmenter encore le rappel une étape été ajouté. Cette étape reçois sous forme de requêtes d'entrée qui ne sont pas définies comme comparaison après le classifieur neuronal. Faites ensuite la moyenne des probabilités de décision des classificateurs basés sur la régression logistique et l'intégration et validez un autre seuil de décision 10 fois pour rappeler d'autres questions comparatives avec une précision parfaite. Avec cette dernière étape, l'ensemble de la cascade atteint une récupération totale de 0,71 avec une précision toujours optimale de 1,0.

— **Système QA Comparatif (CoQAS) :** [Chekalina et al., 2021] ont développé un système qui aide l'utilisateur à faire un choix approprié en décrivant de manière complète et raisonnable les avantages et les inconvénients possibles de chacune des options d'appariement. Le système CoQAS en Anglais : *Comparative Question Answering System*) définit des structures qui peuvent être utilisées pour comparer des objets, des aspects de comparaisons et des prédicats. La comparaison entre objets est souvent représentée par un adjectif ou un adverbe comparatif. Les aspects de comparaison sont des propriétés partagées selon lesquelles les deux objets sont comparés.

Ce système est basé sur le système d'extraction d'arguments comparatifs (CAM) en Anglais : *Comparative Argument Mining*) [Schildwächter et al., 2019] qui récupère les arguments pour/contre pour un couple d'objets comparés. Le processus est le suivant :

— CAM récupère les phrases contenant les deux objets comparés et l'aspect de comparaison à partir du corpus basé sur Common Crawl. Cela comprend 14,3

milliards de phrases.

— Le CAM classe les phrases comme comparant deux choses, et il identifie la phrase avec la valeur la plus élevée. De plus, il extrait des aspects et des prédicats des phrases comparatives récupérées.
— Enfin, CAM produit une liste de phrases argumentatives pour et contre, et montre le "gagnant" de la comparaison ainsi que les aspects de la comparaison.

Pour donner une réponse plus concise et cohérente sur les informations récupérées, le CAM peut traiter des questions en langage naturel et générer des réponses de type humain. Ce module reconnaît les objets et les aspects dans la demande de texte d'un utilisateur et les met en évidence dans la réponse [Schildwächter et al., 2019].

II.5 Conclusion

Ce chapitre résume et organise les résultats des recherches récentes d'une manière qui intègre et enrichit la compréhension de l'étude des QAS en Arabe. Il s'est concentré sur la classification des systèmes existants tout en développant une perspective sur ce domaine et son évaluation des tendances. Cependant, comme il est impossible d'inclure toutes les recherches antérieures ou même la plupart, ce chapitre n'a inclus que les travaux des auteurs les plus cités dans le domaine de QA Arabe.

En outre, nous avons étudié en profondeur les systèmes QA comparatifs en examinant quelques travaux et en nous concentrant sur leur ensemble d'approches et sur la manière dont ils fonctionnent ensemble pour comprendre les QAS comparatif. Ce qui nous amène à la prochaine étape de notre projet, dans laquelle nous expliquerons notre proposition d'architecture de système de questions-réponses comparatif propre à la langue Arabe.

Chapitre III

CONCEPTION D'UN SYSTÈME DE QUESTIONS-RÉPONSES COMPARATIF EN LANGUE ARABE

III.1 Introduction

Les deux chapitres précédents ont servi de point de départ à notre recherche, en nous permettant de voir l'état actuel des QAS Arabes et des QAS comparatifs afin de proposer système qui traite les questions comparatives et qui remède les problèmes des systèmes précédents. Nous proposons désormais un système qui clarifie les questions des utilisateurs pour des scénarios de comparaison en tenant compte de la complexité de la langue Arabe.

Ce chapitre décrit l'architecture et les différentes parties du système QA comparatif proposé. Nous introduisons d'abord le contexte et l'architecture générale. Nous enchaînons ensuite sur le processus suivi pour la création du dataset utilisé. Nous détaillerons plus tard chaque module du système.

III.2 Architecture de l'approche proposée

Nous proposons ,dans le cadre de ce projet de Master, un système de QA comparatif propre à la langue Arabe. Le système proposé nécessite l'intégration de deux axes de recherche. Des méthodes du domaine de l'analyse d'entités sont utilisées pour identifier les éléments de comparaison. De plus, la discipline de la recherche d'informations (RI) fournit des méthodes fiables pour récupérer des documents qui mentionnent ces éléments. Dans cette proposition, nous démontrons comment ces différentes contributions pourraient être réunies pour créer un système de Questions-Réponses pour les questions comparatifs.

III.2.1 Architecture du Système

Dans le but d'améliorer les performances des système de questions-réponses Arabe, nous fournissons ainsi un système qui, en détectant automatiquement les éléments de comparaison et les caractéristiques clés, permet aux utilisateurs d'obtenir des réponses aux questions comparatives. Un système construit en se basant sur des ensembles de données qui ont été soigneusement choisies et qui permettent une comparaison de haute précision dans notre conception. Dans cette étude, nous agrégeons des données issus du dataset : Arabic-SQuAD, ARCD, une traduction automatique de HotpotQA et un ensemble de données des questions comparatives généré automatiquement.

Pour approfondir davantage l'état de la compréhension de la langue naturelle Arabe, nous basons sur le modèle Arabert, qui a été spécifiquement entraîné pour comprendre la

langue Arabe. Ce modèle a donné une amélioration importante de précision dans divers tâches de TALN notamment la tâche des questions-réponses et la reconnaissance des entités nommées (NER).

Ce système est composé de quatre module : une module de filtrage des questions, un module d'extraction des éléments de comparaison, un module de RI et QA et un module de comparaison. Le premier module concerne la classification des questions. Ce module est utilisé pour filtrer les question de l'utilisateur afin de distinguer les questions comparatives des autres types de questions. Après avoir identifié notre question comparative, le module d'extraction des éléments de comparaison permet d'identifier les entités comparées et l'aspect comparatif. Nous utilisons un modèle de classification des tokens pour attribuer aux mots de la question leurs étiquettes respectives (Entité A, Entité B et l'aspect de comparaison). Ces composants seront utilisés comme points d'entrée pour le module suivant de recherche d'information et de QA. Ce dernier récupère les documents qui sont susceptibles de contenir des réponses aux sous-requêtes contenant une entité + l'aspect de comparaison.

Enfin, le dernier module prend en compte les réponses générées puis compare les réponses pour généré la réponse finale. La réponse à la question sera sous forme de oui, non, entité A, entité B ou égalité. L'architecture décrit est présenté par la figure III.1

Dans chaque module dans le système proposé, nous avons entraîné des modèles de réseaux de neurones en utilisant différents datasets. Dans les sections suivantes, nous discuterons la structure de chaque module en détail en mettant en évidence les datasets et les modèles utilisés.

III.3 Les Datasets

Pour concevoir une système qui traite les questions comparatives, les données sont une parties indispensable. Nous avons utilisé une multiple source de données pour l'entraînement des différent modèles. Nous avons d'abord généré automatiquement un ensemble de données de questions comparatives en langue Arabe. De plus, le dataset fournit avec la question, ses éléments de comparaison. En outre, une traduction automatique du dataset HotpotQA [Yang et al., 2018] est effectué pour obtenir des questions comparatives en langue Arabe vu que cet dataset contient ce type de question.

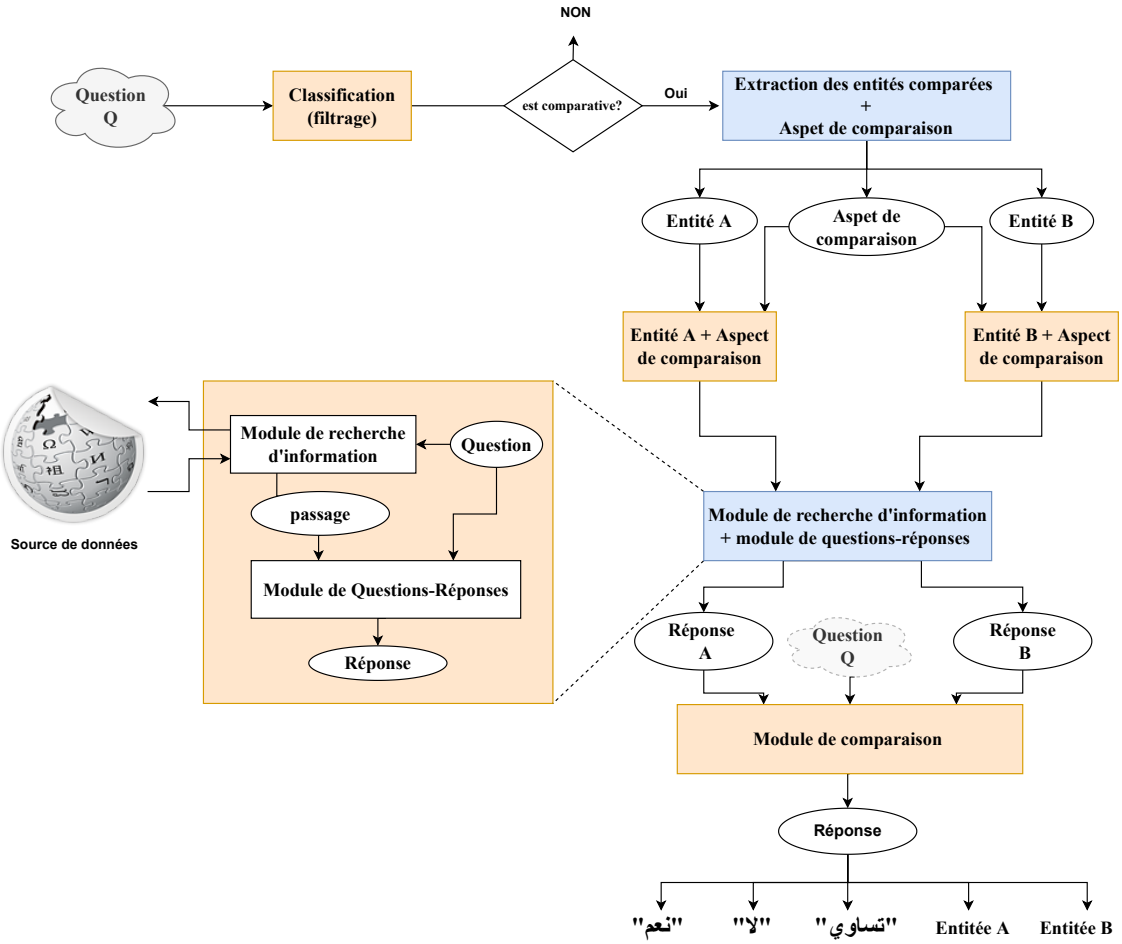

Figure III.1 — Architecture du système de questions-réponses comparatif proposé

Dans les parties suivantes, nous décrivons le processus suivi pour créer notre dataset, ensuite nous donnons une description du dataset hotpotQA :

III.3.1 Processus de Génération de Questions de Comparaison

Une question de comparaison est un type de question qui compare deux ou plusieurs entités similaires dans un certain aspect de l'entité [Yang et al., 2018].
Par exemple : ؟ السويد أم الأوروغواي ، من أصغر مساحة,(*Quel pays a une superficie plus petite, la Suède ou l'Uruguay ?*). Les entités comparées sont la Suède et l'Uruguay, et l'aspect de la comparaison est la superficie. L'idée ici est de trouver des paires d'entités similaires (A, B) qui partagent des aspects ou des propriétés communes f, puis de créer la question Q en utilisant un modèle T. Pour trouver la réponse a, les valeurs des propriétés ($f(A)$ et $f(B)$ pour les entités A et B respectivement) sont comparées et le résultat détermine la réponse.

III.3.1.1 La Collecte de Données

Nous avons initialement établi manuellement des listes d'entités de la même catégorie : animaux (92 entités), villes Arabes (22 entités) et pays du monde (191 entités), soit 304 entités au total. Ensuite, nous devons récupérer les propriétés de ces entités pour les utiliser comme aspects comparables. Pour y parvenir, nous avons utilisé Wikipedia API [1] et BeatifulSoup [2] pour explorer et analyser l'infobox de la page Wikipédia de chaque entité. Les propriétés sont présentées dans l'infobox sous forme de couples (clé ; valeur). Cette représentation structurée facilite la collecte des données et évite d'avoir recours à des techniques NLP sophistiquées pour extraire les propriétés d'une entité à partir d'un texte brut. Les données collectées sont stockées dans un fichier Excel où les colonnes sont les propriétés et les lignes sont leurs valeurs pour chaque entité.

III.3.1.2 Pré-traitement et Sélection des Propriétés

Nous avons effectué un prétraitement sur les données brutes afin de les rendre utilisables pour la génération de questions. Nous avons commencé par la suppression des caractères spéciaux, des liens et des mots non Arabes. Les sous-chaînes entre parenthèses ((), [],{}) ont également été supprimées car il s'agit généralement d'explications ou de notes. Nous avons remarqué que les étiquettes et les valeurs des propriétés peuvent être trouvées dans une variété de formes et d'écritures. Dans ce cas, une normalisation est nécessaire. La normalisation permet de placer des données similaires dans la même colonne et rend les valeurs comparables.

Par exemple, l'étiquette de la propriété الفصيلة*"famille d'un animal"* peut aussi se trouver sous la forme فصيلة où il s'agit du même mot sans le préfixe ال qui est l'article défini dans la langue Arabe, typiquement traduit par "le/la" en Français. Dans ce cas, on supprime simplement le préfixe.

Un autre cas pour les valeurs numériques : l'expression نسبة مئوية, ou le symbole "%" sont tous deux utilisés pour décrire un pourcentage. Nous avons choisi de remplacer l'expression par le symbole. Dans l'écriture des nombres, les parties centaines et milliers d'un nombre peuvent se trouver séparées par un point ".", une virgule "," ou un espace, et même chose

1. https ://ar.wikipedia.org/w/api.php
2. https ://pypi.org/project/beautifulsoup4/

pour les nombres décimaux, ce qui provoque une ambiguïté. Pour les centaines et les milliers, nous avons supprimé tous les séparateurs, et pour les nombres décimaux, nous avons conservé le point ".".

Après la normalisation, nous procédons maintenant à la sélection des propriétés. Nous supprimons tous les tuples incomplets dont la propriété ou la valeur peut être une chaîne vide (par exemple, les titres des groupes de propriétés). Nous préservons les propriétés comparables afin de pouvoir effectuer une comparaison. Par exemple, une propriété comme "site web officiel" n'est pas une caractéristique comparable. Nous nous assurons que les propriétés sélectionnées ont des valeurs pour au moins deux entités. Enfin, nous avons classé les propriétés sélectionnées en aspects quantitatifs et qualitatifs.

III.3.1.3 Préparation des Modèles

Nous définissons manuellement un ensemble de modèles pour générer des questions de comparaison et leurs sous-questions. Pour chaque propriété constituant un aspect de comparaison, nous créons une liste de modèles qui expriment la même question mais de manière différente. Cela permet de créer une plus grande diversité lors de la génération des questions de comparaison. Chaque modèle T comporte deux tokens variables : X et Y, à remplacer par deux étiquettes d'entités. Par exemple, comparer deux animaux en termes de "période de gestation", ما الحيوان الذي لديه فترة حمل أطول ، X أم Y ؟, (en Français : *Quel est l'animal qui a une période de gestation plus longue, X ou Y ?*). Les modèles sont classés en fonction du type de réponse en : Questions Oui/Non ou Questions à choix, dans lesquelles la réponse est l'une des entités comparées.

Pour plus de variété, nous ajoutons des modèles avec les prédicats opposés à ceux utilisés dans les modèles préparés précédemment. Par exemple, au lieu de "أطول" (en Français *plus long*), nous utilisons "أقصر"(en Français *plus court*). De plus, nous fournissons à chaque question de comparaison sa décomposition sous forme de deux sous-questions dérivées de la question initiale. Chaque sous-question cherche la valeur de la propriété d'une entité. Parallèlement, une liste de modèles pour les sous-questions est créée pour chaque propriété avec une variable X à remplacer par chaque étiquette d'entité. Dans le cas de l'exemple précédent, l'un des modèles de sous-questions serait : كم تدوم فترة الحمل عند X ؟, en Français*s :* ℕ- *Quelle est la durée de la période de gestation à X ?* .

III.3.1.4 Méthode de génération

À partir de l'ensemble des entités, nous établissons une liste de combinaisons de deux entités A et B de la même catégorie. Pour chaque couple d'entités, nous générons des questions sur toutes les propriétés dans lesquelles elles ont des valeurs. Pour chaque propriété f, nous sélectionnons un modèle aléatoire et remplaçons les tokens variables dans la chaîne de motifs par des étiquettes d'entités. Pour produire la réponse, nous comparons les valeurs des propriétés $f(A)$ et $f(B)$ en considérant que la question concerne la supériorité ou l'infériorité, le oui/non, ou un choix entre entités et si la propriété est quantitative ou qualitative.

Comme notre dataset est un dataset extractif basé sur le texte, nous fournissons deux paragraphes comme contexte. Chaque paragraphe est obtenu en extrayant tout le texte de l'infobox de la page Wikipedia de l'entité. La structure finale de l'ensemble de données contient les paires question-réponse de comparaison générées avec les deux entités comparées, l'aspect de comparaison et deux paragraphes comme contexte.

III.3.2 HotpotQA

HotpotQA [Yang et al., 2018] est un grand ensemble de questions-réponses contenant plus de 113k questions multi-saut de type passerelle et comparaison. L'ensemble des données a été collecté par le crowdsourcing sur la base d'articles de Wikipedia, où on présente aux crowd-workers des multiples documents ayant un contexte partagé et on leur demande explicitement de formuler des questions nécessitant un raisonnement sur tous les documents, et de fournir des faits justificatifs permettant de répondre à la question. Cela assure qu'il couvre des questions multi-sauts qui sont plus naturelles, et ne sont pas conçues sur la base d'un schéma de base de connaissances préexistant.

HotpotQA vise à faciliter le développement de systèmes de QA capables d'effectuer des raisonnements multi-sauts explicables sur un langage naturel diversifié, et propose un nouveau type de questions factoïdes : les questions de comparaison, dans lesquelles nous demandons aux systèmes de comparer deux entités sur certaines propriétés partagées afin de tester leur compréhension du langage et de concepts communs.

III.3.3 BERT : un modèle de langage pré-entraîné

BERT [Devlin et al., 2018] est un modèle de représentation du langage composé de plusieurs blocs de *transformers* (Vaswani et al., 2017)empilés . BERT a été pré-entraîné sur une grande quantité de texte non étiqueté afin de générer des vecteurs contextuels.

Il peut également être utilisé comme un modèle pré-entraîné avec une couche de sortie supplémentaire pour affiner les tâches NLP en aval, telles que le QA, NER, classification, etc.

Pour la tâche de questions-réponses, étant donné un passage contextuel $P = \{x_1, ..., x_m\}$ et une question $Q = \{q_1, ...q_n\}$, la réponse est $\left\{x_i\right\}_{l_a}^{l_b}$. Elle consiste en des tokens continus dans le passage contextuel, l_a et l_b représentent respectivement la position de début et de fin de l'étendue de la réponse dans le passage contextuel. Le but est de localiser la limite de la réponse l_a et l_b. L'entrée finale du modèle BERT est $\{[CLS], Q', [SEP], P'\}$ où P' et Q' sont des séquences de mots sous-tokénisés des P et Q originaux, et $[CSL], [SEP]$ sont des marques spéciales BERT utilisées pour séparer les paires de phrases et pour certaines tâches de classification. la figure III.2 illustre l'architecture de BERT pour la tâche de QA.

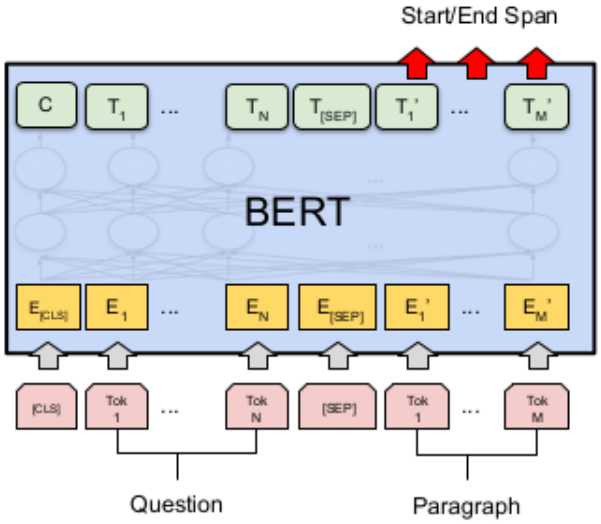

Figure III.2 — Architecture du modèle BERT

BERT est disponible en anglais, en chinois, et il existe une version pour plusieurs langues, appelée bert multilingue . BERT multilingue est disponible publiquement et couvre un large éventail de langues, dont la langue Arabe. Même si la version multilingue fournit d'excellents résultats pour de nombreuses langues, il a été démontré que leurs homologues monolingues obtenaient de meilleurs résultats.

III.4 Les Composants de l'architecture

La figure III.1 illustre l'architecture de notre système proposé. Ce système prend en entré une question de l'utilisateur. Il procède ensuite à l'analyse de celle-ci, en la filtrant puis en extrayant les éléments de comparaison pour produire une requête comparative qui sera divisée en deux requêtes distinctes correspondant à chaque entité de la question liée à l'aspect comparatif. Ensuite, il accède aux informations disponibles dans nos bases de données pour récupérer les réponses. Ces informations seront ensuite comparées et la bonne réponse sera présentée à l'utilisateur.

Cette section fournira la description détaillée de chacun des modules décrits.

III.4.1 Module de Classification des Questions

Ce module de classification des questions constitue l'une des étapes les plus importantes du processus question-réponse, à savoir la compréhension de la question. Comme chaque système QA, celui-ci commence par une question de l'utilisateur.La question doit donc être correctement analysée pour clarifier ce qu'elle signifie, de manière à ce que nous puissions être guidés sur la bonne voie pour trouver la réponse correcte et précise.

Nous nous appuyons entièrement sur le modèle bert entraîné qui vise à identifier le type de la question en extrayant la liste des mots-clés de cette dernière, ainsi que l'ensemble des heuristiques tels que les phrases citées dans la question, les noms propres, les noms, les verbes, les nominaux, les nominaux complexes (détectés par la reconnaissance des entités nommées), et l'objet de la question. Ensuite, en utilisant ces règles, nous pouvons vérifier si la question introduit une relation de comparaison entre deux objets, qui est souvent représentée par un adjectif ou un adverb comparatif.

En Arabe, les noms interrogatifs sont liées aux types et/ou portées de questions attendus, comme :

— هل، منْ : Il s'agit d'un outil interrogatif où ce type de question rechercherait

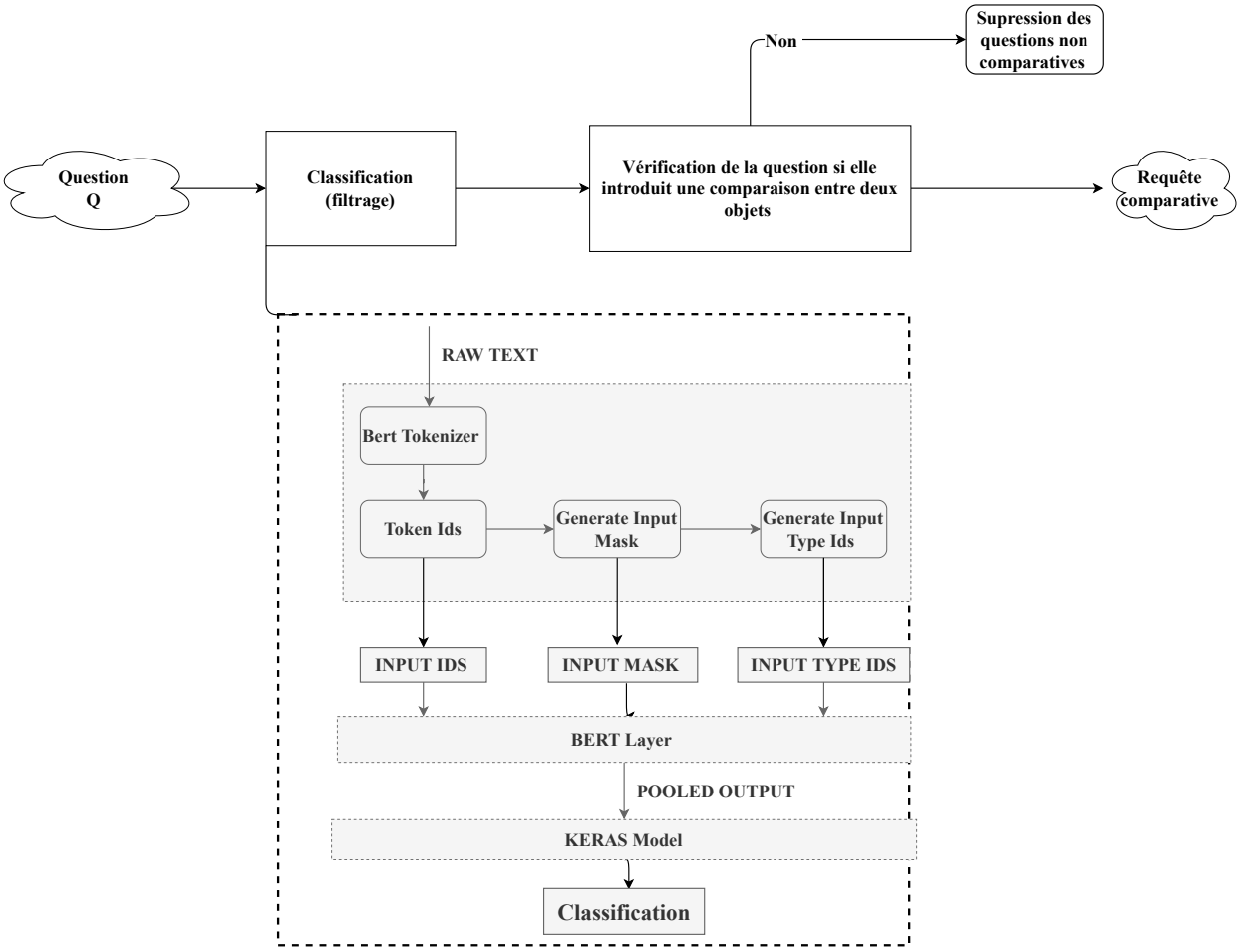

Figure III.3 — Architecture du module de classification

généralement une réponse Oui / Non.

Après avoir identifié le type de question, on peut, à partir du nom interrogatif, identifier la portée attendue ou le type de réponse ciblée ; ce qui sera utilisé plus tard dans les modules. Cependant, toutes les autres questions sont supprimées des résultats de filtrage initiaux car elles ne sont pas des structures argumentatives pertinentes.

III.4.2 Module d'Extraction des entités comparées et l'aspect de comparaison

Ce module est responsable d'attribuer des étiquettes aux mots de la question. Il reconnaît les entités nommées du premier module et utilise la tâche de classification des tokens à l'aide de notre dataset pour identifier les composants de la question comparative (entités, aspects, prédicats ou autres).

Le classifieur marque les composants avec leurs étiquettes respectives, chaque composant est sélectionné et étiqueté, respectivement, comme Entité A, Aspect de comparaison

et Entité B.

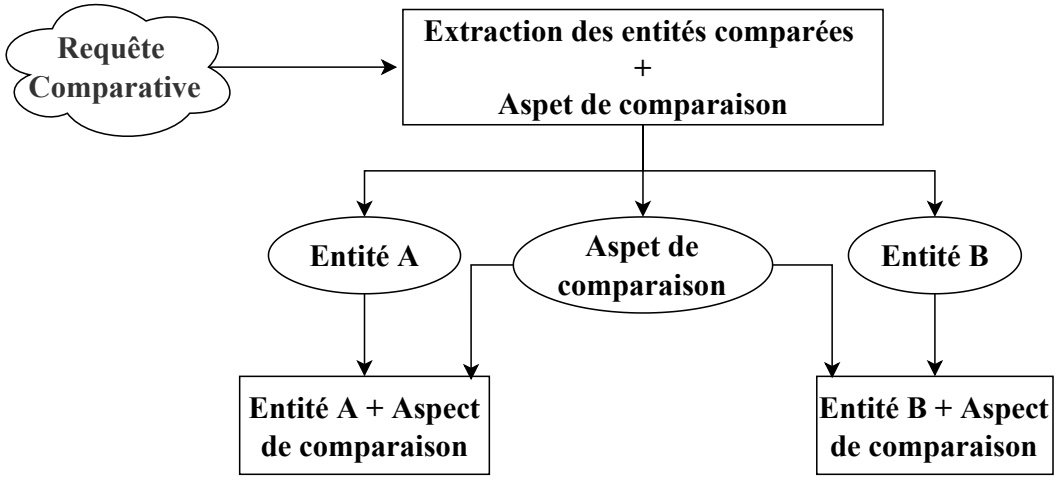

Figure III.4 — Architecture du module d'extraction

Dans la mesure où le principe de base du système que nous proposons est de pouvoir répondre à des questions comparatives sur deux objets différents, nous cherchons à formuler des sous-questions pour chaque sujet de la question par rapport à l'aspect de comparaison. Ces deux sous-questions serviront d'entrée au prochain module pour de récupérer les réponses pertinentes afin de les comparer pour arriver au réponse finale.

III.4.3 Module de recherche d'informations et Répondre aux Questions

Ce module utilise le système SOQAL [Mozannar et al., 2019] pour récupérer la réponse à la question. Ce système de QA à domaine ouvert est composé de trois modules. Un module de recherche de documents qui obtient les documents pertinents pour la question. Ensuite, un module de lecture qui extrait les réponses des documents récupérés. Et enfin un module de classement des réponses qui classe les réponses par ordre de pertinence en tenant compte des scores du récupérateur de documents et du lecteur. Il s'appuie sur la totalité de la Wikipédia arabe, et sa sortie est une portion de texte extrait de la Wikipédia qui devrait répondre à la question.

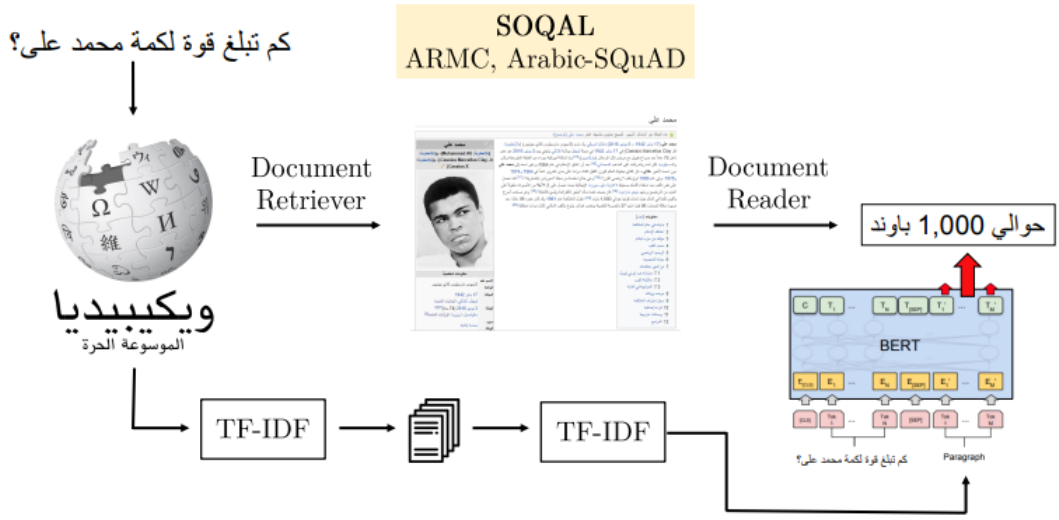

Figure III.5 — Architecture de SOQAL [Mozannar et al., 2019]

Ce module est divisé en trois tâches séquentielles :

— **Récupérateur hiérarchique de documents TF-IDF :** Ce module s'inspire des systèmes classiques de QA [Chen et al., 2017], il utilise un récupérateur de documents basé sur la fréquence des termes et l'inverse de la fréquence des documents (TF-IDF). Chaque document est d'abord tokénisé et déraciné à l'aide du tokéniseur arabe NLTK [Bird, 2006] où les mots d'arrêt sont supprimés. La matrice TF-IDF des poids de l'ensemble des documents est construite en utilisant le comptage des n-grammes pour prendre en compte l'ordre local des mots. Ensuite, les poids du vecteur TF-IDF de la question sont calculés sur la base du vocabulaire de l'ensemble des documents. Le score de chaque document est alors calculé comme la similarité en cosinus entre la question et les vecteurs des documents.

La matrice éparse est ainsi utilisée pour la représentation de la matrice TF-IDF afin d'accélérer les calculs. En dernier lieu, nous retournons les k documents ayant la plus grande similarité où k N est un hyperparamètre. Plus k est élevé, plus il est probable que l'ensemble des documents récupérés contient des documents pertinents, et plus le processus d'extraction des réponses est lent et sujet aux erreurs.

— **Lecteur de documents BERT :** Il utilise le modèle de langage pré-entraîné appelé mBERT (*multilingual BERT*). Le texte d'entrée est d'abord tokénisé à l'aide d'un vocabulaire partagé Wordpiece [Wu et al., 2016] de 104 langues, puis

il est intégré ; Chaque point d'entrée des paires de questions et de paragraphes est représenté comme une seule phrase séparée par un token spécial. Pour chaque token i dans le paragraphe, il prend l'état caché final du transformateur Ti et laisse la probabilité que i soit le début ou la fin de la réponse.

— **Classement des réponses :** Pour s'assurer que les scores des réponses et des documents sont sur la même échelle, ils sont normalisés individuellement en les faisant passer par une fonction softmax. La dernière étape pour obtenir la réponse à la question est de combiner les scores par une combinaison linéaire et de choisir la réponse maximale.

III.4.4 Module de Comparaison

Ce dernier module prend en entré la question comparatif initial avec les deux réponses obtenues du module de RIQA. Pour obtenir les réponses candidates, On obtient une probabilité pour chaque réponse candidate i. L'étape finale pour obtenir la réponse à la question est de combiner les scores par une combinaison linéaire et de choisir la réponse maximale.
Par conséquent, La réponse à la question sera sous forme de oui, non, entité A, entité B ou égale.

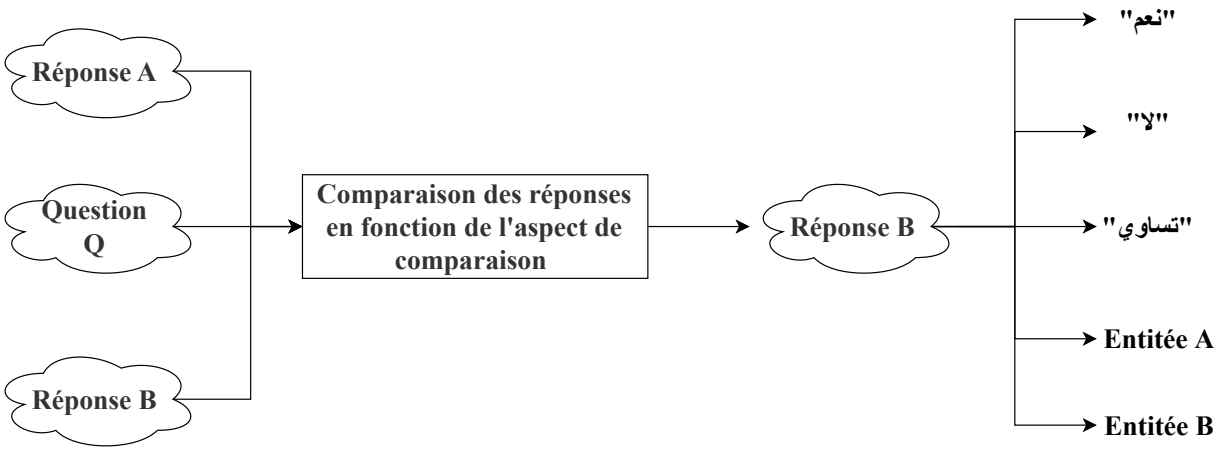

Figure III.6 — Architecture du module de comparaison

III.5 Conclusion

Comme tout cycle de développement logiciel d'un système, la phase de conception est une partie cruciale afin de minimiser les efforts des développeurs et maximiser les résultats.

Ce chapitre fournit les détails de conception de chaque composant de notre système. Le cycle de vie de la requête comparative passe d'abord par une phase de classification afin de filtrer les questions comparatives des autre questions vu que les questions comparatives ont un traitement particulier. Ensuite, les éléments de comparaisons sont extraits afin de formuler des requêtes intermédiaires. Nous avons utilisé le système SOQAL comme module de RI et QA, au final les réponses au requêtes intermédiaires sont introduites au module de comparaison pour produire la réponse finale.

Dans la phase suivante, l'accent sera mis sur la construction et la mise en œuvre du système QA proposé. Nous allons décrire les différentes étapes suivi lors du développement du système. Nous présenterons aussi les résultats d'expériences effectuées pour l'entraînement des modèles de deep-learning employés dans les composants du système.

Chapitre IV

IMPLÉMENTATION, TESTS ET ÉVALUATION

IV.1 Introduction

Le présent chapitre décrit les différentes étapes de réalisation de notre système de questions-réponses comparatif. Ceci a été effectué en suivant la conception présentée dans le chapitre précédent.Nous visons à décrire l'environnement logiciel dans lequel nous avons travaillé et à détailler la mise en œuvre des différents modules, les choix des outils, les résultats obtenues ainsi que l'évaluation du système réalisé.

IV.2 Choix techniques

Dans cette section, nous allons décrire brièvement les différents outils et langages utilisés afin de réaliser notre système questions-réponses.

IV.2.1 python

Python est un langage qui peut s'utiliser dans nombreux contextes et s'adapter aussi à tout type d'utilisation grâce à des bibliothèques spécialisées à chaque traitement. Il est particulièrement utilisé comme un langage de script pour automatiser des tâches simples mais fastidieuses par exemple un script qui récupérerait la météo sur internet ou qui s'intégrerait dans un logiciel de conception assistée par ordinateur afin d'automatiser certains enchaînements d'actions répétitives. On l'utilise comme un langage de développement de prototype lorsqu'on a besoin d'une application fonctionnelle avant de l'optimiser avec un langage de plus bas niveau. [Chaudhary, 2020]

IV.2.1.1 L'utilité du Python en Machine Learning

Ce langage de programmation présente des nombreuses caractéristiques intéressantes comme :

— Il est multiplateforme C'est-à-dire qu'il fonctionne sur de nombreux systèmes d'exploitation : Windows, Linux, Android, iOS, mac os x, depuis les mini-Ordinateurs Raspberry Pi jusqu'aux supercalculateurs.

— C'est un langage interprété, Un script Python n'a pas besoin d'être compilé pour être Exécuté, Il convient bien à des scripts d'une dizaine de lignes qu'à des projets complexes de plusieurs dizaines de milliers de lignes.

— Il est gratuit. Vous pouvez l'installer sur autant d'ordinateurs que vous voulez.C'est

un langage de haut niveau, Il demande relativement peu de connaissance sur le Fonctionnement d'un ordinateur pour être utilisé . [Chaudhary, 2020]

IV.2.2 Google Colab

Google Colaboratory largement connu sous le nom de Google Colab est un service sur le cloud fourni par Google à toute personne possédant un compte Gmail. Google Colab fournit un GPU pour la recherche aux personnes qui n'ont pas assez de ressources ou qui ne peuvent pas se le permettre. Le service Google Colab fournit 12 Go de RAM et 358,27 Go d'espace disque en une seule exécution. Chaque exécution dure 12 heures, après quoi l'exécution est réinitialisée et l'utilisateur doit établir à nouveau une connexion. Il s'agit de s'assurer que les gens n'utilisent pas le service GPU pour l'extraction de crypto-monnaie et à d'autres fins illégales.

Nous avons utilisé Google Colab dans l'entraînement des différents modèle vu que ceci n'est pas faisable en un temps opportun sur nos machine et qui nous permet de tester les différents paramètres.

IV.2.3 Transformers

la bibliothèque Transformers fournit des architectures à usage général (BERT, GPT-3 pour la compréhension du langage naturel (NLU) et la génération de langage naturel (NLG) avec plus de 32 modèles pré-entraînés dans plus de 100 langues et une interopérabilité approfondie entre Jax, PyTorch et TensorFlow [Fuchs and Poulain, 2021]

IV.2.4 Numpy

NumPy est un package utiliser pour les calculs scientifiques en Python. Il est idéal pour les opérations liées à l'algèbre linéaire, aux transformations de Fourier, ou au crunching de nombres aléatoires. Il peut être utilisé en guise de container multi-dimensionnel de données génériques. De plus, il s'intègre facilement avec de nombreuses bases de données différentes.

IV.2.5 Farasa

Farasa [Zhang et al., 2015], est une boîte à outils comprenant un module de segmentation/tokénisation, un tagueur POS, un diacriticien de texte arabe et un analyseur de dépendances. Farasa est différent des autres étiqueteurs POS car il peut conjointement segmenter, étiqueter et analyser le texte, ce qui évite la propagation des erreurs dans

la structure pipelinée et devrait exploiter les informations syntaxiques pour l'étiquetage POS.

IV.2.6 Le modèle AraBERT

Étant donné que la langue Arabe est une langue morphologiquement riche avec relativement peu de ressources et peu d'exploration par rapport à l'anglais, ce processus utilise le modèle basé sur le transformateur Arabe appelé ArabeRT, et nécessite une étape pour faire correspondre les termes de la question posée avec la grammaire normale des questions qui peuvent se trouver en Arabe.
ArabeRT est un modèle basé sur BERT qui définit un nouvel état de l'art pour de nombreuses tâches ultimes de la langue Arabe. Il est également 300 Mo plus petit que le BERT multilingue. Il est utilisé pour accomplir une grande variété de tâches NLP, y compris la reconnaissance d'entité nommée,la traduction automatique et les systèmes de questions-réponses. ArabeRT est un modèle de langage qui s'inspire de l'architecture BERT de Google. Six variantes du même modèle sont disponibles pour l'esxpérimentation : ArabeRTv0.2-base, ArabeRTv1- base, ArabeRTv0.1-base, ArabeRTv2-large, ArabeRTv0.2-large, et ArabeRTv2-base. Les propriétés architecturales de chacun de ces modèles sont indiquées dans le tableau ci-dessous. IV.1 .

Modèle	**Taille**		**Dataset**		
	MB	**Params**	**Taille**	**#Mots**	**#Phrases**
ArabeRTv0.1-base	543MB	136M	23GB	2.7B	77M
ArabeRTv1-base	543MB	136M	23GB	2.7B	77M
ArabeRTv0.2-base	543MB	136M	77GB	8.6B	200M
ArabeRTv0.2-large	1.38G	371M	77GB	8.6B	200M
ArabeRTv2-base	543MB	136M	77GB	8.6B	200M
ArabeRTv2-large	1.38G	371M	77GB	8.6B	200M

Table IV.1 — Attributs architecturaux des variantes de AraBERT

IV.3 Expérimentations et Résultats

IV.3.1 Module de classification de questions

IV.3.1.1 Dataset

Son dataset contient un échantillon de l'ensemble de données Arabic-SQUAD composé de 15000 questions non comparatives, et 15 000 questions de comparaison dont 5 000

proviennent de l'ensemble de données généré et 10000 sont une traduction automatique de des question comparatives dans hotpotQA.

Avant d'entamer le traitement de notre dataset, il est primordial de la diviser en plusieurs parties. Ses différentes parties servent à partitionner les données en deux catégories dénommées comme suit ‘’Train”, et ‘’Test”. Pour arriver à ce résultat, nous avons utilisé la méthode Split Data, qui est aussi facile à utiliser que performante, elle divise notre base de données avec des valeurs prédéfinis tel qu'utilisé dans notre cas 80% pour l'entraînement, 20% pour le test.

IV.3.1.2 Modèle

Dans cette partie nous allons donner plus de détails concernant les paramètres d'entraînement du modèle de classification ainsi que les résultats d'évaluation. D'abord nous avons commencé par l'importation des modèles que nous avons précisé dans les chapitres précédents, puis l'étape la plus importante afin d'aboutir à des scores de précisions les plus satisfaisants qui est la définition des paramétrage de notre modèle.
Dans ce paramétrage nous allons trouvé le nombre d'epochs de notre algorithme d'apprentissage puis la taille du batch et d'autres paramètres permettant l'entraînement de notre modèle.
Le choix des valeurs afin d initialiser les paramètres du modèle été choisi par rapport à sa performance ce qui veut dire que plusieurs tests ont été lancés et on a tirer le plus performant (voir Table IV.7).

Learning rate	2e-5
Batch size	16
Train epochs	2
Optimizer	Adam

Table IV.2 — Paramètres du modèle de classification

IV.3.1.3 Résultats

Dans cette partie nous allons explorer les résultats,ces derniers étaient très satisfaisants.
On peut remarquer dans le tableau descriptif IV.8 des scores obtenues, les taux de précisions étaient excellents avec un pourcentage de 99 % .

Epoch	Training Loss	Validation Loss	Macro F1	Accuracy
1	0.082200	0.025044	0.993500	0.995500
2	0.007300	0.023126	0.995500	0.995500

Table IV.3 — Résultats obtenus après entraînement du module de classification

Nous avons entraîné l'architecture *BertForSequenceClassification* avec la version du Arabert *bert-base-arabertv02*. Le résultat de l'entrainement donne un taux d'exactitude de 99%.Les valeurs de l'erreur de l'entraînement mesure à quel point (ou mal) notre modèle se porte bien. Si les erreurs sont élevées, la perte sera élevée, ce qui signifie que le modèle ne fait pas du bon travail. Sinon, plus il est bas, mieux notre modèle fonctionne.

IV.3.2 Module d'extraction des entités comparées et l'aspect de comparaison

IV.3.2.1 Dataset

Nous avons créé le dataset en format IOB en utilisant les données disponibles dans notre dataset pour créer ce dernier qui est un format d'étiquetage commun pour étiqueter les tokens dans une tâche de chunking en linguistique informatique.

Il a été introduit par Ramshaw et Marcus dans leur article [Ramshaw and Marcus, 1995]. Le préfixe I- devant une balise indique que la balise se trouve à l'intérieur d'un chunk. Une balise O indique qu'un jeton n'appartient à aucun chunk. Le préfixe B- devant une balise indique que la balise est le début d'un chunk qui suit immédiatement un autre chunk sans balise O entre eux. Il n'est utilisé que dans ce cas : lorsqu'un chunk vient après une balise O, le premier token du chunk prend le préfixe I-.

La figure IV.1 illustre un exemple d'une phrase généré sous forme IOB au quel les entités comparées sont mentionnés par B-ENT et l'Aspect de comparaison I-ASP B-ASP

	sentence #	word	tag
0	sentence: 1	من	O
1	NaN	من	O
2	NaN	هتين	O
3	NaN	البلدين	O
4	NaN	به	O
5	NaN	متوسط	B-Asp
6	NaN	عمر	I-Asp
7	NaN	،أعلى	O
8	NaN	البرازيل	B-Ent
9	NaN	أم	O
10	NaN	فالواتو	B-Ent
11	NaN	؟	O

Figure IV.1 — Exemple de phrase sous format IOB

Nous avons utilisé 5 tags ,le tableau suivant IV.4 illustre les statistiques de ces derniers .

tags	Nombre
O	98634
B-Ent	32565
I-Ent	7140
B-asp	12334
I-asp	6006

Table IV.4 — Statistique des tags

IV.3.2.2 Modèle

Avant de passé aux résultats, on va vous expliquer comment on est y arrivé à cette étape. On a commencé par l'importation du modèle préalablement mentionné, suivie d'une étape cruciale qui est l'initialisation des paramètres et qui constitue une étape majeure dans notre programme. En effet, le nombre d'epochs, la taille du batch, le Learning Rate, et d'autres paramètres ont été initialisé avant la partie d'entraînement du modèle, et le choix des valeurs n'a pas été fait au hasard, par contre il a fallu plusieurs tests pour arriver à cette solution optimale. Nous avons opté pour les valeurs suivantes pour les paramètres.LR

= 1e-5, la taille du Batch d'entrainement = 4, le Batch de validation = 2, et le nombre d'Epochs a été définie à 3.

IV.3.2.3 Résultats

Les résultats obtenus sont plus que concluant, étant donnée les excellentes valeurs de l'accuracy ainsi que celle de Macro F1 qui sont satisfaisante et qu'on va vous présenter dans la tableau IV.8

Epoch	Training Loss	Validation Loss	Macro F1	Accuracy
1	0.4563	0.6658	0.7958	0.7971
2	0.5189	0.6662	0.7998	0.79900

Table IV.5 — Résultats obtenus après entraînement du module d'extraction des éléments de comparaison

IV.3.3 Module de recherche d'information et répondre aux questions : SoQal

IV.3.3.1 Dataset

Le système Soqal est un système pour répondre à des questions en Arabe factuel dans un domaine ouvert en utilisant Wikipédia comme source de connaissances. Il est basé sur trois composants : (1) un récupérateur de documents utilisant une approche hiérarchique TF-IDF, (2) un modèle neuronal de lecture utilisant le transformateur bidirectionnel pré-entraîné BERT et enfin (3) un module de classement linéaire des réponses à obtenir. Son dataset contient l'ensemble de données ARCD composé de 1 395 questions posées par des crowdworkers sur des articles de Wikipedia, et Arabic-SQuAD, une traduction automatique de l'ensemble de données SQuAD contenant 48 344 questions.

Hussein Mozannar [Mozannar et al., 2019] ,a utilisé la méthode Split Data, Arabic-SQuad est divisé à 80-10-10% en trois parties pour le train, le développement et le test : Arabic-SQuad-Test est composé de 2 966 questions sur 24 articles ; notez que les articles sont distincts entre les parties. De même, l'ARCD est divisé 50-50 entre la train et le test avec ARCD-Test qui comporte 702 questions sur 78 articles.

IV.3.3.2 Modèle

On a commencé par l'importation du modèle AraBERT au lieu de mBERT (multi-lingue) , puis nous avons initialiser les paramètres . En effet, le nombre d'epochs, la taille

du batch, le Learning Rate, et d'autres paramètres ont été initialisé avant la partie d'entraînement du modèle. Nous avons opté pour les valeurs suivantes pour les paramètres.LR = 3e-5, la taille du Batch d'entrainement = 8, l'étape d'accumulation du gradient = 1, et le nombre d'Epochs a été définie à 5.

IV.3.3.3 Résultats

L'entraînement était réussi et il nous a donné comme résultat à un taux de correspondance exacte de 35.3276% et un score F1 de 69.4685%.
Aprés changement de modèle nous avons obtenues un meilleur resultat. le tableauIV.6 illustre la comparaision entre ses derniers .

	EM	**F1**
mBERT	34.2	61.3
BERT	35.3276	69.468

Table IV.6 — Comparaison entre les résultats après utilisé le modèle BERT

IV.3.4 Module de comparaison

IV.3.4.1 Dataset

Son dataset est extrait de notre premier dataset. Nous avons définie 5 classes à prédire a partir des donnée d'entrée qui sont la question et les deux sous-réponses qui présente la réponse de chaque entité après avoir posés la question principale sur chacune d'eux , il contient 20000 questions.

IV.3.4.2 Modèle

Les paramètres de notre modèle étaient choisi par attention afin de nous retrouver les meilleurs résultats. C'est un modèle de classification multi labels on a définit 5 classes à prédire à partir de la donnée d'entrée qui est la question.
On a défini 2 couches pour notre algorithmes d'apprentissage. Plusieurs pistes d'optimisation de notre résultats sont à envisager dans le futur notamment l'augmentation ou l'enrichissement des données.(voir Table IV.7).

Learning rate	2e-5
Batch size	16
Train epochs	2
Optimizer	Adam

Table IV.7 — Paramètres du modèle de comparaison

IV.3.4.3 Résultats

Dans cette partie nous allons découvrir l'efficacité de notre modèle de classification. L'entraînement était bien réussi. On peut remarquer dans le tableau descriptif IV.8 des scores obtenues, les taux était satisfaisantes avec un pourcentage de 92% . Ces résultats étaient obtenues suite à plusieurs tests et Re paramétrage de notre modèle .

Epoch	Training Loss	Validation Loss	Macro F1	Accuracy
1	0.210400	0.153998	0.892371	0.925750
2	0.147800	0.153403	0.895033	0.928000

Table IV.8 — Résultats obtenus après entraînement du module de comparaison

IV.4 Conclusion

Nous avons décrit le long de ce chapitre l'ensemble des étapes réalisation de notre système de Questions-réponses et son évaluation. Dans un premier lieu, nous avons présenté les différents outils, plateformes et bibliothèques utilisées lors du développement du système. Ensuite, nous avons détaillé pour chaque module : l'ensemble des données utilisé ainsi que les hyperparamètres des modèles d'apprentissage profond entraînés. Nous avons décrit aussi la phase évaluation, selon plusieurs configuration, en commençons par le protocole d'évaluation. Nous avons calculé et présenté les différents résultats des tests obtenus, selon plusieurs métriques, en l'occurrence, la F-mesure, l'exactitude, l'exact-matching, etc.

les résultats obtenues montrent clairement l'intérêt des QAS comparatives, et pour les concrétiser il nous faut élargir le dataset des données stockées .

CONCLUSION ET PERSPECTIVES

Conclusion et perspectives

Dans ce projet de master, nous avons souhaité surmonter les limites des approches actuelles en soulignant explicitement la nécessité de clarifier les enquêtes comparatives, en améliorant spécifiquement les performances du système de questions-réponses en arabe, et en approfondissant l'état de la compréhension de la langue arabe. Nous proposons un système comparatif de QA spécifique à la langue arabe, basé sur le modèle AraBERT, qui a été spécifiquement entraîné sur la langue Arabe, car la précision de ce modèle a montré une amélioration significative dans diverses tâches NLP, notamment la tâche question-réponse et la reconnaissance d'entités nommées (NER).

Notre système est composé de quatre modules, à savoir un module de filtrage des questions, un module d'extraction des éléments de comparaison, un module de RI et de QA et un module de comparaison. Le premier module concerne le filtrage des questions. Ce module est utilisé pour filtrer les questions des utilisateurs afin de distinguer les questions de comparaison des autres types de questions en utilisant le modèle pré-entraîné BERT avec une couche de sortie supplémentaire pour affiner les tâches NLP en aval, telles que le QA, le NER et la classification. Après avoir identifié notre question comparative, le module d'extraction de caractéristiques de comparaison identifie les entités comparées et l'aspect comparatif. Nous utilisons un modèle de classification token pour attribuer aux mots de la question leurs étiquettes respectives (Entité A, Entité B, et l'aspect comparatif). Ces composants sont ensuite utilisés comme points d'entrée pour le module suivant de recherche d'information et de QA. Ce dernier récupère les documents susceptibles de contenir les réponses aux sous-requêtes contenant une entité + l'aspect comparaison. Enfin, le dernier module prend en compte les réponses générées, puis compare les réponses pour générer la réponse finale. La réponse à la question sera sous forme de oui, non, entité A, entité B ou égalité.

Après avoir testé notre système, nous nous sommes aperçus que les taux de précision du classification des questions étaient excellents, avec un pourcentage de 99%. Puis de plus les résultats du module d'extraction des entités comparés et l'aspect de comparaison étaient satisfaisantes, avec une précision de 79%. De même, les résultats du retriever du module de recherche d'information se sont avérés positifs avec un taux de 69%. Puis finalement notre module de comparaison était bien réussi avec un score de précision très satisfaisant qui s'élève a 92% .

Néanmoins, notre travail reste à perfectionner à travers plusieurs propositions d'amélioration. Pour cela, nous envisageons de :

— Ajouter un module d'extraction des arguments lors d'une comparaison plus complexe.
— Introduire dans les datasets le types de questions de comparaisons à plusieurs aspects.
— Ajouter une interface graphique pour une meilleur expérience d'utilisation du système.
— Étendre le système pour qu'il prend en charge d'autres types de questions.
— Intégrer le système dans un véritable environnement de test à grande échelle (web) pour une évaluation plus fiable.

Bibliographie

[Abdelnasser et al., 2014] Abdelnasser, H., Ragab, M., Mohamed, R., Mohamed, A., Farouk, B., El-Makky, N. M., and Torki, M. (2014). Al-bayan : an arabic question answering system for the holy quran. In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 57–64.

[Abdi et al., 2020] Abdi, A., Hasan, S., Arshi, M., Shamsuddin, S. M., and Idris, N. (2020). A question answering system in hadith using linguistic knowledge. *Computer Speech Language*, 60 :101023.

[Abouenour et al., 2012] Abouenour, L., Bouzoubaa, K., and Rosso, P. (2012). Idraaq : New arabic question answering system based on query expansion and passage retrieval.

[Ahmed et al., 2017] Ahmed, W., Bibin, P., and Anto, B. P. (2017). Question answering system based on neural networks. *International Journal of Engineering Research*, 6(3) :142–144.

[Akour et al., 2011] Akour, M., Abufardeh, S., Magel, K., and Al-Radaideh, Q. (2011). Qarabpro : A rule based question answering system for reading comprehension tests in arabic. *American Journal of Applied Sciences*, 8(6) :652–661.

[Al Chalabi, 2015] Al Chalabi, H. M. (2015). *Question processing for Arabic question answering system.* PhD thesis, The British University in Dubai (BUiD).

[Al-Khawaldeh, 2019] Al-Khawaldeh, F. T. (2019). Answer extraction for why arabic questions answering systems : Ewaq. *arXiv preprint arXiv :1907.04149.*

[Al-Shargi and Rambow, 2015] Al-Shargi, F. and Rambow, O. (2015). Diwan : A dialectal word annotation tool for arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 49–58.

[Alwaneen et al., 2021] Alwaneen, T. H., Azmi, A. M., Aboalsamh, H. A., Cambria, E., and Hussain, A. (2021). Arabic question answering system : a survey. *Artificial Intelligence Review*, pages 1–47.

[Aouichat and Guessoum, 2017] Aouichat, A. and Guessoum, A. (2017). Building talaa-afaq, a corpus of arabic factoid question-answers for a question answering system. In

International Conference on Applications of Natural Language to Information Systems, pages 380–386. Springer.

[Artetxe et al., 2019] Artetxe, M., Ruder, S., and Yogatama, D. (2019). On the cross-lingual transferability of monolingual representations. *arXiv preprint arXiv :1910.11856*.

[Atef et al., 2020] Atef, A., Mattar, B., Sherif, S., Elrefai, E., and Torki, M. (2020). Aqad : 17,000+ arabic questions for machine comprehension of text. In *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–6. IEEE.

[Bakari et al., 2016] Bakari, W., Bellot, P., and Neji, M. (2016). Researches and reviews in arabic question answering : principal approaches and systems with classification. In *Proc. Int. Arab Conf. Inf. Technol.(ACIT)*, pages 1–9.

[Bakari et al., 2014] Bakari, W., Trigui, O., and Neji, M. (2014). Logic-based approach for improving arabic question answering. In *2014 IEEE international conference on computational intelligence and computing research*, pages 1–6. IEEE.

[Basuki and Purwarianti, 2016] Basuki, S. and Purwarianti, A. (2016). Statistical-based approach for indonesian complex factoid question decomposition. *Statistical-based Approach for Indonesian Complex Factoid Question Decomposition*, 8(2) :356–373.

[Bekhti and Al-Harbi, 2013] Bekhti, S. and Al-Harbi, M. (2013). Aquasys : A question-answering system for arabic. In *WSEAS International Conference. Proceedings. Recent Advances in Computer Engineering Series*, volume 25, pages 19–27. WSEAS.

[Benajiba et al., 2007] Benajiba, Y., Rosso, P., and Lyhyaoui, A. (2007). Implementation of the arabiqa question answering system's components. In *Proc. Workshop on Arabic Natural Language Processing, 2nd Information Communication Technologies Int. Symposium, ICTIS-2007, Fez, Morroco, April*, pages 3–5. Citeseer.

[Biltawi et al., 2021] Biltawi, M. M., Tedmori, S., and Awajan, A. (2021). Arabic question answering systems : Gap analysis. *IEEE Access*, 9 :63876–63904.

[Bird, 2006] Bird, S. (2006). Nltk : the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

[Bondarenko et al., 2022] Bondarenko, A., Ajjour, Y., Dittmar, V., Homann, N., Braslavski, P., and Hagen, M. (2022). Towards understanding and answering comparative questions. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining (WSDM 2022). ACM.*

[Bondarenko et al., 2020] Bondarenko, A., Braslavski, P., Völske, M., Aly, R., Fröbe, M., Panchenko, A., Biemann, C., Stein, B., and Hagen, M. (2020). Comparative web search questions. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 52–60.

[Brini et al., 2009] Brini, W., Ellouze, M., Mesfar, S., and Belguith, L. H. (2009). An arabic question-answering system for factoid questions. In *2009 International Conference on Natural Language Processing and Knowledge Engineering*, pages 1–7. IEEE.

[Chaudhary, 2020] Chaudhary, M. (2020). tf idf vectorizer scikit learn.

[Cheddadi, 2014] Cheddadi, A. (2014). Three-levels approach for arabic question answering systems. *Diss. Ecole Mohammadia d'Ingénieurs*.

[Chekalina et al., 2021] Chekalina, V., Bondarenko, A., Biemann, C., Beloucif, M., Logacheva, V., and Panchenko, A. (2021). Which is better for deep learning : python or matlab ? answering comparative questions in natural language. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics : System Demonstrations*, pages 302–311.

[Chen et al., 2017] Chen, D., Fisch, A., Weston, J., and Bordes, A. (2017). Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv :1704.00051*.

[Clark et al., 2020] Clark, J. H., Choi, E., Collins, M., Garrette, D., Kwiatkowski, T., Nikolaev, V., and Palomaki, J. (2020). Tydi qa : A benchmark for information-seeking question answering in typologically diverse languages. *Transactions of the Association for Computational Linguistics*, 8 :454–470.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert : Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv :1810.04805*.

[Education, 2020] Education, I. C. (2020). What is natural language processing ?

[Ezzeldin et al., 2013] Ezzeldin, A. M., Kholief, M. H., and El-Sonbaty, Y. (2013). Alqasim : Arabic language question answer selection in machines. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 100–103. Springer.

[Ezzeldin and Shaheen, 2012] Ezzeldin, A. M. and Shaheen, M. (2012). A survey of arabic question answering : challenges, tasks, approaches, tools, and future trends. In *Proceedings of The 13th international Arab conference on information technology (ACIT 2012)*, pages 1–8.

[Foundation, 2022] Foundation, W. (2022). Information retrieval.

[Fuchs and Poulain, 2021] Fuchs, P. F. and Poulain, P. (2021). *Introduction à la programmation Python pour la biologie*. PhD thesis, Université de Paris.

[Gal and Ghahramani, 2016] Gal, Y. and Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. *Advances in neural information processing systems*, 29.

[Hammo et al., 2002] Hammo, B., Abu-Salem, H., Lytinen, S. L., and Evens, M. (2002). Qarab : A : Question answering system to support the arabic language. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*.

[Hovy et al., 2000] Hovy, E., Gerber, L., Hermjakob, U., Junk, M., and Lin, C.-Y. (2000). Question answering in webclopedia. In *TREC*, volume 52, pages 53–56.

[Hu, 2006] Hu, H. (2006). A study on question answering system using integrated retrieval method. *Unpublished Ph. D. Thesis, The University of Tokushima, Tokushima*.

[Indurkhya and Damerau, 2010] Indurkhya, N. and Damerau, F. J. (2010). *Handbook of natural language processing*. Chapman and Hall/CRC.

[Ismail and Homsi, 2018] Ismail, W. S. and Homsi, M. N. (2018). Dawqas : A dataset for arabic why question answering system. *Procedia computer science*, 142 :123–131.

[Jijkoun et al., 2003] Jijkoun, V., Mishne, G., de Rijke, M., et al. (2003). Building infrastructure for dutch question answering. *DRIPROCEEDINGS*, page 22.

[Kaur and Gupta, 2013] Kaur, J. and Gupta, V. (2013). Effective question answering techniques and their evaluation metrics. *International Journal of Computer Applications*, 65(12).

[Khillare et al., 2014] Khillare, S. A., Shelke, B. A., and Mahender, C. N. (2014). Comparative study on question answering systems and techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, 4(11) :775–778.

[Kolomiyets and Moens, 2011] Kolomiyets, O. and Moens, M.-F. (2011). A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24) :5412–5434.

[Kurdi et al., 2014] Kurdi, H., Alkhaider, S., and Alfaifi, N. (2014). Development and evaluation of a web based question answering system for arabic language. *Computer Science Information Technology (CS IT)*, 4(02) :187–202.

[Kwiatkowski et al., 2019] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. (2019). Natural questions : a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7 :453–466.

[Labs, 2020] Labs, C. F. F. (2020). Evaluating qa : Metrics, predictions, and the null response.

[Latifi, 2018] Latifi, M. (2018). Using natural language processing for question answering in closed and open domains.

[Lewis et al., 2019] Lewis, P., Oğuz, B., Rinott, R., Riedel, S., and Schwenk, H. (2019). Mlqa : Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv :1910.07475*.

[Liddy, 2001] Liddy, E. D. (2001). Natural language processing.

[Lim et al., 2009] Lim, N. R., Saint-Dizier, P., and Roxas, R. E. (2009). Some challenges in the design of comparative and evaluative question answering systems. In *Proceedings of the 2009 Workshop on Knowledge and Reasoning for Answering Questions (KRAQ 2009)*, pages 15–18.

[Longpre et al., 2020] Longpre, S., Lu, Y., and Daiber, J. (2020). Mkqa : A linguistically diverse benchmark for multilingual open domain question answering. *arXiv preprint arXiv :2007.15207*.

[Manning et al., 2018] Manning, C. D., Raghavan, P., and Schutze, H. (2018). Introduction to information retrieval cambridge university press 2008. *Ch*, 20 :405–416.

[Mishra and Jain, 2016] Mishra, A. and Jain, S. K. (2016). A survey on question answering systems with classification. *Journal of King Saud University-Computer and Information Sciences*, 28(3) :345–361.

[Mohammed et al., 1993] Mohammed, F., Nasser, K., and Harb, H. M. (1993). A knowledge based arabic question answering system (aqas). *ACM SIGART Bulletin*, 4(4) :21–30.

[Moldovan et al., 2003] Moldovan, D., Paşca, M., Harabagiu, S., and Surdeanu, M. (2003). Performance issues and error analysis in an open-domain question answering system. *ACM Transactions on Information Systems (TOIS)*, 21(2) :133–154.

[Mooers, 1961] Mooers, C. (1961). From a point of view of mathematical etc. techniques. *Towards information retrieval, pp. xvii–xxiii. Butterworths*, 17 :528.

[Mooers, 1950] Mooers, C. S. (1950). Editor's corner :" coding, information retrieval, and the rapid selector". *Journal of the American Society for Information Science*, 1(4) :225.

[Mozannar et al., 2019] Mozannar, H., Hajal, K. E., Maamary, E., and Hajj, H. (2019). Neural arabic question answering. *arXiv preprint arXiv :1906.05394*.

[Nabil et al., 2017] Nabil, M., Abdelmegied, A., Ayman, Y., Fathy, A., Khairy, G., Yousri, M., El-Makky, N. M., and Nagi, K. (2017). Alquans-an arabic language question answering system. In *KDIR*, pages 144–154.

[Neji et al., 2016] Neji, Z., Ellouze, M., and Belguith, L. H. (2016). Iqas : Inference question answering system for handling temporal inference. In *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, pages 1–5. IEEE.

[NIOS, 2020] NIOS (2020). Chapter 5b15 information retrieval concept of scope.

[Niu, 2007] Niu, Y. (2007). *Analysis of semantic classes : toward non-factoid question answering*. University of Toronto.

[Pechsiri and Piriyakul, 2016] Pechsiri, C. and Piriyakul, R. (2016). Developing a why–how question answering system on community web boards with a causality graph including procedural knowledge. *Information Processing in Agriculture*, 3(1) :36–53.

[Rajpurkar et al., 2018] Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know : Unanswerable questions for squad. *arXiv preprint arXiv :1806.03822*.

[Rajpurkar et al., 2016] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad : 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv :1606.05250*.

[Ramshaw and Marcus, 1995] Ramshaw, L. A. and Marcus, M. P. (1995). Text chunking using transformation-based learning.

[Roberts and Gaizauskas, 2004] Roberts, I. and Gaizauskas, R. (2004). Evaluating passage retrieval approaches for question answering. In *European Conference on Information Retrieval*, pages 72–84. Springer.

[Sabour et al., 2017] Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. *Advances in neural information processing systems*, 30.

[Schildwächter et al., 2019] Schildwächter, M., Bondarenko, A., Zenker, J., Hagen, M., Biemann, C., and Panchenko, A. (2019). Answering comparative questions : Better than ten-blue-links ? In *Proceedings of the 2019 Conference on Human Information Interaction and Retrieval*, pages 361–365.

[Shaalan, 2014] Shaalan, K. (2014). A survey of arabic named entity recognition and classification. *Computational Linguistics*, 40(2) :469–510.

[Tanwar et al., 2014] Tanwar, P., Prasad, T., and Datta, K. (2014). An effective reasoning algorithm for question answering system. *International Journal of Advanced Computer Science and Applications (IJACSA)*.

[Tellex, 2003] Tellex, S. (2003). *Pauchok : A modular framework for question answering.* PhD thesis, Massachusetts Institute of Technology.

[Trigui et al., 2010] Trigui, O., Belguith, L. H., and Rosso, P. (2010). Defarabicqa : Arabic definition question answering system. In *Workshop on Language Resources and Human Language Technologies for Semitic Languages, 7th LREC, Valletta, Malta*, pages 40–45.

[Turner and Rainie, 2020] Turner, E. and Rainie, L. (2020). Most americans rely on their own research to make big decisions, and that often means online searches.

[Wang et al., 2007] Wang, C., Xiong, M., Zhou, Q., and Yu, Y. (2007). Panto : A portable natural language interface to ontologies. In *European Semantic Web Conference*, pages 473–487. Springer.

[Wu et al., 2015] Wu, Y., Hori, C., Kashioka, H., and Kawai, H. (2015). Leveraging social qa collections for improving complex question answering. *Computer Speech Language*, 29(1) :1–19.

[Wu et al., 2016] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system : Bridging the gap between human and machine translation. *arXiv preprint arXiv :1609.08144*.

[Yang et al., 2018] Yang, Z., Qi, P., Zhang, S., Bengio, Y., Cohen, W. W., Salakhutdinov, R., and Manning, C. D. (2018). Hotpotqa : A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv :1809.09600*.

[Yogish et al., 2016] Yogish, D., Manjunath, T., and Hegadi, R. S. (2016). A survey of intelligent question answering system using nlp and information retrieval techniques. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(5) :536–540.

[Zeng et al., 2015] Zeng, D., Liu, K., Chen, Y., and Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1753–1762.

[Zhang et al., 2015] Zhang, Y., Li, C., Barzilay, R., and Darwish, K. (2015). Randomized greedy inference for joint segmentation, pos tagging and dependency parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 42–52.