Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE SAAD DAHLEB DE BLIDA 1

Faculté des Sciences

Département d'Informatique

MÉMOIRE DE MASTER INFORMATIQUE IL

Les Word-Embedding pour l'évalutaion automatique des réponses courtes en apprentissage en ligne : Application à la langue arabe.

Réalisé par :

HANNOUFI Mohammed Hamza
HENNICHE Adel Nassim

Proposé et encadré par :

Mme OUAHRANI Leila

Composition de jury :

M.	BALA	MAHFOUD	Président
M.	FERFERA	SOFIANE	Examinateur

Soutenu le 30/06/2018

ملخص

يعتبر تقييم المتعلمين اليوم أضعف حلقة في منصات التعلم عبر الإنترنت ، فباستثناء الاسئلة ذات الإجابات المتعددة فإن تقييم المتعلم لا يزال إما عمل يدوي أو عمل آلي يميل إلى التطرف (صواب أو خطأ). في هذا العمل، نحن مهتمون بالأسئلة ذات الإجابات القصيرة.و بشكل أكثر تحديدًا، لقياس تأثير الـ

"Word Embeddings"

على أنظمة التقييم التلقائي أو الآلي للإجابات القصيرة التي تتناول اللغة العربية. تكمن الفائدة من استخدام هذا النهج في اللغة العربية في أنه لا يتطلب الكثير من الموارد اللغوية وكونه مستقلاً عن اللغة. للقيام بذلك ، قمنا بإجراء طريقة تجريبية على أساس 3 مقاربات تجذيرية: بدون التجذيع ، تجذيع خفيف وتجذيع ثقيل. لقد تم في هذا العمل اقتراح و برمجة العديد من نماذج حساب التشابه بالإضافة إلى 4 أدوات تم تطويرها. لقد قمنا بدمج هذا النهج مع نماذج تشابه أخرى دلالية ونحوية. ساعدت النماذج على تحسين خطوط الأساس الخاصة بنا والخاصة بتلك الأعمال ذات صلة بعملنا

الكلمات المفتاحية : التقييم التلقائي للإجابات القصيرة، مقاييس التشابه، المعالجة الآلية للغة، التجذيع.

Résumé

L'évaluation de l'apprenant est aujourd'hui le maillon faible des plateformes de formation en ligne. À l'exception des réponses à choix multiples, l'évaluation de l'apprenant reste une tâche manuelle ou tend vers du binaire (vrai ou faux). Dans ce travail, nous nous intéressons aux questions à réponses courtes. Plus précisément, à mesurer l'effet des " Word Embeddings" sur les systèmes d'évaluation automatiques des réponses courtes traitant de la langue arabe. L'intérêt d'utiliser une telle approche pour la langue arabe réside essentiellement dans le fait qu'elle ne nécessite pas beaucoup de ressources linguistiques et se présente comme indépendante de la langue. Pour ce faire, nous avons réalisé une synthèse expérimentale basée sur 3 approches de Stemming: Sans stem, un stem léger et un stem lourd. Au cours de cette synthèse, plusieurs modèles de calcul de similarité ont été proposés et implémentés et 4 outils ont été développés. Nous avons combiné cette approche à d'autres modèles de calcul de similarité sémantique et syntaxique. Les modèles ont permis d'améliorer nos propres baselines et d'améliorer ceux des travaux connexes.

Mots clés : Evaluation automatique des réponses courtes, ASAGS, Word Embedding, Mesures de similarité, Traitement automatique de la langue, Stem, modèle d'espace vectoriel

Abstract

Learner assessment is now the weak link in online learning platforms. With the exception of multiple-choice answers, the learner's assessment remains a manual task or tends to binary (true or false). In this work, we are interested in questions with short answers. More specifically, to measure the effect of "Word Embeddings" on the automatic evaluation systems of short answers dealing with the Arabic language. The advantage of using such an approach for the Arabic language lies mainly in the fact that it does not require a lot of language resources and is independent of the language. To do this, we carried out an experimental synthesis based on 3 Stemming approaches: Without stem, a light stem and a heavy stem. During this synthesis, several similarity calculation models have been proposed and implemented as well as 4 tools have been developed. We have combined this approach with other semantic and syntactic similarity models. The models helped improve our own baselines and improve those of related works.

Keywords: Automatic short answer grading, ASAGS, Word Embedding, Similarity Measures, Natural language processing, Stem, Vector space model

Remerciements

Ce présent travail de mémoire qui marque la fin de notre cycle de Master n'a pu aboutir que grâce à la conjonction des efforts de plusieurs personnes. Qu'il nous soit permis d'exprimer notre profonde gratitude à tous ceux qui nous ont encouragés et accompagnés jusqu'au terme du présent mémoire.

A notre Dieu Tout puissant, qui nous a comblé de sa force et son amour ;

Nos profondes gratitudes et sincères remerciements vont à nos enseignants et membres de l'équipe pédagogique du département d'informatique qui nous ont garanti de bonnes conditions de travail.

Nous tenons à remercier tout particulièrement et à témoigner vivement toute notre reconnaissance à notre promotrice Mme Ouahrani Leila pour tous les efforts qu'elle a fournis, de s'être investi corps et âme avec rigueur scientifique pour la direction de ce mémoire. Ses qualités pédagogiques remarquables nous ont permis de profiter de ses connaissances et ont contribué à l'avancement de notre travail en ne négligeant ni ses conseils avisés ni ses critiques constructives.

Nos gratitudes vont aussi à Dr Wail GOMAA de l'université du Caire (Egypte) de nous avoir fourni son jeu de données de qualité (GOMMA DataSet) ainsi qu'aux responsables du centre de calcul de l'université de Bouira qui ont mis à notre disposition les ressources matérielles nécessaires à l'accomplissement de ce projet.

Nos vifs remerciements de reconnaissance sont également adressés aux membres du jury pour l'intérêt qu'ils ont porté à notre travail, nous en sommes honorés et nous leur exprimons notre profonde reconnaissance.

Nous remercions nos très chers parents, qui ont toujours été là pour nous, « Vous avez tout sacrifié pour vos enfants n'épargnant ni santé ni efforts. Vous nous avez donné un magnifique modèle de labeur et de persévérance. Nous sommes redevables d'une éducation dont nous sommes fiers ».

Nous tenons à gratifier les familles : Hannoufi, Henniche, Yousri, Touel, Nechadi, Hadji, Debieb, Madour et tous ceux qui nous ont aidés tant moralement que matériellement, qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont fait de nous ce que nous sommes aujourd'hui.

A nos compagnons de lutte universitaire pour le courage et l'amour qu'ils nous ont caractérisés durant tout ce temps de vie estudiantine.

En fin que les amis, frères et soeurs dont les noms ne sont pas cités ne nous tiennent pas rigueur, nos pensées vont aussi vers eux.

Liste des tableaux

Liste de figures

Table des matières

I. Introduction générale

Ce premier chapitre introduit le contexte global de notre travail. Il nous permet de préciser la problématique, de formuler les objectifs et de situer les contraintes et les enjeux de notre travail de recherche.

i. Introduction

Un des domaines les plus avantagés par l'avancement technologique des moyens de communications est sûrement le domaine de l'apprentissage en ligne, plus communément appelé « E-Learning ». Nous retrouvons dans la littérature plusieurs définitions concernant ce dernier. Une définition des plus pertinentes est celle de l'Union européenne : «Le e-Learning est l'utilisation des nouvelles technologies multimédias de l'Internet pour améliorer la qualité de l'apprentissage en facilitant d'une part l'accès à des ressources et à des services, d'autre part les échanges et la collaboration à distance». [1]

Aujourd'hui pratiquement tout établissement universitaire en Algérie est doté d'une plateforme offrant des services d'apprentissage en ligne et de plus en plus d'enseignants exploitent la panoplie de services offerts par ces plateformes qui couvrent la totalité du cursus d'apprentissage et permettent la diffusion des ressources pédagogiques, une communication plus efficace entre apprenants, enseignant et concepteur de cours et d'autre part le contrôle continu des connaissances des étudiants.

Plusieurs systèmes d'évaluation automatique sont disponibles dans le domaine de l'enseignement et particulièrement l'enseignement en ligne depuis de nombreuses années, mais principalement pour les questions de reconnaissance où les étudiants (apprenants) doivent choisir la réponse correcte à partir d'options données telles que les *Questions à Choix Multiples* (QCM). Les recherches antérieures ont montré que de telles questions de

reconnaissance sont insuffisantes car elles ne permettent pas de saisir de multiples aspects des connaissances acquises, comme le raisonnement et l'auto-explication. En revanche, les questions *à réponses courtes* (quelques mots à quelques phrases construites en langage naturel) qui recherchent les réponses construites par les examinés en langage naturel ont été jugées plus efficaces pour évaluer les connaissances acquises par les apprenants. Cependant, l'automatisation de l'évaluation de ces réponses n'est pas simple en raison de variations linguistiques (une réponse donnée pourrait être articulée de différentes façons), de la nature subjective de l'évaluation (multiples réponses possibles), du manque de cohérence dans la notation humaine, … etc. Ce qui peut expliquer le manque d'outils connus sous l'abréviation « ASAGS » pour Automatic Short Answer Grading Systems permettant une correction automatique efficace et ce malgré la pertinence des questions à réponses courtes dans le processus d'apprentissage.

ii. Problématique

Le concept principal de l'évaluation automatique des réponses courtes consiste à comparer la réponse de l'apprenant(RA) à la réponse de référence de l'enseignant appelée Réponse Modèle(RM) et à mesurer la similitude(ou similarité) entre les deux réponses puis à convertir cette similarité en note appelée aussi score. La première étape qui est le calcul de la similarité constitue une difficulté essentielle dans l'automatisation du processus de correction. Chaque apprenant a son propre style d'écriture ainsi qu'un niveau de maitrise linguistique et un vocabulaire différent, ce qui rend difficile la mise en œuvre d'un système capable de reconnaitre toutes les réponses similaires à la réponse type et d'évaluer la consistance d'une réponse donnée.

La plupart des recherches dans l'évaluation automatique des réponses courtes traitent de *l'anglais*. L'arabe est une langue répandue parlée par plus de 300 millions de personnes à travers le monde. Du point de vue du langage naturel, la langue arabe se caractérise par une ambiguïté élevée et une morphologie riche et complexe. Ce sont des aspects qui ralentissent les progrès dans la considération de la langue arabe dans le contexte de l'évaluation automatique des questions à réponses courtes, par rapport aux progrès réalisés dans l'anglais et dans d'autres langues latines. Une autre limite importante est constatée par **le manque considérable de ressources linguistiques** pour la langue arabe : corpus spécialisés (de domaine), lexiques et dictionnaires, outils de traitement,… Très peu de travaux ont traité de l'arabe dans le contexte de l'évaluation automatique de questions courtes.

Plusieurs mesures de similarité sont utilisées pour mesurer la similarité entre la réponse de référence ou modèle de l'enseignant et la réponse de l'étudiant(ou l'apprenant). Ces mesures

sont classées dans trois approches principales ; les mesures syntaxiques (comparaison des chaines de caractères) et les mesures sémantiques (comparaison de contenu sémantique) et les mesures hybrides (qui combinent les deux). Les Word Embeddings (issus du Deep Learning et du modèle d'espace vectoriel (représentation vectorielle de mots)), très répondus ces dernières années, promettent une meilleure précision pour le calcul de la similarité sémantique en appliquant une approche statistique basée sur le calcul vectoriel de mots. L'intérêt d'utiliser une telle approche pour la langue arabe réside essentiellement dans le fait qu'elle ne nécessite pas beaucoup de ressources linguistiques et se présente comme indépendante de la langue.

iii. Objectifs

Nous voulons avec ce travail, d'un coté évaluer l'impact d'une telle approche statistique sur les outils ASAGS et d'un autre coté évaluer la flexibilité des approches Word Embeddings à être combinées avec d'autres mesures de similarité notamment syntaxique et ce pour la langue arabe.

En effet, nous voulons atteindre les objectifs suivants :

1. Evaluer l'impact des Word Embeddings sur la langue arabe dans le but d'estimer leur potentiel à répondre aux problèmes connus de la langue arabe particulièrement celui du manque de ressources linguistiques comme les corpus, les lexiques, les dictionnaires et outils de traitement . Pour ce faire, nous développons plusieurs modèles de calcul de similarité autour des Word Embeddings. Ces modèles sont évaluées de façon expérimentale sur des ensembles de données (DataSets) exprimées dans la langue arabe et tirés de la littérature.
2. Etudier la combinaison des approches déjà développées avec les Word Embeddings avec d'autres mesures de similarité notamment syntaxiques et sémantique et analyser à quel point l'hybridation améliore les résultats.
3. Enfin, le travail est couronné par un système d'évaluation automatique permettant de comparer la réponse modèle de l'enseignant avec la réponse de l'étudiant et de générer un score le plus proche possible de l'évaluation manuelle. Ce système tourne autour de plusieurs outils de traitement de la langue arabe développés par nous-mêmes pour palier au manque de ressources linguistiques de traitement de la langue arabe.

iv. Importance de l'étude

L'intérêt croissant des chercheurs envers les systèmes d'évaluation automatique des réponses courtes est compréhensible par rapport aux avantages espérés d'une automatisation d'un tel processus:

1. Réduire la charge de travail des enseignants en automatisant une partie de la tâche d'évaluation des apprenants. En effet, la correction manuelle des copies est une des tâches les plus couteuses en temps et en charge cognitive pour un enseignant et nécessite des efforts ainsi que des ressources qui pourraient être économisées.
2. Fournir aux étudiants des informations détaillées sur leur période d'apprentissage de manière plus efficace que l'évaluation traditionnelle.
3. Fournir aux apprenants des résultats immédiats ainsi que des commentaires détaillés à propos de leurs erreurs ou une redirection vers la partie du cours à revoir.
4. Une objectivité maximale des résultats étant donné qu'un programme est insensible aux conditions externes comme des émotions, contrairement à un jury qui est influencé par son état mental (heureux, triste, en colère…).
5. Un système d'évaluation non supervisé et totalement automatisé réduirait le risque d'erreurs humaines et les soupçons de triche ou de subjectivité.
6. Intégrer la culture d'évaluation au travail quotidien des apprenants dans un environnement d'e-Learning.

Le calcul de similarité étant la base de plusieurs autres domaines du Traitement Automatique de la Langue(TAL), ce travail peut être adapté à d'autre domaines du TAL tel que la traduction automatique, la détection de plagiat de contenu, la recherche d'information, …

v. Portée et limites de la recherche

Ce travail a pour ambition de mesurer l'impact des Word Embedding dans les systèmes ASAGS conçu pour la langue arabe. Ainsi :

- Toute autre ressource, mot ou paraphrase d'une langue autre que l'arabe sera ignoré.
- Les systèmes d'évaluation automatique des réponses courtes traitent par définition les questions à réponses courtes uniquement, les autres types de questions tels que : les questions de sélection, les essaies, la classification, les codes (programmes à évaluer) ou les formules mathématiques ne peuvent être considérés dans notre travail.

  La figure I.1 permet de situer les questions à réponses courtes parmi les types de questions courantes où les méthodes d'évaluation automatique peuvent être appliquées ainsi que la profondeur de l'objectif cognitif à atteindre (le rappel de connaissances appelé aussi « RECALL ») dans le processus d'apprentissage.

Dans la suite de notre travail nous considérons qu'une réponse est « courte » si elle respecte les propriétés suivantes:

1. La réponse doit être rédigée dans un langage naturel (l'arabe dans notre cas).
2. Elle doit être issue (formulée) de connaissances externes à la question elle-même (contrairement aux questions à réponses multiples).
3. Sa longueur doit être comprise entre une phrase et un paragraphe ne dépassant pas les 100 mots.
4. La réponse valorise le contenu et non pas le style d'écriture (contrairement aux essais).
5. La question à réponse courte doit être précise et objective, si elle est susceptible de varier en fonction de la personnalité et la perception de chacun alors elle est inappropriée pour une évaluation informatisée. Un exemple de question à réponse courte inappropriée pour les tests informatisés est: "Définir le terme « Démocratie» ". Il existe de nombreuses définitions standard du terme "démocratie". De plus, divers répondants peuvent avoir leur propre point de vue sur le terme et peuvent le définir différemment. Autrement dit, les réponses attendues seront probablement subjectives et ne seront pas simplement des paraphrases d'un seul concept. Le critère du bien et du mal pour les réponses des élèves n'est pas très clair. [2]

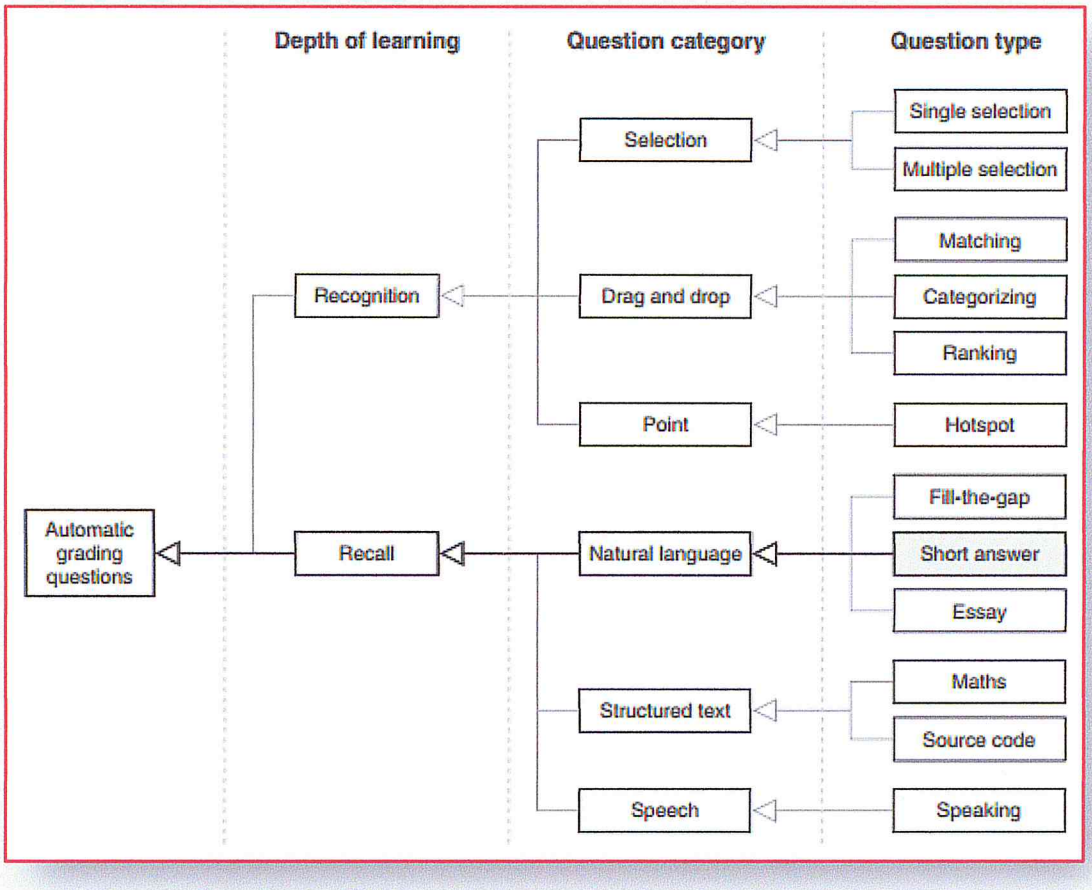

Figure I.1 Une vue hiérarchique des types de questions courantes où les méthodes d'évaluation automatique peuvent être appliquées [3]

- Le calcul de similarité étant une tâche répondue dans plusieurs bronches du traitement automatique de la langue et peut être appliqué dans le cadre de plusieurs thématiques telles que la traduction automatique, la détection de plagiat de contenu ou la recherche d'information. Ce projet a pour but d'optimiser ce calcul de similarité dans un contexte précis d'évaluation automatique des réponses courtes où l'étudiant propose une réponse proche au maximum des concepts et principes vus en cours. Et c'est autour de ce contexte que les interprétations des résultats sont faites.
- Outres les mots non arabes, les chiffres sont aussi ignorés. Toute valeur numérique contenue dans les textes à traiter est supprimée en espérant pouvoir lever cette limite dans les prochains travaux.

- Enfin, et au vu des résultats peu concluants des analyseurs morphosyntaxique à identifier les entités nommées pour la langue arabe (noms de personnes, de villes, …), ces derniers sont considérés dans cette recherche comme de simple chaines de caractères et ne bénéficient pas d'un traitement particulier en tant qu'entité nommée.

- Notre travail étant orienté vers toute personne impliqué dans un quelconque processus d'apprentissage, nous confondons dans toute la suite du document les termes : « élève », « étudiant », « apprenant », « examiné ».

vi. Structure du mémoire:

Le reste de notre document est structuré en 3 chapitres:

- Dans le chapitre deux, nous présentons l'état de l'art du domaine pour mieux situer notre travail.
- Dans le chapitre trois, nous exposons notre approche méthodologique ainsi que tous les modèles de calcul de similarité que nous proposons.
- La synthèse expérimentale que nous avons réalisé est décrite dans le chapitre quatre où nous présentons une évaluation des approches et où nous mettons l'accent aussi sur les outils que nous avons développés. Nous clôturons le chapitre par une discussion sur les résultats obtenus.
- Enfin, la conclusion générale nous permet de mettre le point sur le travail réalisé ainsi que ses perspectives futures.

II. Etat de l'art

Notre travail se situe à l'intersection de plusieurs domaines de recherche à savoir le domaine de similarité de textes, celui des systèmes d'évaluation automatique, des Word Embeddings ainsi que le domaine du TAL. Ainsi cet état de l'art nous permet de situer notre travail par rapport aux différents domaines : retracer l'évolution méthodologique des systèmes ASAG, revoir les différentes approches de calcul de similarité de textes, introduire les Word Embeddings avec leurs différents modèles de génération. Nous focalisons l'analyse sur les défis et enjeux de la langue arabe lorsqu'elle est considérée dans le domaine de la similarité et celui de l'évaluation automatique. Nous terminons l'état de l'art par la description des travaux connexes à notre approche.

i. Les systèmes « ASAG »

1. Définition

La recherche dans la notation des réponses en langage naturel a une histoire remontant aux années soixante. Depuis, les techniques se sont ramifiées en fonction du type de question générant plusieurs sous domaines de recherche. Pour ce qui est des questions ciblées par ce travail : « L'évaluation automatique des réponses courtes formulées en langage naturel à des questions objectives en utilisant des méthodes computationnelles » [4]. Par définition, ses

méthodes cherchent à comprendre et à reproduire le comportement humain en s'appuyant sur les concepts fondamentaux de l'informatique théorique [5]. Dans les systèmes ASAGS, le comportement cible de reproduction est la correction de copies par un examinateur. Comme pour la correction manuelle, la correction automatique passe par plusieurs étapes nécessitant, pour certaines, des ressources et connaissances qui influent sur la qualité du système. Les ASAGS se basent sur le contenu plutôt que sur le style. Une mauvaise qualité d'écriture(de formulation) est, jusqu'à un certain point, tolérée facilitant ainsi la production d'une idée, une méthode, ou un produit original qui est la meilleure façon de vérifier l'acquisition des informations et des connaissances selon la taxonomie de Bloom[6] comme le montre la figure II.1 :

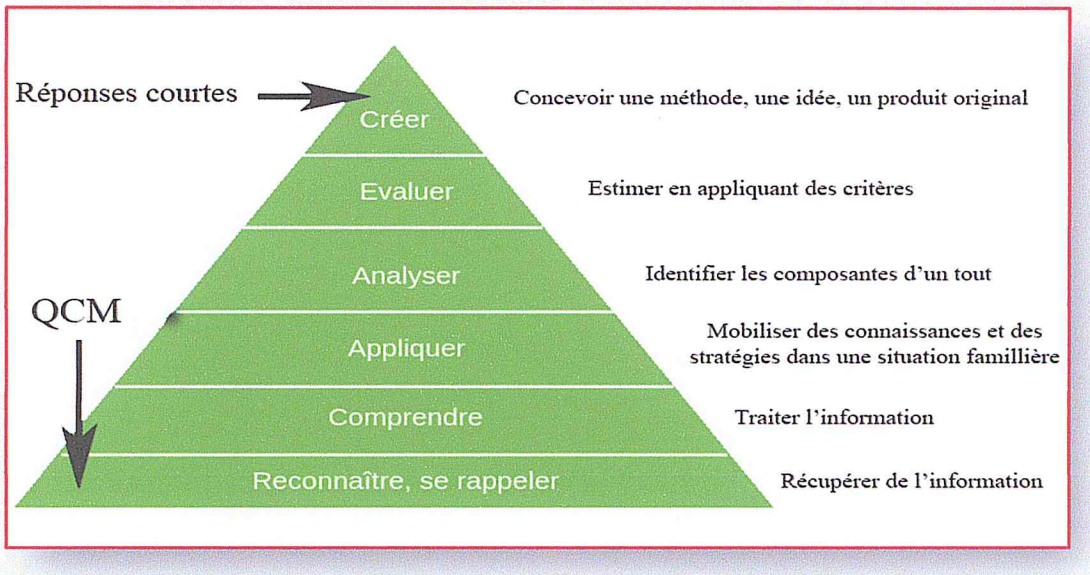

Figure II.1 La taxonomie de Bloom [6]

2. Pipeline de développement de système ASAG

La notion de pipeline est bien soutenue par plusieurs domaines de la recherche en traitement du langage naturel, comme il a été démontré par Wachsmuth dans divers de ses travaux tel que l'extraction de relation et le remplissage de Template [7] ou l'extraction efficace de l'information [8]. Pour les systèmes ASAGS, la forme générale d'un pipeline de développement est constituée de 5 processus et 6 artéfacts comme le montre la figure II.2.

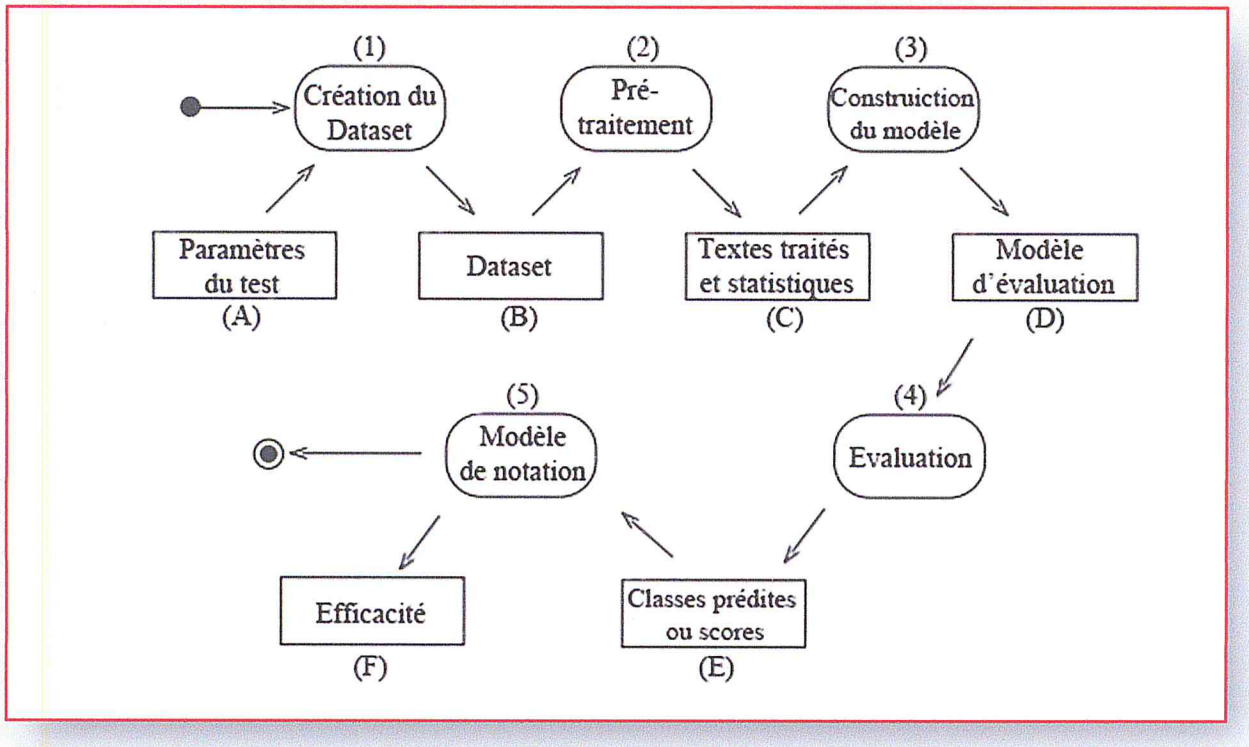

Figure II.2 Pipeline de développement des systèmes ASAGS [3]

Le fonctionnement général du pipeline est relativement simple :

1) Création du dataset(Ensembles de tests) :

Il est nécessaire pour la création du dataset de bien définir le contexte et les besoins du teste comme le domaine d'évaluation, la langue, et les propriétés des questions... . Une fois cette étape (A) terminée, le dataset est créé en regroupant chaque couple de « RM, RA » (réponse modèle, réponse de l'apprenant) pour avoir un dataset organisé (B).

2) Prétraitement :

Ici on fait appel aux techniques de traitement automatique des langages naturels tel que la normalisation (regroupement des différentes formes que peut revêtir un mot, soit : le nom, le pluriel, le verbe à l'infinitif …) pour générer du texte comprenant des formes de mots normalisées de la façon souhaitée et générer des statistiques (C).

3) Construction du modèle :

La construction du modèle est une des étapes les plus importantes du pipeline étant donné que le modèle construit(D) est l'entité chargée du calcul de similarité entre les réponses modèles et les réponses des apprenants. L'approche et les mesures de similarité à utiliser sont définies

et implémentées au cours de cette étape et certaines connaissances du domaine ou ressources peuvent être requises.

4) Evaluation :

Une fois le modèle construit et le dataset préparé, l'évaluation des réponses est effectué générant un ensemble de similarités ou de classes(E).

5) Modèle de notation :

Les prédictions issues de l'étape d'évaluation sont généralement des similarités et non pas des notes à proprement parlé. Pour assurer un passage vers des notes fiables, un module de notation est construit en se basant sur des algorithmes de classification. Dès lors, des comparaisons entres les notes manuelles et les notes du système sont faites afin de calculer l'efficacité de ce dernier(D).

ii.Analyse historique des systèmes « ASAG »

De nombreux travaux retracent les grandes lignes historiques du développement des systèmes ASAG, à savoir le travail de Valenti [9] et Pérez-Marín [10] et bien d'autres. Un des plus récents et les plus complets est « The Eras and Trends of Automatic Short Answer Grading » [3] où les auteurs regroupent les systèmes d'évaluation automatiques des réponses courtes dans 5 ères différentes représentés dans la figure II.3 avec les outils ASAGS les plus connus de chaque ère:

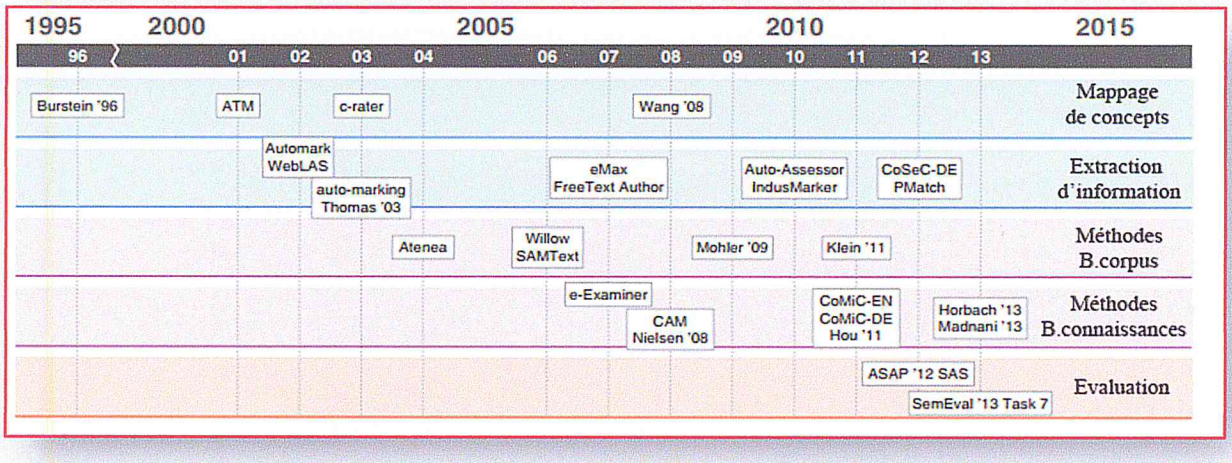

Figure II.3 Historique des systèmes ASAG [3]

1. Ere mappage de concepts

L'idée du mappage de concepts est de considérer les réponses des apprenants comme constituées de plusieurs concepts, et de détecter la présence ou l'absence de chaque concept lors de l'évaluation. Des questions appropriées doivent donc faciliter cette idée, comme une question qui demande une solution à un problème plus une justification, ou une question qui demande plusieurs explications au même problème. En voici un exemple tiré de la littérature[11] : les apprenants devraient fournir de multiples raisons pour expliquer la baisse du nombre de décès dans les forces de police au fil du temps. Trois exemples de concepts sont:

- «De meilleures conditions économiques signifient moins de crimes»
- «La technologie médicale avancée a permis de sauver plus de vies»
- «Les escrocs ont maintenant une capacité réduite d'acheter des armes».

Notons que le mappage de concept est exprimé au niveau de la phrase. Il est possible de plonger dans un niveau de détail plus précis concernant des fragments individuels (typiquement des paires de mots et des triplets), mais ce problème est généralement connu sous le nom de mappage de facettes. Par exemple, Nielsen [12] effectuent une «classification par facettes» et posent une question où les élèves sont interrogés sur les sons produits par les instruments à cordes, et la réponse de référence est: «Une longue corde produit un son faible». Toujours en se référant à Nielsen[12], cette réponse au niveau de la phrase peut être décomposée en quatre facettes:

- corde / longue: "Il y a une longue corde"
- produit / corde: "La corde produit quelque chose"
- produit / son: "Un son est en cours de production"
- son / bas : "Le son est bas".

Basé sur ce processus, pratiquement n'importe quel concept peut être décomposé en facettes.

Un des programmes les plus aboutis de cette ère est le « C-rater » (The Concept Rater) [13]. Son approche vise à faire correspondre les concepts entre les réponses de l'enseignant et de l'élève en jouant sur la flexibilité et la maniabilité de la langue. L'appariement est basé sur un ensemble de règles et une représentation canonique des textes utilisant la variation syntaxique, l'anaphore, la variation morphologique, les synonymes et la correction orthographique. Plus précisément, les réponses des enseignants sont introduites sous forme de phrase distincte pour

chaque concept. Cela simplifie l'évaluation puisque un seul concept est considéré à la fois lors du la notation. Cette technique évite d'avoir recours à une solution indirecte comme le partitionnement de la question en plusieurs parties [14] ce qui a conduit à une plus grande précision [15]. En outre, le format d'entrée en langage naturel est avantageux par rapport à d'autres systèmes qui nécessitent une expertise et l'utilisation d'un langage de balisage [16].

2. Ere d'extraction d'information

Les techniques d'extraction d'informations (IE) extraient des informations pertinentes à partir de morceaux de texte analysés syntaxiquement en appliquant un ensemble de patterns. Les patterns sont définis soit par rapport à une analyse de surfaces des textes (mots, phrases) en utilisant des méthodes tels que les expressions régulières, soit sur des éléments structuraux tels que des étiquettes de parties du discours (PoS : Part Of Speech). Dans le cas des réponses courtes, elles sont généralement créées par des experts en la matière pour indiquer des concepts importants qui devraient être présents dans les réponses. Nous pourrions citer pour l'exemple Dzikovska[17] qui a utilisé un analyseur syntaxique et un ensemble de règles écrites à la main pour extraire des représentations sémantiques à partir de réponses des apprenants qui ont ensuite été comparées à des représentations sémantiques de réponses correctes attendues fournies par des tuteurs. Le principal défi avec les techniques basées sur l'extraction de l'information est d'arriver à des modèles capables de couvrir toutes les variations possibles dans les réponses des apprenants. En outre, cela doit être fait manuellement pour chaque exercice d'évaluation par des experts en la matière, ce qui rend l'ensemble de l'exercice coûteux. D'un autre côté, comme ces techniques fonctionnent sur le principe de l'identification des patterns, elles ont l'avantage de créer facilement un retour d'information pour les élèves, basé sur patterns manquants.

Le correcteur AutoMark [18] est bien représentatif de cette ère ; il effectue pour la notation une comparaison entre les réponses des enseignants et des apprenants qui sont représentés par des arbres d'analyses générés grâce à des techniques d'extraction d'information. Deux approches sont décrites, à savoir, une approches «aveugle» et autre approche dite «modérée».

3. Ere des méthodes basées sur le corpus

Les méthodes basées sur le corpus exploitent les propriétés statistiques des grands corpus de documents. Ces méthodes permettent la prise en charge des synonymes dans les systèmes ASAGS évitant la limitation au vocabulaire de la réponse modèle. Une autre technique pour augmenter le vocabulaire consiste à utiliser des corpus bilingues parallèlement pour analyser la fréquence des paires de termes en langue secondaire. Des synonymes avec des traductions particulièrement communes peuvent ensuite être incorporés dans les réponses des enseignants.

D'autres techniques basées sur les corpus telles que l'analyse sémantique latente (LSA)[19] se sont révélées performantes pour traiter quelques lacunes des approches utilisant une analyse de surfaces des textes. LSA utilise une structure de données matricielle basée sur de grands corpus pour modéliser la relation sémantique. Ces techniques projettent des documents dans un sous-espace vectoriel de faible dimension choisi de façon appropriée, et, combiné avec la similarité cosinus, arrivent à une estimation raisonnable de la similarité sémantique.

Un exemple des systèmes utilisant des corpus et, plus précisément, le web comme corpus est le SAMText (Short Answer Measurement of TEXT) [20]. Le programme applique une variante de LSA basée sur une structure de données d'index inversé qui est répertorié à partir d'une analyse Web. En revanche Bukai et al. [20] soutiennent que l'idée de l'exploration du web comme corpus est plus appropriée pour les réponses courtes que pour les réponses longues parce que l'exploration web peut être adaptée à chaque sujet au lieu d'essayer de modéliser tout le langage à la fois.

Les approches basées sur la connaissance font aussi recours au corpus et utilisent des ressources telles que Wordnet qui est un excellent système de référence lexicale en ligne pouvant être utilisé pour trouver beaucoup d'informations sur un mot tels que les synonymes, ou des ontologies spécifiques à un domaine pour estimer à quel point deux propositions sont similaires. Cependant, peu de langues ont assez de ressources linguistiques nécessaires au bon fonctionnement des ces approches.

4. Ere de l'apprentissage automatique

Les systèmes d'apprentissage automatique utilisent généralement un certain nombre de mesures extraites grâce aux techniques de traitement du langage naturel, qui sont ensuite combinées en une seule note en utilisant un modèle de classification ou de régression. La classification et la régression sont les deux paradigmes d'apprentissage supervisé les plus populaires dans la littérature sur l'apprentissage automatique. Les deux techniques tentent de construire des fonctions à partir d'un ensemble de données étiquetées dans le but de prédire par la suite les étiquettes des futures données non marquées. Dans un espace de caractéristiques à valeur réelle de n dimensions, les techniques de classification apprennent des fonctions de type Rn → A où A est un ensemble d'étiquettes de classes discrètes. Dans notre contexte, les données sont des réponses et les scores sont des étiquettes.

Les méthodes impliquant des mots-clés et des n-grammes sont typiques de cette catégorie, tout comme les arbres de décision et les machines vectorielles. Le correcteur CAM (Content Assessment Module) [21] utilise un classificateur k-plus proche voisin (KNN) et des caractéristiques qui mesurent le pourcentage de chevauchement du contenu sur différents niveaux linguistiques entre les réponses de l'enseignant et celles de l'élève.

5. Ere d'évaluation

Contrairement aux quatre ères précédentes qui décrivent les méthodes, l'ère de l'évaluation est indépendante des méthodes. Cela signifie l'utilisation de corpus partagés, de sorte que les progrès dans le domaine peuvent être comparés de manière significative. Cela fait également référence à des concours et à des forums d'évaluation où des groupes de recherche du monde entier s'affrontent sur un problème particulier pour l'argent ou le prestige. En règle général, la performance des techniques d'évaluation automatique est mesurée en termes d'accord avec les scores attribués par l'homme (souvent la moyenne de plusieurs scores humains). Diverses mesures de corrélation ont été utilisées pour mesurer quantitativement l'étendue de l'accord. Nous utiliserons pour notre part, le coefficient de Pearson et l'erreur quadratique moyenne (RMSE) afin de mesurer la fiabilité des méthodes développées qui seront détaillées dans la partie des résultats.

« The Joint Student Response Analysis and Eighth Recognizing Textual Entailment Challenge» [22] était la première compétition ASAGS à grande échelle et non commerciale.

Les corpus comprenaient des données provenant d'un système de dialogue tutoriel pour la physique de l'école secondaire (Beetle) et des questions de sciences de l'école primaire de la 3e à la 6e année (SciEntsBank). Environ 8 000 réponses d'élèves sont incluses. Un système de notation catégorique à 5 niveaux est défini avec des étiquettes "correctes", "partiellement correctes incomplètes", "contradictoires", "non pertinentes" et "non-domaine". Une des dimensions de données est le degré d'adaptation du domaine requis dans les solutions [23].

iii. Calcul de similarité et documents textuels

La manipulation des documents textuels dans un contexte de traitement automatique de la langue est une tâche complexe voir impossible pour des outils d'apprentissage automatique qui manient uniquement des valeurs numériques. L'utilisation des documents textuels en leur forme brute comporte beaucoup de contraintes notamment le volume de ces derniers. En effet, traiter de grand corpus nécessiterai un temps considérablement élevé et un matériel à la pointe de la technologie.

Dans le cadre de notre recherche, nous allons être contraints, principalement, à calculer la similarité entre textes en se basant sur de grands corpus de textes aussi. Pour cela, nous passons en revu plusieurs algorithmes de manipulation de documents textuels adaptés au calcul de similarité notamment : le modèle d'espace vectoriel et les vecteurs sémantiques.

1. Modèle d'espace vectoriel

Aussi appelé « vecteur de terme », le modèle d'espace vectoriel est un modèle mathématique pour représenter des documents textuels en tant que vecteurs d'identificateurs, comme des termes ou des jetons (tokens). Le terme dépend de ce qui est comparé mais sont normalement des mots simples, des mots-clés, ou des phrases. Une collection de documents composée de documents Y, indexée par Z termes, peut être représentée dans une matrice YxZ (figure II.4). Ainsi, les requêtes (textes à évaluer ou comparer) et les documents sont représentés comme des vecteurs et chaque dimension correspond à un terme distinct, autrement dit, les mots sont considérés comme indépendants et l'ordre est sans importance. Chaque élément de la matrice M est un poids pour le terme Z dans le document Y. Si le terme apparaît dans le document, la valeur dans la matrice pour l'élément spécifique change, sinon non. La variation de cette valeur dépend de l'importance du mot dans le document et peut être calculée par des mesures statistiques telles que la fréquence inverse du mot dans le document (IDF) que nous verrons dans le prochain chapitre ainsi que d'autre mesures de pondération.

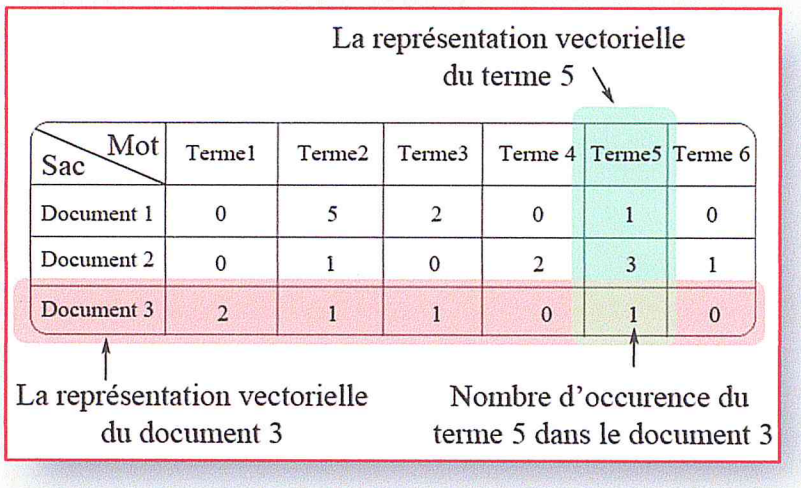

Figure II.4 Exemple de modèle vectoriel

En utilisant ceci conjointement avec l'hypothèse de la théorie des similarités de documents, la similarité entre deux documents peut être calculée en comparant la différence entre les angles de chaque vecteur de document et le vecteur de requête. Ce calcul aboutit à un résultat allant de 0 à 1, les deux nombres étant inclus. Si le document et le vecteur de requête sont orthogonaux, le résultat est 0 et il n'y aura pas de correspondance, le terme de requête n'existe pas dans le document. Si le résultat est 1, cela signifie que les deux vecteurs sont égaux. Pour calculer ces valeurs, de nombreuses manières ont été trouvées, nous en verrons quelques unes quand nous présenterons les mesures de similarité.

2. Vecteurs sémantiques [24]

Contrairement au modèle d'espace vectoriel où le positionnement des mots n'est pas pris en compte, ici l'idée consiste à déterminer la sémantique d'un mot en consultant les autres termes utilisés à ses côtés dans des phrases. Evidemment, ceci doit être appliqué sur un corpus significatif comportant un nombre important de mots. Le plus difficile est d'obtenir de tels vecteurs. Il faut donc construire un ensemble de vecteurs pour chaque mot dans le dictionnaire utilisé. Les vecteurs sont définis dans un espace vectoriel orthogonal à n dimensions (figure II.5) où chaque base se voit attribuer un mot de vocabulaire unique (donc chaque entrée du dictionnaire a une base dans l'espace vectoriel). Pour chaque mot du dictionnaire, on détermine un vecteur dans cet espace, où la composante du vecteur pour chaque base est le nombre d'occurrences du mot dans la base qui le représente où il apparaît dans le contexte du mot pour lequel un vecteur a été construit. Le mot "Contexte" ici peut être vu au sens large.

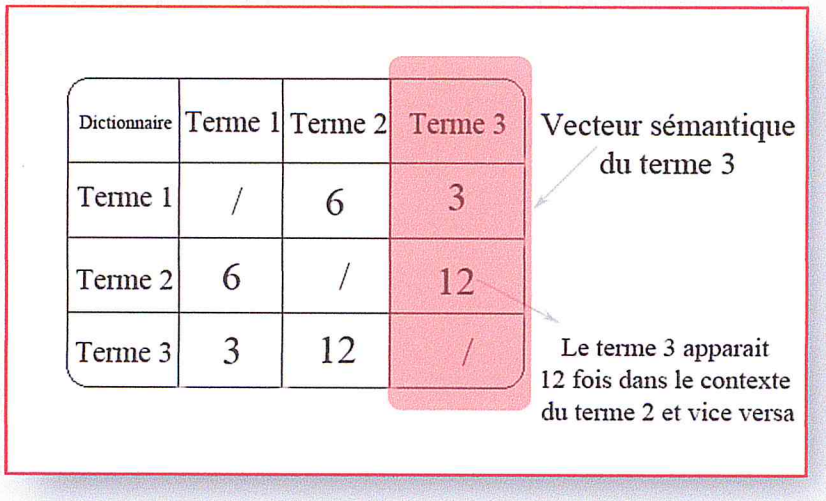

Figure II.5 Exemple de vecteur sémantique

3. Exemple d'utilisation :

Une fois les documents textuels ont été convertis vers une représentation mathématique comme l'un des deux formats que nous venons de voir, plusieurs processus de traitement automatique de la langue peuvent leur être appliqués directement. Si nous voulons par exemple mesurer la similarité entre deux mots dont nous avons les représentations vectorielles, il nous suffit d'appliquer une mesure de calcul vectoriel tel que le la similarité cosinus [24] pour calculer l'angle entre les deux:

$$SimCos(vMot1, vMot2) = \frac{\overrightarrow{vMot1} \,.\, \overrightarrow{vMot2}}{\|\overrightarrow{vMot1}\| . \|\overrightarrow{vMot2}\|}$$

Pour une meilleure compréhension, la figure II.6 illustre l'opération sur un plan 2D :

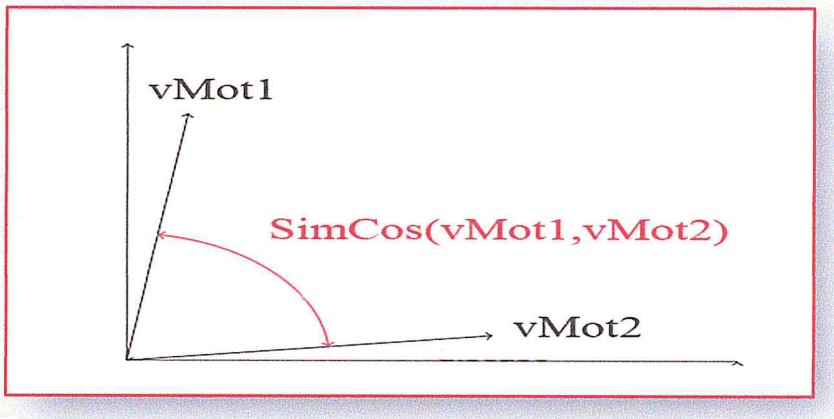

Figure II.6 Similarité cosinus entre deux vecteurs de mots

iv. Les approches de mesure de similarité

La détection de plagiat, l'analyse de données textuelles, l'extraction d'information, l'évaluation automatique des réponses et bien d'autres disciplines issues du traitement automatique de la langue partagent tous la problématique du calcul de similarité. « Les mesures de similarité mappent la distance ou la similarité entre la description symbolique de deux objets en une seule valeur numérique, qui dépend de deux facteurs, les propriétés des deux objets et la mesure elle-même » [25]. Les objets ici sont des mots. Les mots peuvent être similaires de deux manières : lexicalement et sémantiquement. Ils sont similaires lexicalement ou syntaxiquement s'ils partagent une séquence de caractères et ils sont similaires sémantiquement s'ils ont le même sens, sont opposés l'un à l'autre, s'utilisent de la même manière, s'utilisent dans le même contexte ou l'un est un type d'un autre. Concernant les mesures de similarités on retrouve dans la littérature 3 grandes approches :

1. Les approches syntaxiques

« La similarité syntaxique est une métrique qui mesure la similarité ou la dissimilarité entre deux chaînes de caractères. Par exemple, les chaînes de caractères "Sam" et "Samuel" peuvent être considérées comme similaires. Une telle mesure sur les chaines de caractères fournit une valeur obtenue algorithmiquement. » [24]. L'utilisation d'une telle approche est recommandée là où chaque détail est important et doit être pris en compte. Par exemple, dans le cas d'un test où le but est d'évaluer les compétences d'écriture de l'apprenant c'est-à-dire l'évaluation de l'écrit, une mesure de syntaxe est appropriée car tout a un impact : variations graphiques (majuscules, espacements …), ponctuations, fautes d'orthographe, fautes de grammaire etc. Cependant, les approches syntaxiques ne prennent pas en compte la sémantique ce qui rend leurs utilisations dans certain contextes critiques.

La figure II.7 recense les mesures syntaxiques les plus utilisées, on retrouve notamment: la distance de Levenshtein, l'indice de Jaccard, la distance euclidienne ou encore le coefficient de Dice qui mesure la similarité entre deux proposition p1 et p2 en se basant sur le nombre de termes communs à p1 et p2 : [24]

$$\text{SimDice}(p1, p2) = \frac{2N_c}{N_1+N_2}$$

où N c'est le nombre de termes communs à p1 et p2, et N1 (resp. N2) est le nombre de termes de p1 (resp. p2).

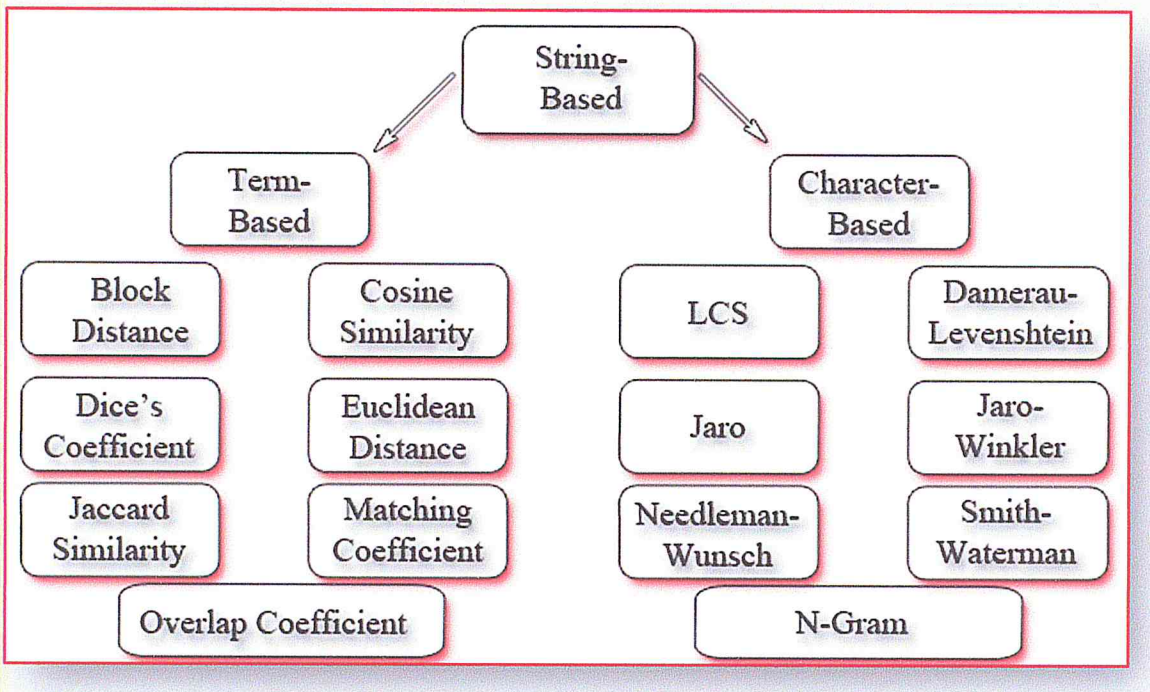

Figure II.7 Quelques mesures de similarité syntaxique [26]

2. Les approches sémantiques

« La similarité sémantique est un concept selon lequel un ensemble de documents ou de termes se voient attribuer une métrique basée sur la ressemblance de leur signification ou contenu sémantique. » [24]. Ces approches sont les meilleurs pour ce qui est de capturer le sens des textes. Dans un cadre où le but est d'évaluer les compétences d'un individu dans un domaine précis (évaluation ***par*** écrit), les mesures de similarité sémantique excellent remarquablement par rapport aux mesures syntaxiques. Nous distinguons deux sous catégories :

2.1. Similarité basée corpus

Du terme anglais « Corpus-Based Similarity », la similarité basée corpus est un ensemble de mesures de similarité sémantique qui déterminent la similitude entre les mots et donc entre les textes en fonction de l'information obtenue depuis de grands corpus. Un Corpus est une grande collection de textes écrits ou parlés utilisés pour la recherche linguistique. La figure II.8 montre les mesures de similarité basées corpus les plus répondues.

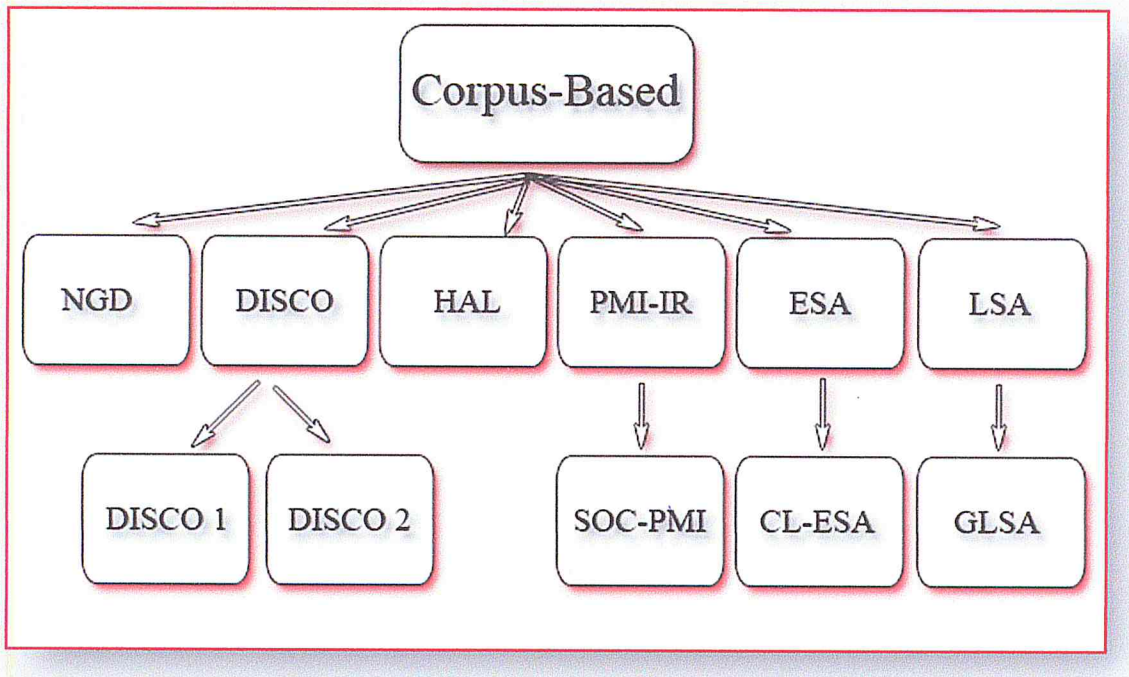

Figure II.8 Quelques mesures de similarité sémantique basée corpus [26]

2.2. Similarité basée connaissance

Du terme anglais « Knowledge-Based Similarity », la similarité fondée sur la connaissance est l'une des mesures de similarité sémantique qui repose sur l'identification du degré de similitude entre les mots à l'aide d'informations dérivées de ressources externes tel que le réseau WordNet ou les dictionnaires de synonymes. La figure II.9 comporte quelques mesures de similarité basée connaissance. Nous remarquons que les mesures sont classées dans deux sous branches : les mesures de similarité sémantique et les mesures de relation sémantique, ces dernières couvrent un plus large éventail de relations entre concepts qui inclut des relations de similarité supplémentaire telles que : « est un genre de », « est un exemple spécifique de », « est une partie de », « est le contraire de », …

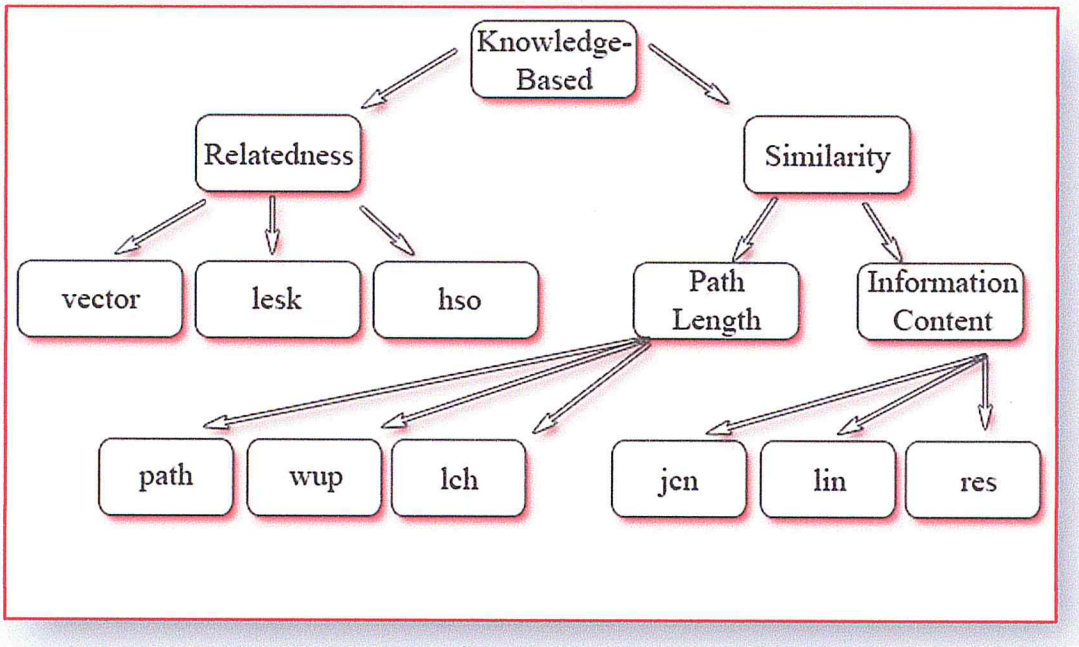

Figure II.9 Quelques mesures de similarité sémantique basée connaissance [26]

3. Les approches hybrides

Les méthodes hybrides utilisent des mesures de similarité multiples, soit en combinant des méthodes sémantiques et syntaxiques à la fois où bien plusieurs méthodes de même type. De nombreuses recherches ont couvert ce domaine et ont démontré qu'une telle approche est nettement plus performante que les approches individuelles. On peut citer pour l'exemple l'expérience de Mihalcea [27] où huit mesures de similarité sémantique ont été testées, deux de ces mesures étaient des mesures fondées sur des corpus et les six autres étaient basées sur des connaissances. Premièrement, ces huit algorithmes ont été évalués séparément, puis ils ont été combinés ensemble. La meilleure performance a été obtenue en utilisant une méthode qui combine plusieurs métriques de similarité en une seule. Une autre recherche concernant les approches hybrides est celle Islam & al. [28] qui ont présenté une méthode et l'ont appelée similitude de texte sémantique (STS). Cette méthode détermine la similarité de deux textes grâce à une combinaison entre des informations sémantiques et syntaxiques. Ils ont considéré deux fonctions obligatoires (similarité de chaîne et similarité de mots sémantique) et une fonction optionnelle (similarité d'ordre de mots communs). La méthode STS a obtenu un très bon coefficient de corrélation de Pearson pour 30 paires de données.

V. Les Word Embeddings

1. Définitions et généralités

Le traitement des documents textuels est une tâche qui nécessite un prétraitement de ces derniers les faisant passer de leur format brut à une représentation mathématique pouvant être manipulée par des algorithmes d'apprentissage automatique. Nous avons déjà vu dans la section calcul de similarité et documents textuels, une des représentations possibles qui est le modèle d'espace vectoriel. Cependant, malgré son rendement efficace, cette représentation souffre de deux faiblesses principales : Premièrement, l'ordre des mots est perdu, ce qui fait que plusieurs phrases peuvent avoir la même représentation du moment qu'elles sont composés par les mêmes mots. Deuxièmement, la sémantique n'est pas prise en compte, deux mots similaires sémantiquement n'auront pas forcement une représentation similaire. De plus, étant donné que les termes sont traités d'une façon atomique, les collocations de mots telles que « Air Algérie » ou « New York Times » perdent leur signification réelle qui laissent place à la signification consensuelle de mots qui les composent. Ces faiblesses ont poussé les chercheurs à s'investir dans d'autres techniques de représentation et une des plus performantes aujourd'hui qui est les Word Embeddings.(WE)

Aussi appelée représentation distribuée des mots, les WE caractérisent chaque mot par un ou plusieurs vecteurs denses, de faible dimension ayant des éléments réels, capturant les spécificités latente (de contexte) du mot et les propriétés syntaxiques et sémantiques utiles.

Il existe deux types de Word embeddings :

- Ceux basés sur la fréquence : qui peut faire référence au nombre d'occurrences du mot, son importance estimé avec une approche statistique ou bien par un vecteur de cooccurrence.

- Ceux basés sur la prédiction : et qui ont révolutionné le domaine du TAL surtout après l'introduction du modèle Word2vec par Mitolov [29]. Dans ces modèles, chaque mot est représenté par un vecteur qui est concaténé ou moyenné avec d'autres vecteurs des mots du contexte, et le vecteur résultant est utilisé pour prédire d'autres mots du contexte grâce à un calcul de probabilité.

Word2Vec est un modèle de représentation distribué des mots peu profond (par rapport à un réseau de neurone traditionnel) où le principe est de générer des vecteurs dimensionnels à partir d'un apprentissage sur des données d'entrée non étiquetées. En fait, il prédit les mots en fonction de leur contexte en utilisant l'un des deux modèles neuronaux distincts: **CBOW** et **Skip-Gram**.

2. Continuous Bag-of-Words « CBOW »

Le modèle « CBOW » prédit un mot courant basé sur son contexte. Le contexte correspond à un certain nombre de mots voisins à gauche et à droite du mot. Dans le processus CBOW, trois couches sont utilisées comme nous pouvons le voir sur la figure II.10. La couche d'entrée correspond au contexte. La couche cachée correspond à la projection de chaque mot de la couche d'entrée dans la matrice de poids qui est projetée dans la troisième couche qui est la couche de sortie. La dernière étape de ce modèle est la comparaison entre sa sortie et le mot lui-même afin de corriger sa représentation en fonction de la propagation arrière du gradient d'erreur. Ainsi, le but du réseau de neurones CBOW est de maximiser l'équation suivante [30] :

$$\frac{1}{V}\sum_{t=1}^{V} \log p(m_t | m_t - \frac{c}{2} \cdots m_t + \frac{c}{2})$$

où V (et C) est la taille du vocabulaire (contexte).

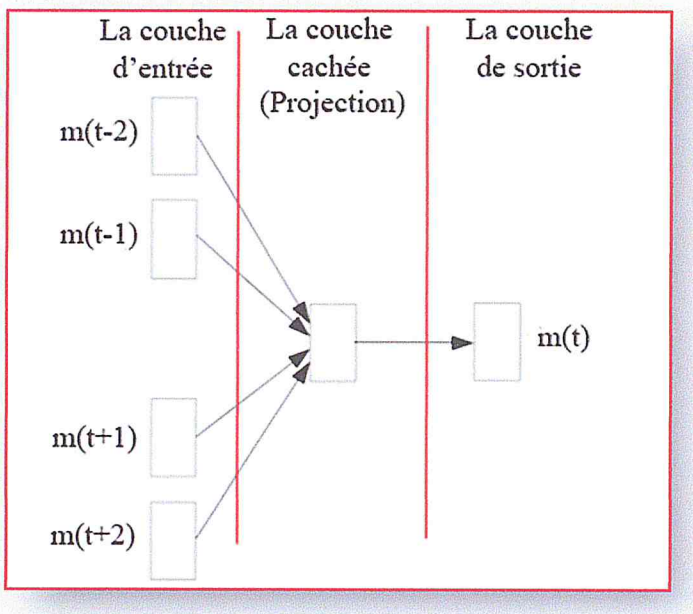

Figure II.10 Schéma du modèle CBOW [31]

3. Skip-Gram « SG »

Skip-Gram: c'est le contraire des modèles CBOW. En effet, la couche d'entrée correspond au mot cible et la couche de sortie correspond au contexte comme on peut le voir sur la figure II.11. Ainsi, Skip-Gram cherche la prédiction du contexte d'un mot donné au lieu de la prédiction d'un mot sachant son contexte comme CBOW. La dernière étape de Skip-Gram est la comparaison entre sa sortie et chaque mot du contexte afin de corriger sa représentation en fonction de la propagation arrière du gradient d'erreur. En fait, il cherche la maximisation de l'équation suivante [30]:

$$\frac{1}{V}\sum_{t=1}^{V}\sum_{j=t-c,j\neq t}^{t+c}\log p(m_j|m_t)$$

où V (et C) est la taille du vocabulaire (contexte).

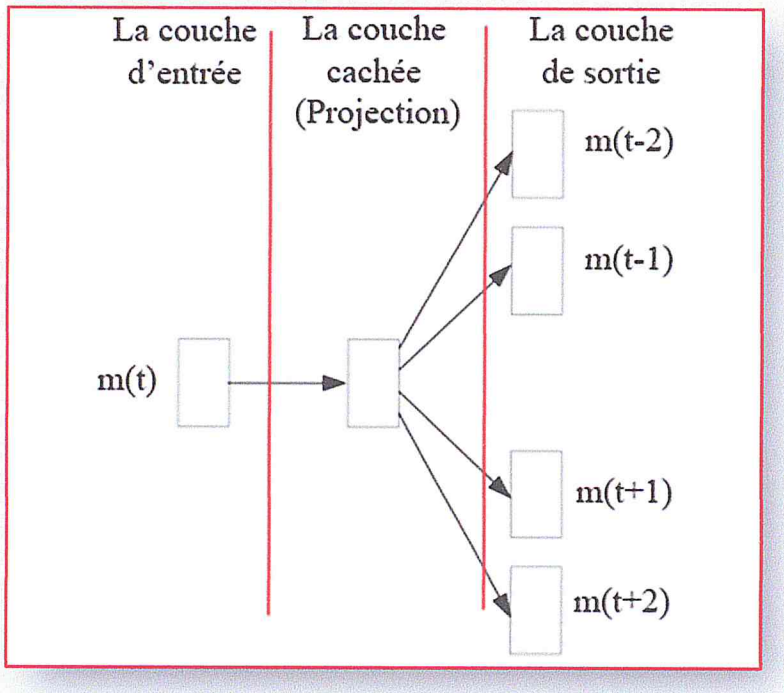

Figure II.11 Schéma du modèle Skip-Gram [31]

Chacun de ces modèles a son propre avantage. À titre d'exemple, Skip-Gram est plus efficace avec de petites données d'entraînement. De plus, les mots peu fréquents sont bien présentés. D'un autre côté, CBOW fonctionne bien avec des mots fréquents.

vi. La langue arabe et le traitement automatique de la langue

1. Les enjeux de la langue arabe dans le contexte de l'évaluation automatique [32]

La langue arabe est parlée et écrite par plus de 300 millions de personnes dans plus de vingt pays du monde entier. L'application des tâches de NLP (Natural Language Processing) en général et dans l'évaluation automatique des réponses courtes en particulier est très difficile en langue arabe. La langue arabe a beaucoup de caractéristiques, qui sont considérées comme des enjeux (défis) à soulever pour l'évaluation automatique :

Le premier enjeu est qu'il existe trois types de langue arabe, connus sous le nom de classique, moderne et familier. L'arabe classique, qui est utilisé dans le Coran, est plus complexe dans sa grammaire et son vocabulaire que l'arabe moderne. Il a un grand nombre de signes diacritiques qui facilitent la prononciation et la détection des mots dans leurs cas grammaticaux. Le deuxième type est l'arabe moderne, tous les signes diacritiques ont été omis pour faciliter et accélérer le processus de lecture et d'écriture. Ce type est considéré comme la langue officielle des pays arabes et est utilisé dans la langue de tous les jours, dans l'éducation et dans les médias. Habituellement, les recherches arabes basées sur l'arabe utilisent l'arabe moderne. En arabe parlé (dit aussi familier), qui est le troisième type, la grammaire et le vocabulaire sont moins sophistiqués par rapport à l'arabe moderne. Cependant, la plupart des gens l'utilisent dans leurs conversations quotidiennes et dans des lettres écrites de manière informelle en raison de sa simplicité. Les arabes font beaucoup d'erreurs dans la grammaire quand ils utilisent l'arabe moderne et ils mélangent entre l'arabe moderne et l'arabe familier.

Le deuxième enjeu est la morphologie arabe. La langue arabe est complexe en raison de la variation morphologique. La forme des lettres change en fonction de leur position dans le mot. De plus, le mot peut être constitué de préfixes, de lemmes et de suffixes dans des combinaisons différentes, ce qui aboutit à une morphologie très compliquée.

Le troisième enjeu est la capitalisation. La langue arabe ne supporte pas la capitalisation de noms propres tels que les noms de pays, les noms de personnes. Considérant que, dans les langues latines, ceux-ci commencent par une lettre majuscule. L'évaluation automatique de

texte en arabe peut ne pas reconnaître ces entités nommées, ce qui augmente la difficulté de détecter de tels noms dans les réponses en arabe.

Le dernier enjeu et que nous considérons le plus important est celui lié au manque de ressources linguistiques (outils NLP, Corpus, Datasets, …). Généralement, il y a une limitation sur le nombre de ressources linguistiques arabes, qui sont disponibles gratuitement à des fins de recherche. Plus récemment, un certain nombre de corpus arabes ont été développés; Cependant, peu d'amélioration globale de la situation globale a été observée [33].

Les défis précédents doivent être résolus lors de la construction d'un système pour l'évaluation automatique des réponses courtes. Nous les reprenons dans la discussion des travaux de similarité utilisant la langue arabe. Ces travaux ne concernent pas directement l'évaluation automatique des réponses courtes (que nous reprenons dans les travaux connexes dans la section suivante) mais nous donnent des indications sur l'utilisation de mesures de similarité dans le contexte de la langue arabe et nous permettent de confirmer ou d'infirmer certains résultats ou constatations.

2. Les travaux sur la similarité de textes utilisant la langue arabe [32]

2.1. Similarités syntaxiques

Pour la langue arabe, de nombreux chercheurs ont utilisé l'algorithme de distance de Levenshtein dont [34] qui l'a utilisé pour développer l'outil de vérification orthographique pour les mots arabes. Cependant, Levenshtein ne donne pas de résultats précis lorsqu'il est appliqué sur la langue arabe selon les auteurs.

Le travail de [35] a utilisé une méthode N-gram pour convertir un mot en une suite de N-grammes et l'appliquer dans le contexte des systèmes de recherche textuelle arabes. L'étude indique que l'approche N-gram ne semble pas fournir une approche efficace dans le contexte arabe. [36] a étudié les différentes mesures de similarité syntaxiques dans la recherche d'information arabe et la mesure de similarité Cosinus (appelée souvent Cosine) est la meilleure mesure par rapport à d'autres mesures: coefficient de Dice, coefficient de Jaccard, coefficient de similarité d'inclusion, Mesure du coefficient de chevauchement, mesure de distance euclidienne et mesure de distance de Manhattan.

[37] ont conçu un thésaurus arabe automatique en utilisant la similarité terme-terme. Ils ont comparé la mesure de similarité de Jaccard avec d'autres mesures telles que Cosine et Dice.

Les résultats indiquent que les mesures de similarité de Jaccard et de Dice ont la même performance, alors que le Cosinus est légèrement plus efficace que les mesures de Jaccard et de Dice.

2.2. Similarités sémantiques

L'arabe est une langue mal adaptée pour les approches basées sur les corpus par rapport à l'anglais, car il y a un manque de données, ce qui affecte négativement la recherche sur les approches sémantiques basées sur les corpus en arabe. [38] ont passé en revue quatorze corpus arabes et les ont catégorisés par leur langue cible, objet, date du texte, lieu, domaine de texte, représentativité, mode de texte, taille. Plusieurs de ces corpus ne fournissent aucune information concernant la période couverte par les textes. De plus, pour tous les corpus, les textes constitutifs ne sont pas classés en fonction de leurs dates ou de la période à laquelle ils appartiennent; il y a donc une limite à l'utilisabilité du corpus et une difficulté à comparer les variétés de la langue utilisées à différentes périodes, et à observer comment la langue arabe a évolué.

Pour les approches basées sur la connaissance, WordNet est utilisé dans divers domaines tels que la recherche d'information et la similarité sémantique, En raison du succès de WordNet dans les applications en anglais, plusieurs projets sont actuellement menés pour développer WordNet pour d'autres langues. WordNet arabe (AWN) a été développé en utilisant la même méthodologie qu'EuroWordNet. Il se compose de 11 270 synsets et contient 23 496 expressions arabes (mots et multi-mots). Les principales limitations de l'AWN actuel sont un manque d'informations et de concepts par rapport à WordNet en anglais, et quelques relations sémantiques entre les synsets. De nombreux concepts arabes n'ont pas été inclus dans la base de données AWN. Cette limitation constitue un obstacle majeur à l'utilisation d'AWN en tant que source d'approches basées sur la connaissance. AWN pourrait être amélioré et étendu par plusieurs approches différentes, par exemple l'ajout de nouveaux synsets,…Par conséquent, nous pensons que l'approche de similarité sémantique utilisant AWN nécessite des recherches supplémentaires afin d'être plus fiable et plus mûre. Nous la jugeons insuffisante dans le contexte de l'évaluation automatique et c'est pour cette raison que nous l'avons écarté momentanément de nos travaux.

vii. Les travaux connexes à notre recherche [32]

Les travaux que nous menons dans le cadre d'une approche hybride qui permet de combiner plusieurs approches syntaxiques et sémantiques (particulièrement basés sur le corpus) est celle basées sur les Word Embedding. Dans ce contexte notre travail est connexe aux travaux menés par Gomaa & al [39] [40]. Les auteurs ont utilisé des mesures de similarité syntaxiques et des mesures basées sur le corpus pour développer leur système de notation à réponse courte. Ils ont testé les mesures sur le dataset(GOMAA Dataset) qu'ils ont construit eux-mêmes. Leurs résultats ont montré que les meilleures valeurs de corrélation obtenues en utilisant des mesures syntaxiques ont été obtenues en utilisant respectivement les approches de distance de n-gramme et de distance de Manhattan. Dans la deuxième étape, ils ont mesuré la similarité en utilisant des mesures de similarité basées sur le corpus [41]: la mesure DISCO1 (Calcule la similarité du premier ordre entre deux mots basés sur leurs ensembles de collocation) et la mesure DISCO2 (Calcule la similarité du second ordre entre deux mots basés sur leurs ensembles de distribution des mots similaires). Les résultats ont montré que DISCO1 atteint des valeurs de corrélation plus efficaces. Dans la troisième étape, la similarité a été évaluée en combinant des mesures basées sur la syntaxe et le corpus. La meilleure valeur de corrélation a été obtenue en combinant n-gramme avec les techniques de similarité DISCO1.

Le travail de [43] et [44] sont tout aussi intéressants en considérant leurs résultats par rapport à une approche basée sur le calcul vectoriel et les word Embedding. [43] ont évalué leur approche sur Gomaa Dataset alors que [44] a obtenu le 2ème meilleur score du SemEval 2017 d'où l'intérêt que nous portons pour ces travaux utilisant les mêmes DataSets que nous. Ces Datasets vont être décrits et le contenu détaillé dans la section « acquisition des dataset » du prochain chapitre.

Pour ce qui est de notre travail, en utilisant le SemEval Dataset, nous allons avoir une indication sur la généralisation de nos approches dans des domaines connexes en le comparant aux résultats de la compétition 2017 fournis dans [42].

Conclusion

Nous venons de faire un tour de littérature où nous avons présenté les principaux domaines et approches chevauchant notre projet ainsi que les définitions nécessaires pour la compréhension de la suite. En considérant de plus les travaux liés à la langue arabe en termes de similarité et d'évaluation automatique ainsi que les enjeux posés par la langue arabe, nous tentons dans la suite de notre travail d'améliorer les résultats obtenus en développant nos propres modèles pour atteindre une meilleure hybridation en termes de maximisation du Coefficient de Pearson(CP) et de minimisation de l'erreur quadratique moyenne(RMSE). Les deux métriques d'évaluation CP & RMSE sont décrites dans la section « Calcul des notes et évaluation» dans la partie mise en œuvre du système du chapitre 3.

Il est à noter que nous avons pris le choix d'éviter la description des datasets et des métriques d'évaluation dans l'état de l'art dans la mesure où l'acquisition des datasets et l'évaluation constituent des modules à proprement dits inclus dans le vif de notre système développé.

III. Développement du système d'évaluation automatique

Ce chapitre traite le noyau de notre travail. L'approche suivie pour évaluer l'impact des Word Embeddings sur la langue arabe est explicitée sur plusieurs parties commençant par une description de l'idée générale, puis les différentes étapes de réalisation comme l'acquisition des ressources et l'implémentation de nos outils.

i. Approche méthodologique

Dans ce travail, nous considérons un mécanisme basé sur le Stemming afin d'analyser l'impact sur l'évaluation automatique des questions à réponses courtes en langue arabe. En effet, il est très difficile de mettre en œuvre les mécanismes d'évaluation automatique pour la langue arabe, en raison de sa nature complexe, étant très flexionnelle et ambigüe en l'absence de signes diacritiques. Il n'y a eu que peu de tentatives de recherche sur ce sujet, et jusqu'à présent, aucun d'entre eux n'a été en mesure de fournir un système d'évaluation automatique entièrement fonctionnel. Les techniques de stemming ont été exploitées en combinaison avec les mesures de similarités que nous avons développées. Un algorithme de stemming peut être défini comme la procédure de réduction de tous les mots qui partagent la même racine à une forme commune [45].

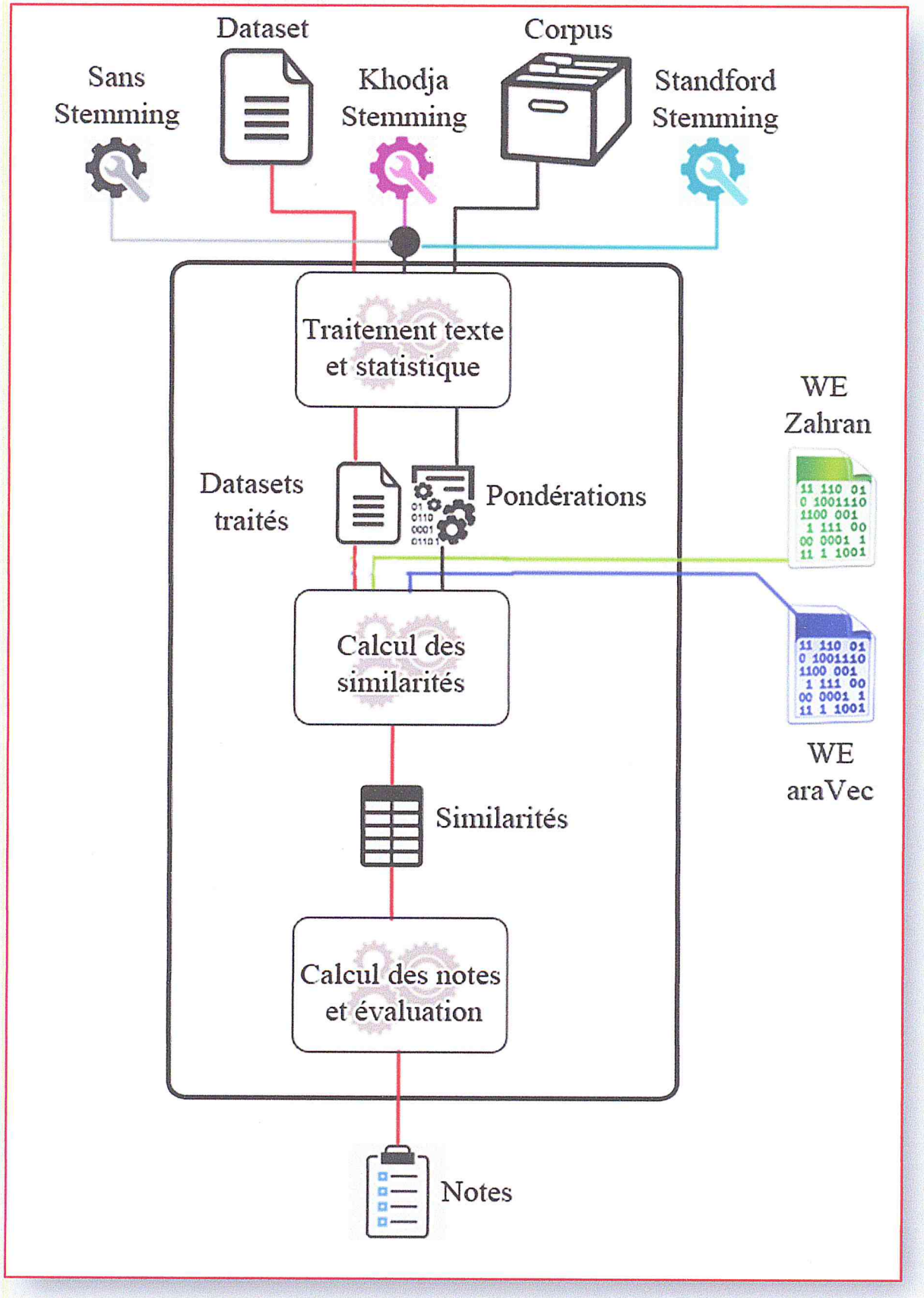

Figure III.1 Approche méthodologique dans le développement du système

Pour toutes les approches de similarités développées nous avons considéré les 3 cas suivants représentés dans la figure III.1:

1. Aucune technique de stemming n'est considérée aux deux réponses (Réponse de l'étudiant et la réponse Modèle) à comparer et qui sont considérées dans leur nature brute. Cette technique est représentée par la couleur noire sur le schéma.
2. Une technique de stemming lourde (Heavy Stemming) est appliquée aux réponses à comparer. Le streaming lourd, également appelé « Root-Stemming » (Stemming à la racine), consiste à supprimer les préfixes et les suffixes bien connus pour extraire la racine réelle d'un mot et à identifier le motif en correspondance avec le mot restant. La couleur violète illustre cette méthode.
3. Une technique de stemming légère (Light Stemming) est appliquée aux réponses à comparer. Le streaming léger est un processus moins complexe, où le stemming est arrêté sur la suppression des préfixes et des suffixes, sans tenter d'identifier la racine réelle du mot. Cette dernière technique est identifiée par la couleur bleue claire.

Il est utile de noter qu'hormis les effets de l'approche appliquée sur les données, le processus est exactement identique pour les 3 approches. Aussi, les méthodes de calcul de similarité que nous avons développées et implémentées nécessitent des ressources telles que les WEs ou les corpus qu'il faut acquérir avant de faire recours à ces dernières.

Basé sur le pipeline de développement des systèmes ASAG que nous avons vu précédemment, les étapes suivantes ont été appliquées pour chaque approche :

- Formatage des datasets selon l'approche appliquée. Cette tâche est réalisée au niveau du module « traitement des textes et statistiques ».
- Préparation des données statistiques pour le calcul de similarité. Ces mesures doivent être issus de corpus ayant été traité par la même approche appliqué aux datasets. Cette tâche est également réalisée par module « traitement des textes et statistiques ».
- Calcul des similarités : Pour chaque approche, tous les modules de calcul de similarité que nous avons développés sont exécutés avec les ressources adéquates à l'approche. Le module « calcul des similarités » est chargé de cette tâche.
- Calcul des scores et évaluation, cette tâche est affectée au module portant le même nom « calcul de score et évaluation».

La synthèse que nous avons réalisée sera décrite en détaille dans le chapitre 4.

ii. Mise en œuvre du système :

La démarche étant présentée, nous allons passer aux étapes de réalisation du système :

1. Acquisition des ressources

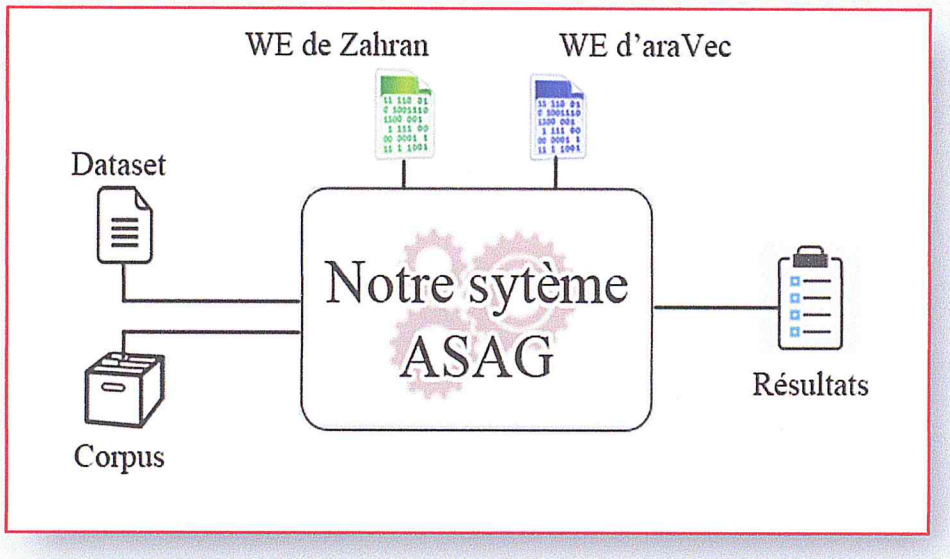

Figure III.2 Ressources nécessaires au système

Si nous considérons notre système comme une sorte de boite noire (Figure III.2), les entrées sont identifiables par : Jeux de données, Word Embeddings et Corpus dont nous allons décrire le processus d'acquisition dans la suite :

1.1. Les jeux de données (Datasets)

L'évaluation assistée par ordinateur est caractérisée par des progrès isolés avec peu de capacités à comparer les approches et à s'appuyer sur le travail des autres chercheurs particulièrement quand nous considérons la langue arabe. Il n'existe pas à ce jour des ensembles de données publiquement disponibles pour comparer efficacement deux systèmes côte à côte. En ce qui concerne la langue arabe il existe un seul DataSet [39] [40] largement cité dans l'évaluation des ASAGS en langue arabe et que les auteurs ont accepté de nous transmettre. Dans toute la suite nous allons considérer ce dataset et l'identifier par « GOMAA Dataset ».

Nous avons donc effectué le test de nos différentes approches sur ce dataset dans le but de comparer nos résultats par rapport à d'autres travaux ayant utilisé ce même dataSet.

- **Gomaa Dataset:**

Les questions présentées dans le dataset couvrent un chapitre du programme d'études égyptien officiel pour le cours de sciences de l'environnement (ES), qui représente 25% du programme global. L'ensemble de données contient 61 questions, 10 réponses pour chacune, avec un nombre total de 610 réponses. La longueur moyenne de la réponse d'un étudiant est de 2,2 phrases, 20 mots ou 103 caractères. L'ensemble de données contient une collection de réponses et notes des élèves, notées par deux annotateurs experts humains qui ont donné des notes entre 0 et 5 et obtenu un coefficient de corrélation de Pearson (r) et une erreur quadratique moyenne (RMSE) de **0,86 et 0,69**, respectivement entre les deux annotateurs. Dans toute évaluation par rapport à ce dataset, l'idéal est d'approcher le plus possible ces valeurs. Nous disposons de la version XML du dataset qui nous a été fournie par les auteurs. Le tableau suivant représente des exemples de questions, des réponses modèles et des réponses courtes fournies par deux étudiants, et des notes attribuées manuellement par deux experts humains.

Question	Réponse modèle	Réponses des apprenants	Notes manuelles
عرف مصطلح الإيكولوجيا	الدراسة التى تتناول جوانب الطبيعة بما يحدده حياة الكائن و كيفية استخدامه لمكونات البيئة	الدراسة التى تتناول مكونات البيئة و استخدام الإنسان لها	3.5
		هو العلم الذى يتناول كل ما له علاقة بالأرض من حيث مكوناتها وحركتها و تاريخها و الظواهر التى تحدث عليها	2.5
		هو العلم الذى يتناول كل ما له علاقة بالأرض من حيث مكوناتها وحركتها و تاريخها و الظواهر التى تحدث عليها	1
اشرح بيئة الإنسان	الإطار الذى يحيا فيه الإنسان مع غيره من الكائنات الحية و يحصل منها على مقومات حياته	هى الإطار الذى يحيا الإنسان فيه مع غيره من الكائنات الحية و يحصل منها على مقومات حياته	5
		الحيز الذى يحيط بالإنسان مع الكائنات الحية الأخرى الذى يستفيد منها للقدرة على العيش	3.5
		كل ما يحيط بالإنسان من مكونات حية أو غير حية يؤثر فيها و يتأثر بها	1.5

Tableau III-1 Aperçu du dataset de Gomaa

- **SemEval Datasets :**

Afin d'évaluer l'applicabilité et la généralisation des techniques utilisées dans notre système à d'autres domaines connexes, nous avons utilisé des ensembles de données supplémentaires qui ont été largement utilisés dans le domaine de la similarité du texte, de l'implication textuelle et de la paraphrase dans le cadre du « Semantic Evaluation(SemEval) workshop for Semantic Textual Similarity(STS) » ; une compétition qui se déroule chaque année depuis 2012. Nous avons profité du SEMEval 2017(composé de 6 tracks) [46] qui a introduit dans son « Track 1 », dédié aux couples de textes courts « arabe- arabe », plusieurs DataSets de tests en langue arabe. Nous avons choisi parmi les datasets 2 datasets à savoir :

Le **STS 250 SemEval 2017** : dataset d'évaluation des travaux en compétition dans le track 1.

Le **MSRvid 368 SemEval 2017** : dataset proposé pour le training des données du Track 1 et que nous avons exploité pour l'évaluation des approches.

STS est l'évaluation de paires de phrases en fonction de leur degré de similarité sémantique. La tâche implique de produire des scores de similarité à valeur réelle pour les paires de phrases. La performance est mesurée par la corrélation de Pearson des scores de machine avec des jugements humains. L'échelle ordinale guide l'annotation humaine, allant de 0 pour un chevauchement sans signification à 5 pour l'équivalence de sens. Les valeurs intermédiaires reflètent des niveaux interprétables de recouvrement partiel de sens. Les données arabes sont produites en traduisant un sous-ensemble des données anglaises et en transférant les scores de similarité. Le corpus SNLI (Stanford Natural Language Inference) [47] est la principale source de données des deux datasets. Les phrases sont traduites indépendamment de leurs paires. La traduction en arabe est assurée par le CMU-Qatar par des arabophones natifs avec de solides compétences en anglais. Cinq annotations humaines sont collectées par paire. Les scores d'or font la moyenne des cinq annotations individuelles.

Année	Dataset	Nombre de paires	Source
2017	**STS 250 AR**	250	SNLI
2017	**MSRvid 368 AR**	368	Vidéo (speech)

Tableau III-2 Description des deux datasets STS 250 AR et MSRvid 368 AR

Voici quelques exemples des Datasets :

Datasets	Première phrase	Deuxième phrase	Notes
STS 250 AR	تمشي النساء جنبا إلى جنب	هناك فتيات يمشين متجاورات	2.600
STS 250 AR	كلب يعدو عبر العشب	كلب يعدو عبر العشب	5.000
STS 250 AR	هناك عرض عسكري في الخلاء	هناك حشد يشاهد معرضا	1.600
MSRvid 368 AR	أحدهم يقلي لحما	أحدهم يعزف البيانو	0.250
MSRvid 368 AR	رجل يمشي على طول الطريق من خلال البرية	رجل يقشر بصلة	0.750
MSRvid 368 AR	يعزف الرجل غيتاره	رجل يغني في حين يعزف غيتاره	3.000

Tableau III-3 Aperçu des Datasets STS 250 AR et MSRvid 368 AR

1.2. L'acquisition des Word Embeddings

Les word embeddings sont générés grâce un des réseaux de neurones, ce processus fait trainer des corpus de milliards de mots, ce qui nécessite du matériel très performant. En plus, une bonne représentation dépend de beaucoup de paramètres comme nous allons le voir par la suite, ce qui fait qu'énormément de tests doivent être effectués pour aboutir à des vecteurs de qualité. Ces contraintes expliquent en grande partie le manque de telles ressources dans la littérature. L'objectif dans ce travail étant d'évaluer l'impact des Word Embeddings sur le processus d'évaluation en langue arabe, nous avons utilisé les deux seuls modèles de WE déjà générés pour les mots de la langue arabe décrits dans : [48] [49] et disponible sur le web [50] [51]

- **Les Word Embeddings de Zahran**

La représentation des mots dans un espace vectoriel de Zahran a été réalisée grâce aux modèles CBOW et Skip-gram proposés par Mikolov [52]. Ces derniers ont été entrainés sur plusieurs corpus d'arabe moderne :

3. Wikipédia en arabe.
4. Le corpus Gigaword en arabe.

5. Le fil de Press arabe LDC.
6. Wiktionnaire en arabe.
7. Le corpus parallèle ouvert[53, 54].
8. Les definitions des mots arabes dans Arabase [55].
9. MultiUN; qui est la collection de documents traduits des Nations Unies [56].
10. OpenSubtitles 2011,2012,et 2013. Il s'agit d'une collection de sous-titres de films [57].
11. Les textes brut du Coran [58].
12. Un corpus de fichier de localisation KDE4 [59].
13. Une collection de phrases traduites de Tatoeba [54].
14. Khaleej 2004 and Watan 2004 [60].
15. Les corpus BBC et CNN arabe[61].
16. Le corpus Meedan arabe [62].
17. Ksucorpus; Corpus de l'Université King Saud (King Saud University Corpus) [63].
18. Une version texte du livre Zad-Almaad.
19. Le corpus arabe de Microsoft.

Toutes ces ressources ont été combinées ensembles et ont subit plusieurs étapes de nettoyage et de normalisation pour le corpus combiné:

- Nettoyage des caractères bruyants, étiquettes et suppression des signes diacritiques.

- Normalisation des caractères arabes: (أ - إ - آ) à (ا) et (ة) à (ه) .

- Normaliser tous les chiffres numériques au jeton "NUM".

La taille du vocabulaire du corpus compilé est d'environ 6,3 millions d'entrées (unigrammes et bigrammes), et le nombre total de mots est d'environ 5,8 milliards. La formation de ces modèles a nécessité le choix de certains hyper-paramètres affectant les résultats:

- **La taille du vecteur du mot:** La taille de vecteur est un paramètre d'entrée. Quelques centaines d'éléments par vecteur est le choix recommandé. Ce paramètre affecte la performance du modèle, ce qui signifie qu'il est utile de régler ce paramètre selon la tâche voulu.
- **Fenêtre (contexte):** s'agit de la quantité de mots voisins à considérer autour du mot pivot dans la formation du modèle.
- **Softmax hiérarchique (HS):** Softmax hiérarchique est une approximation efficace du calcul de la softmax complète utilisée pour prédire les mots pendant l'entraînement.
- **Négatif:** il s'agit du nombre d'exemples négatifs dans la formation.
- **Seuil de fréquence:** Les mots apparaissant avec une fréquence inférieure à ce seuil seront ignorés.

Les valeurs affectées à ces paramètres lors de l'entrainement des modèles que nous avons utilisés sont :

	CBOW	SKIP-G
Taille du vecteur	300	300
Fenêtre	5	10
HS	Non	Non
Négatif	10	10
Seuil de fréquence (mot)	10	10
Seuil de fréquence (Phrase)	200	200

Tableau III-4 Paramètres des word embeddings de Zahran

- **Les Word Embeddings d'AraVec**

L'équipe d'AraVec à aussi opté pour les deux modèles CBOW et SkipGram pour leur représentation. Le processus suivi est assez similaire à celui de Zahran et il a été détaillé dans leur document de travail [49]. La spécificité d'AraVec réside dans les jeux de données choisies pour l'entrainement. Les deux modèles ont été entrainés sur 3 corpus séparément, obtenu à partir de Twitter, quelques sites internet, et Wikipedia.

Étant donné que l'évaluation automatique des réponses courtes est un domaine qui s'intéresse principalement à des sujets académiques, nous avons utilisé les modèles CBOW et Skip-Gram d'AraVec générés grâce aux articles de wikipedia qui est une encyclopédie rédigée en collaboration par des utilisateurs du monde entier. En tant que ressource, elle fournit plus de 45 millions d'articles catégorisés ciblant 285 langues, y compris l'arabe. L'arabe a été la première langue sémitique à dépasser 100 000 articles dans Wikipedia. La partie arabe de Wikipedia a maintenant plus de 520 000 articles [64]. Pour construire les modèles, le fichier arabe daté de janvier 2017 a été utilisé. Après avoir segmenté les articles en paragraphes 1 800 000 paragraphes ont été générés, chacun représentant un document.

Model Name	Documents (Millions)	Tokens (Millions)	Min Word Freq. Count	Window size	Technique
Wiki-CBOW	1.8	78.9	20	5	CBOW
Wiki-SG					Skip-Gram

Tableau III-5 Description des Word Embeddings d'AraVec

1.3. Acquisition des corpus

Dans l'optique l'optimiser les calculs de similarité, nous avons utilisé des modèles statistiques expliqués plus loin dans ce chapitre. Ces modèles nécessitent, comme ressources, de grands corpus contenant des milliards de mots. Notre travail portant sur la langue arabe, nous avons parcouru la littérature à la recherche de corpus arabes volumineux et représentatifs. Les critiques concernant les manques de ressources linguistiques par rapport à l'arabe se sont confirmés fortement aux vues de la rareté de tels corpus et la qualité de ceux disponibles. Les corpus les plus cités et libres de droit que nous avons utilisé sont : BBC Arabic[65], CNN Arabic[65], Osac[65] et Watan[66], voici leurs caractéristiques :

	BBC Arabic	CNN Arabic	Osac	WATAN
Nombre de mots total	1 860 000	2 241 348	18 183 511	11 120 000
Nombre de mots uniques	106 733	144 460	449 600	400 000
Nombre de documents	4763	5 070	22 429	21 000
Contenus	- Actualités du moyen orient -Actualité du monde -Économie et travaux -Sports -Presse internationale -Science et technologies -Arts et cultures	-Actualités du moyen orient -Actualité du monde -Économie et travaux -Sports -Science et technologies -Arts et cultures -Loisir	-Économie -Histoire -Éducation et famille -Religion -Sports -Sante -Astronomie -Lois -Récits -Recettes de cuisine	-Culture -Religion -Economie -Actualités local -Actualité internationale -Sports

Tableau III-6 Caractéristiques des corpus utilisés

2. Fonctionnement des modules et modèles de similarité

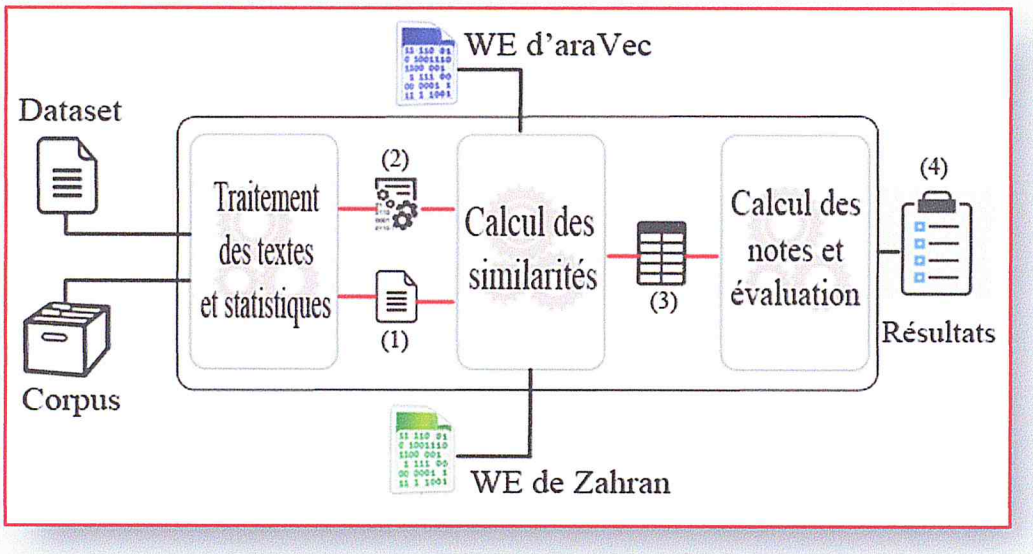

Figure III.3 Architecture fonctionnelle du système

Comme il apparait dans la figure III.3, nous avons représenté notre système par 3 modules principaux, englobant à leur tour d'autres modules comme nous allons le voir en détaillant le mode de fonctionnement et le rôle de chacun :

2.1. Traitement des textes et statistiques

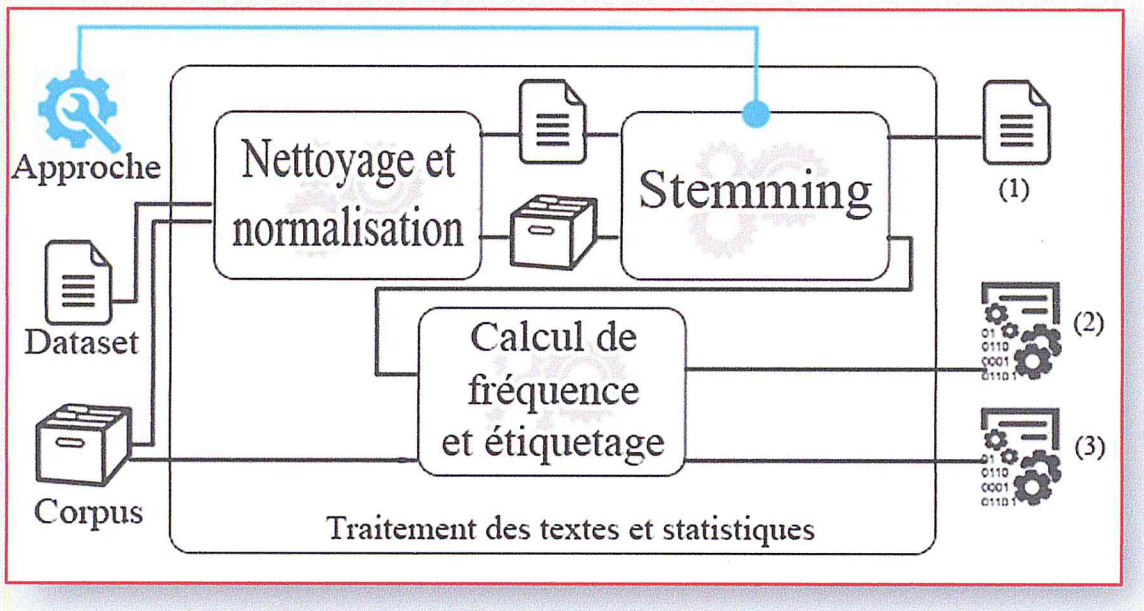

Figure III.4 Composants du module « Traitement des textes et statistiques »

Le module « Traitement des textes et statistique » schématisé par la figure III.4 est un élément clé de notre approche qui assure le formatage des données selon les critères de cette dernière, la cohérence entre le traitement des ressources et l'estimation de l'importance des mots dans la langue arabe. Il comporte 3 sous modules chargés de : la normalisation, le Stemming et le calcul de fréquence et l'étiquetage morphosyntaxique.

2.1.1. La normalisation

Figure III.5 Exemple de normalisation

Toute application traitant des documents textuels comporte un module de prétraitement de ces derniers. Cette étape permet la prise en considération des contraintes liées à la langue. Outre, il est nécessaire de prétraiter les données brutes afin de pouvoir ensuite les traiter avec des processus unifiés et non une multitude de processus adaptés à tous les cas possibles. Un exemple de ce traitement est illustré dans la figure III.5.

La normalisation vise à unifier les diverses manières d'écrire un même mot, à corriger les fautes ou les incohérences évidentes. Etant donné que nous allons chercher la représentation des mots dans les modèles de Zahran et AraVec, le texte des réponses doit subir le même traitement qu'ont subit les données d'entrainement des modèles :

- Suppression des nombres des deux réponses
- Suppression des signes diacritiques des deux réponses :
- Suppression de toutes les lettres d'autres langues.
- Normalisation des caractères: (أ - إ - آ) à (ا) et (ة) à (ه) pour les modèle de Zahran.

- Normalisation des caractères: (أ - إ - آ) à (ا) et (ة) à (ه) et (ى) à (ي) pour les modèles d'AraVec.
- Supprimer les ajouts esthétiques : الأعمـــــــــال→ الأعمال

Afin de mieux illustrer l'effet de la normalisation, prenons une phrase du dataset STS 250 :

- Avant normalisation : «تؤدي الفرقة 8 أغنيات تحت الأضواء»
- Après normalisation : « تؤدي الفرقه اغنيات تحت الاضواء »

2.1.2. Le Stemming

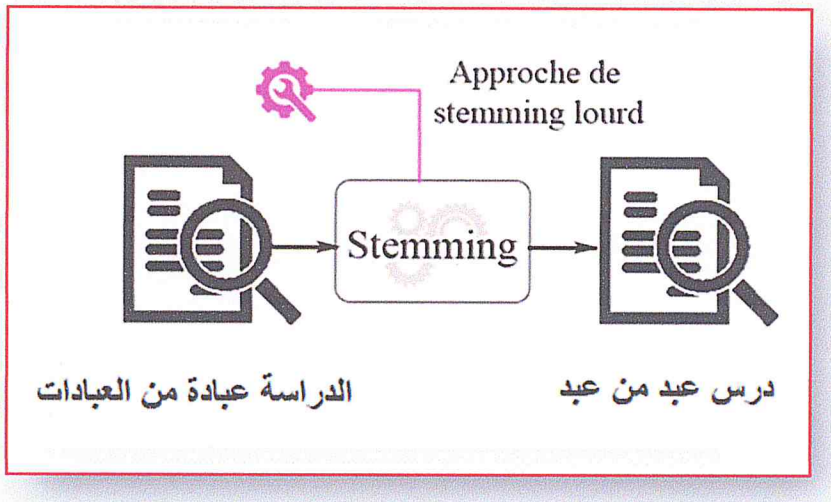

Figure III.6 Exemple de stemming

Comme ça a été introduit au début du chapitre, un algorithme de stemming peut être défini comme la procédure de réduction de tous les mots qui partagent la même racine à une forme commune. Le stemming consiste dans notre cas à réaliser les actions suivantes pour chaque couple de réponses à comparer. :

- Enlever le (ال : AL), et ses Dérivés, (فبال, تال, وال, كال بال, لل, ال, فال, لبال, وبال, ...)
- Supprimer le préfixe si la longueur du mot est supérieure à 3
- Supprimer le suffixe, si la longueur du mot est supérieure à 3. Une liste de préfixes, suffixes est disponibles et utilisée par le programme du stemmer. Cette liste est différente selon que le stemming est lourd ou léger.
- Supprimer les mots vides. Une liste de mots d'arrêt est disponible dans la base de données (هما هي, هو, اذا, ان, و, في, ...).

Les mots d'arrêt ou les stopwords sont des mots considérés sans importance pour le calcul de similarité et sont généralement supprimés des traitements (aussi bien dans les réponses que

dans les corpus). Ces dernier peuvent être calculés à partir d'un seuil de fréquence sur un corpus assez volumineux, néanmoins il existe plusieurs listes de stopwords plus ou moins similaires qui sont déjà calculées et rendues disponibles sur le net. Pour notre cas, nous avons maintenu la liste par défaut fournie par l'outil de stemming lourd que nous avons utilisé.

Pour appliquer l'approche de Stemming à notre travail, nous avons recherché parmi plusieurs stemmers existants dans la langue arabe. Nous avons testé les différents stemmers sur beaucoup de couples de réponses :

- Light10 Stemmer [67]
- Khoja Stemmer [68]
- ISRI Stemmer [69]
- Tashaphyne Stemmer [70]
- Motaz Stemmer [71]
- Assem Stemmer [72]

Pour l'exemple, voici les résultats de deux stemmers vers le root pour une réponse type concernant la définition de l'écologie tiré du dataset de Gomaa:

« الدراسة التى تتناول جوانب الطبيعة بما يحدده حياة الكائن و كيفية استخدامه لمكونات البيئة »

Word	Stem	Type	Evaluation
الإيكولوجيا	كولوج	ROOT	0
الدراسة	درس	ROOT	1
التى	لتى	ROOT	1
تتناول	نول	ROOT	0
جوانب	جنب	ROOT	1
الطبيعة	طبع	ROOT	1
بما	بما	ROOT	1
يحدده	حدد	ROOT	1
حياة	حية	ROOT	0
الكائن	كئن	ROOT	0
و	و	ROOT	1
كيفية	كيف	ROOT	1
استخدامه	استخدامه	ROOT	0
لمكونات	مكو	ROOT	0
البيئة.	ئة.	ROOT	0
Total			8/15

Tableau III-7 Exemple de stem avec ISRI Stemmer

Word	Stem	Type	Evaluation
الإيكولوجيا	ايكولوجيا	NOT STEMMED	0
الدراسة	درس	ROOT	1
التى	التى	NOT STEMMED	1
تتناول	نول	ROOT	0
جوانب	جنب	ROOT	1
الطبيعة	طبع	ROOT	1
بما	ما	STOPWORD	1
يحدده	حدد	ROOT	1
حياة	حيا	ROOT	0
الكائن	كون	ROOT	1
و	و	STOPWORD	1
كيفية	كيف	STOPWORD	1
استخدامه	خدم	ROOT	1
لمكونات	كون	ROOT	1
البيئة.	بيئة	NOT STEMMED	1
Total			12/15

Tableau III-8 Exemple de stem avec Khoja Stemmer

Il est à noté que comparé à l'anglais, les quelques stemmers qui existent ne présentent pas de documentation disponible et ne présentent pas une évaluation de la précision des résultats

obtenus. L'avis d'un expert en langue arabe nous a été difficile de procurer et par conséquent nous nous sommes basés sur l'appréciation de l'équipe pour évaluer les résultats obtenus et choisir d'utiliser les deux stemmers suivants dans la suite du travail :

- Khodja Stemmer [68] pour un stemming lourd
- Standford Stemmer [73] pour un stemming léger.

Concrètement, voici l'effet des deux stemmers sur un couple de réponses:

- Réponse modèle brute:
  « بعضها يلجأ إلى البيات الشتوى و البعض الأخر إلى الهجرة لمناطق أكثر ملائمة »
- Réponse candidate brute :
  « تلجأ إلى فترة السكون التى تسمى البيات الشتوى »

- Réponse modèle avec Standford Stemmer :
  « بعض يلجأ البيات الشتوى البعض الأخر الهجرة مناطق أكثر ملائمة »
- Réponse candidate avec Standford Stemmer:
  « تلجأ فترة السكون تسمى البيات الشتوى »

- Réponse modèle avec Khodja Stemmer:
  « بعض لجأ بوت شتوى بعض أخر هجر نطق كثر لوم »
- Réponse candidate avec Khodja Stemmer:
  « لجأ فتر سكن التى سمي بوت شتوى »

2.1.3. Calcul de fréquence et étiquetage morphosyntaxique

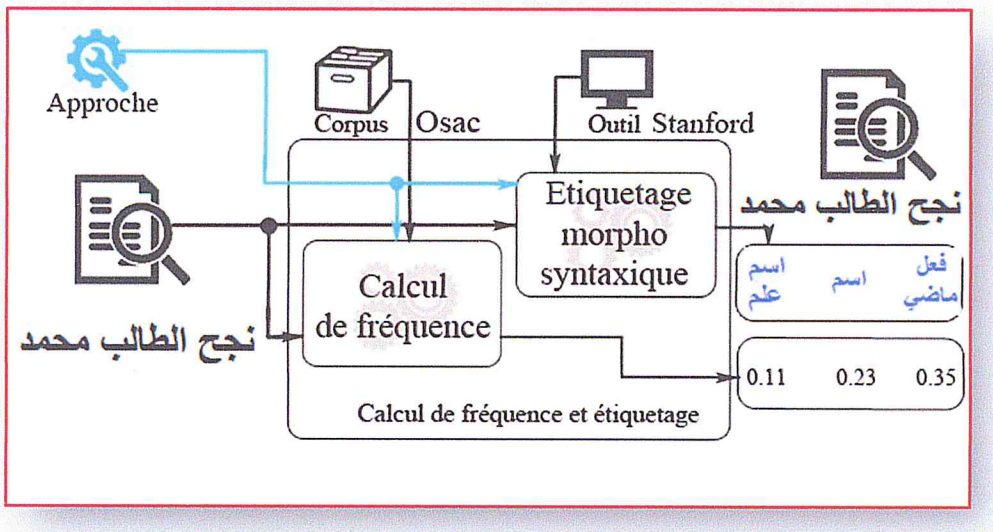

Figure III.7 Fonctionnement du module « Calcul de fréquence et étiquetage»

Plusieurs méthodes de calcul de similarité font recours aux méthodes statistiques pour améliorer les résultats de similarité en accordant de l'importance à la fréquence des mots dans les corpus ainsi qu'à l'étiquetage morpho syntaxique comme l'illustre la figure III.7.

2.1.3.1. Pondération des mots basée sur la fréquence de mots dans le corpus

Une des mesures les plus utilisés pour estimer l'importance des mots est la pondération IDF (Inverse Document Frequency) [74]. Elle opère une transformation des effectifs bruts des mots qui sont calculés dans le cadre du traitement de texte, et permet d'exprimer simultanément les fréquences auxquelles certains termes ou mots spécifiques apparaissent dans un ensemble de documents, ainsi que leurs spécificités sémantiques. Le principe de l'approche est de déterminer les mots les plus pertinents dans un texte et ceux qui sont insignifiants comme les stopwords. Ainsi les mots les moins fréquents sont les plus discriminants.

Cette mesure est généralement utilisée avec le calcul des TFs (Term Frequency) qui consiste à calculer la fréquence d'un mot par rapport à un document. Plusieurs variations de TF existent comme nous pouvons le voir dans la figure III.8 :

Schéma de pondération	formule du TF
binaire	$0, 1$
fréquence brute	$f_{t,d}$
normalisation logarithmique	$1 + \log(f_{t,d})$
normalisation « 0.5 » par le max	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
normalisation par le max	$K + (1 - K)\frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Figure III.8 Variantes de TF [62]

Le TF-IDF est une mesure largement utilisé, elle permet d'avoir une estimation globale sur l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus, elle est obtenue en multipliant TF par IDF :

$$TF\text{-}IDF\ (W) = IDF\ (W) * TF\ (W)$$

Dans le cadre de notre travail, les documents sont substitués par les réponses à comparer. Une réponse est de toute évidence beaucoup plus petite par rapport à un document, ce qui fait que les propriétés sémantiques des mots la composant sont nettement moins explicites et difficile à cerner. Pour cette raison, nous avons proposé dans notre approche d'autres façons de pondérer en considérant les mots par rapport à un corpus représentatif et par conséquent, à la langue de façon générale :

A. La pondération TFminmax [75]:

La pondération TFminmax utilise une autre mesure statistique calculée par le logarithme du nombre de fois où le mot apparait dans le corpus divisé par le nombre de mot total dans le corpus :

$$TFlog\ (W) = -\log\left(\frac{Cpt}{N}\right)$$

Où Cpt et le nombre de fois qu'apparait W dans le corpus, N = nombre de mots total dans le corpus. Une fois les TFlogs de tous les mots calculés, une normalisation est appliquer à ces derniers afin d'obtenir les TFminmax en divisons le tf-log du mot par le max des tf-logs obtenus :

$$TFminmax(W) = \frac{TFlog(W)}{Max(TFlogs)}$$

B. La pondération TF-IDFminmax :

Contrairement à TF-IDF de base que nous avons vue précédemment et où le but est d'avoir une estimation globale sur l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus, notre TF-IDFminmax génère une estimation globale de l'importance du mot par rapport à la langue entière. Cette estimation est obtenue par la formule :

$$TF\text{-}IDFminmax(W) = IDF(W) * TFminmax\ (W)$$

2.1.3.1. Pondération des mots basée sur l'étiquetage morpho-syntaxique POS (Part Of Speech):

Contrairement à la pondération de mots basée sur la fréquence des termes qui tire ses principes des méthodes statistiques, la pondération basée sur l'étiquetage morphosyntaxique fait recours au domaine linguistique. L'avantage principal de cette approche est qu'elle est indépendante et qu'elle ne nécessite pas des corpus ou de ressource externe à l'analyseur morphosyntaxique. C'est d'ailleurs pour cette raison que nous nous sommes orientés vers cette approche. Pour identifier les mots les plus importants dans un texte, cette approche se base sur les caractéristiques grammaticales et syntaxiques de la langue. D'abord un analyseur morphosyntaxique opère une analyse des textes afin de pouvoir classer les mots dans leurs catégories (Nom, verbe, adjectifs…). En suite, selon les propriétés de la langue utilisée, chaque catégorie se voit attribuer une valeur représentant son importance dans la langue.

Dans notre cas, nous avons utilisé l'analyseur « Stanford coreNLP »[73] pour générer l'étiquetage des réponses, ensuite, nous avons réalisé une série de tests qui a abouti à la considération de 3 catégories : les noms "N", les verbes "V" et la 3eme catégories "A" pour le reste. Voici quelques exemples :

- CC : conjonction, pondéré par 0.1
- CD : cardinal ou numérique pondéré par 0.1
- IN : préposition pondéré par 0.1
- JJ : Adjectif, pondéré par 0.1
- NN : nom commun, pondéra par 0.4
- RB : adverbe, pondéré par 0.1
- VB : verbe, pondéré par 0.5

2.2. Calcul des similarités

Le calcul de similarité entre les réponses modèles et les réponses des apprenants est le noyau de notre travail. Le module « Calcul des similarités »(Figure III.9) est un regroupement de plusieurs sous module de calcul des similarités indépendants les un des autres que nous avons développés. Notons que, dans la continuité de notre approche méthodologiques, les datasets ont été traités selon une approche de stemming, de ce fait, les résultats renvoyés par les modules de calculs sont dépendants de l'approche appliquée.

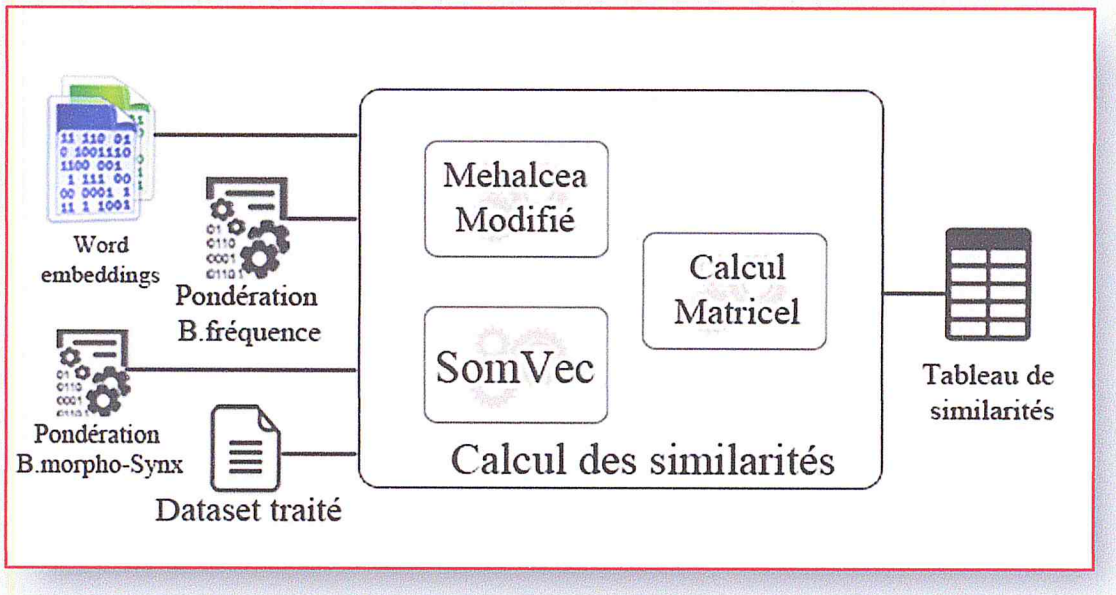

Figure III.9 Composants du module « Calcul des similarités »

Le calcul de similarité dans nos systèmes se fait au niveau des phrases pour certains et au niveau des mots pour d'autres :

2.2.1. Méthodes basées sur les similarités de phrases

Les méthodes basées sur la similarité au niveau des phrases appliquent un principe relativement simple : elles traitent la réponse modèle indépendamment de la réponse de l'apprenant dans un premier temps, puis, une fois les deux phrases représentées d'une façon adéquate, on fait appel à une mesure de similarité pour évalué la similitude entre les deux représentations. Le module SomVec implémente une méthode basée sur la similarité de phrase :

2.2.1.1. La somme des vecteurs : avec et sans pondération [76]

La somme des vecteurs est largement utilisée dans la littérature, elle consiste à sommer les vecteurs représentatifs de chaque mot de la phrase avant de calculer la similarité cosinus entre la somme des deux phrases. La similarité cosinus a été vue dans le chapitre 2.

Soit :

- Une phrase S1 composée des mots M_1, M_2 … M_N
- Une phrase S2 composée des mots K_1, K_2 … K_N

La première étape est de calculer V1 et V2 :

- $V1 = \sum_{i=1}^{N} v(M_i) * \beta$
- $V2 = \sum_{i=1}^{N} v(K_i) * \beta$

Ensuite il suffit de calculer la similarité cosinus entre les deux vecteurs :

- simCos(V1,V2) = cos (V1,V2)

Dans la somme des vecteurs simple, aussi appelé somme des vecteurs unitaire, le coefficient β est égal à 1. Nous considérons cette configuration comme Baselines de notre module d'évaluation comme nous allons le voir par la suite dans le chapitre 4.

Une amélioration de cette méthode consiste à modifier la pondération des vecteurs de mot en affectant à β une valeur représentative de l'importance de chaque terme. Sur ce principe, nous avons étendu le fonctionnement de notre module SOMVEC afin de pondérer suivant les approches de pondérations implémentées par le modules « calcul des fréquences et étiquetage » qui sont : IDF, TFminmax, TFlog, TF-IDFminmax, POS, MIXTE (combinaison entre TFminmax et POS).

Pour une meilleure compréhension, prenons l'exemple suivant:

P= « بعضها يلجأ إلى البيات الشتوى و البعض الأخر إلى الهجرة لمناطق أكثر ملائمة »

R= « تلجأ إلى فترة السكون التى تسمى البيات الشتوى »

- Extraction des vecteurs et calcul des sommes :

بعضها	[0.07 0.07 0.11 -0.28 ...]←	تلجأ	[-0.2 -0.09 0.07 -0.22 ...]←
يلجأ	[-0.18 0.16 0.1 -0.05 ...]←	إلى	[-0.12 0.13 -0.21 0.04 ...]←
إلى	[-0.12 0.13 -0.21 0.04 ...]←	فترة	[-0.37 0.07 -0.18 0.11 ...]←
البيات	[-0.34 -0.18 0.15 0.2 ...]←	السكون	[0.01 0.08 -0.18 0.18 ...]←
الشتوى	[-0.79 -0.06 0.12 0.44 ...]←	التى	[-0.11 0.14 0.37 -0.14 ...]←
و	[-0.16 0.1 -0.03 0.03 ...]←	تسمى	[-0.14 0.17 0.07 -0.11 ...]←
البعض	[0.02 0.05 0.11 0.07 ...]←	البيات	[-0.34 -0.18 0.15 0.2 ...]←
الأخر	[-0.05 0.25 0.09 0.09 ...]←	الشتوى	[-0.79 -0.06 0.12 0.44 ...]←
إلى	[-0.12 0.13 -0.21 0.04 ...]←		
الهجرة	[-0.05 0.37 0.14 -0.06 ...]←		
لمناطق	[-0.26 -0.22 0.07 -0.07 ...]←		
أكثر	[-0.14 0.02 -0.15 -0.31 ...]←		
ملائمة	[-0.24 0.07 -0.17 0.13 ...]←		
Somme R : [-2.36 0.67 0.12 0,31 ...]		Somme P : [-2.06 0.26 0.21 0.5 ...]	

Tableau III-9 Exemple explicatif de SomVec

- Calculer similarité : Similarité (P,R) = Cosinus(Somme1,Somme2)= 0.842

2.2.2. Méthodes basées sur les similarités de mots

Les approches de calcul de similarité basée sur les mots consiste à calculer la similarité de chaque mot d'une phrase avec tous les des mots de l'autre phrase, avant de passer vers la similarité des globale entre ces phrases suivant une formule qui défère d'une méthode à une autre. Nous allons détailler ici les méthodes que nous avons implémentées :

2.2.2.1. Calcul matriciel mot à mot (ou Matrice d'ordre) :

Le module « calcul matriciel » de la figure III.9 est implémenté par cette méthode. Notre approche inspirée des travaux de [28] utilise deux fonctions afin de calculer la similarité entre deux phrases : une fonction optionnelle de similarité d'ordre des mots en communs pour incorporer des informations syntaxiques dans le calcul, et une méthode de calcul de similarité sémantique mot à mot en utilisant les Word Embeddings.

Pour aller plus en profondeur, détaillons le fonctionnement des deux fonctions :

1) Similarité d'ordre des mots en commun dans les deux réponses

Cette similarité donne une importance aux mots en commun suivant leur ordres dans les phrases à comparer, Pour deux phrases P et R de taille m et n (m étant inférieure ou égale n sinon nous inversons les deux phrases), Nous retirons les K mots en commun de chaque phrase P et R, et nous les mettons dans les vecteurs X et Y respectivement en suivant le même ordre dans lequel ils apparaissaient dans P et R.

La 2eme consiste à remplacer chaque mot dans X par sa position dans X, en démarrant de 1 jusqu'à arrivée à α = |X|, ensuite remplacer dans Y chaque terme Yi où : Xi=Yi par le numéro remplacé avec ce dernier dans X. Ceci fait, le prochain travail à faire est de calculerS_0.

$$S0 = \begin{cases} 1 - \dfrac{2\sum_{i=1}^{\alpha} |X_i - Y|}{\alpha^2} & Si\ \alpha \text{ est pair} \\ 1 - \dfrac{2\sum_{i=1}^{\alpha} |X_i - Y|}{\alpha^2 - 1} & Si\ \alpha \text{ est impair et } \alpha > 1 \\ 1 & Si\ \alpha = 1 \end{cases}$$

2) Similarité sémantique mot par mot :

C'est une procédure itérative qui intervient après avoir retrié les mots en communs des phrases P et R pour construire la matrice comme suite :

D'abord, Il faut construire une matrice de taille (m-α)*(n-α) de sorte que la ligne contient la phrase la plus courte, sinon nous inversons entre la ligne et la colonne puis on calcule la similarité entre chaque mot Pi et Rj en appliquant le cosinus sur leurs vecteurs : Cosinus (V(Pi), V(Rj)).

Une amélioration possible est d'appliquer une pondération sur la matrice d'ordre, il suffit de multiplier les vecteur V(Pi) et V(Ri) par la pondération voulu avant de calculer le cosinus. Nous avons étendu ce module calcul matriciel pour qu'il prenne en charge la pondération MIXTE (TFminmax et POS).

Une fois la matrice construite, il faut extraire le maximum de la matrice pour l'ajouter à une liste de max (qu'on notera Max i pour la suite) et retirer la ligne et la colonne de ce maximum.

Ces étapes doivent être répétées jusqu'à la satisfaction d'une des deux conditions :

- la somme des éléments de la matrice devient nulle
- où bien : m $-\alpha-|P|<=0$

Une fois une des deux conditions atteinte, la similarité entre P et R peut être calculé comme suit :

$$S(P,R)=\frac{(\alpha(1-W_f+W_fS_0)+\sum_{i=1}^{|Maxi|}Maxi_i)+(m+n)}{2mn}$$

Wf est la pondération accordé à l'ordre pour les mots en communs, Nous pouvons ignorer l'ordre en affectant un 0 à Wf ce qui revient à calculer la formule suivante :

$$S(Phrase1, Phrase\ 2)=\frac{(\alpha+\sum_{i=1}^{|Maxi|}Maxi_i)+(m+n)}{2mn}$$

- Exemple explicatif :

P= « بعضها يلجأ إلى البيات الشتوى و البعض الأخر إلى الهجرة لمناطق أكثر ملائمة »
R= « تلجأ إلى فترة السكون التى تسمى البيات الشتوى »

- Etape 1 : parcourir les deux phrases et retirer les mots en commun, puis les mettre dans des vecteurs V1, V2 suivant l'ordre de base.

P= « بعضها يلجأ و البعض الأخر إلى الهجرة لمناطق أكثر ملائمة » → V1 = « إلى, البيات, الشتوى »
R= « تلجأ فترة السكون التى تسمى» → V2 = « إلى, البيات, الشتوى »

- Etape 2 : remplacer chaque mot dans V1 par sa position dans et remplacer chaque mot dans V2 par le chiffre équivalent dans le vecteur V1 où Xi=Yi

Nous aurons alors : V1 = V2 = « 1,2,3 »

- Etape 3 : Calculer S0

Pour α=3 : S0= 1-1 $-\frac{2\sum_{i=1}^{\alpha}|x_i-y_i|}{\alpha^2-1}$ → : S0= $1-\frac{2|1-1|+|2-2|+|3-3|}{3^2-1}=1$

- Etape 4 : Construction de la matrice (sans les mots en communs)

Ligne = « بعضها يلجأ و البعض الأخر إلى الهجرة لمناطق أكثر ملائمة »
Colonne = « تلجأ فترة السكون التى تسمى»

Nous constatons : |colonne|< |ligne|, de ce fait, on inverse les deux phrases :

Ligne= « تلجأ فترة السكون التى تسمى»
Colonne = « بعضها يلجأ و البعض الأخر إلى الهجرة لمناطق أكثر ملائمة »

Pour la création de la matrice, nous récupérons les vecteurs des mots depuis les modèles WE utilisés et on applique le cosinus :

mot1 = تلجأ vecteur1 : [-0.2 -0.09 0.07 -0.22 …]
mot2 = ملائمة vecteur2 : [-0.24 0.07 -0.17 0.13 …]
matrice [0][0] = cosinus(vecteur1,vecteur2)

En appliquant le même processus sur le reste des mots de ligne et colonne nous obtenons la matrice suivante :

	ملائمة	أكثر	لمناطق	الهجرة	إلى	الأخر	البعض	و	يلجأ	بعضها
تلجأ	0.37	0.72	0.20	0.30	0.45	0.27	0.33	0.22	0.32	0.42
فترة	0.39	0.33	0.39	0.42	0.43	0.33	0.43	0.28	0.46	0.40
السكون	0.23	0.17	0.27	0.23	0.17	0.26	0.23	0.23	0.19	0.22
التى	0.39	0.32	0.40	0.40	0.40	0.27	0.38	0.25	0.33	0.34
تسمى	0.43	0.34	0.24	0.38	0.39	0.26	0.34	0.28	0.31	0.37

Nous appliquons à présent le partie itérative : maximum = 0.46

	ملائمة	أكثر	لمناطق	الهجرة	إلى	الأخر	البعض	و	بعضها
تلجأ	0.37	0.72	0.20	0.30	0.45	0.27	0.33	0.22	0.42
السكون	0.23	0.17	0.27	0.23	0.17	0.26	0.23	0.23	0.22
التى	0.39	0.32	0.40	0.40	0.40	0.27	0.38	0.25	0.34
تسمى	0.43	0.34	0.24	0.38	0.39	0.26	0.34	0.28	0.37

Maxi = [0.46] → $\sum_i^{p-\alpha}\sum_j^{p-\alpha} mat[i][j] > 0$ *et* 8 -3 – 1 = 4

	ملائمة	أكثر	لمناطق	الهجرة	الأخر	البعض	و	بعضها
السكون	0.23	0.17	0.27	0.23	0.26	0.23	0.23	0.22
التى	0.39	0.32	0.40	0.40	0.27	0.38	0.25	0.34

تسمى	0.43	0.34	0.24	0.38	0.26	0.34	0.28	0.37

Maxi=[0.46 0.45] → $\sum_i^{p-\alpha}\sum_j^{p-\alpha} mat[i][j] > 0$ *et* 8 -3 – 2 = 3

	أكثر	لمناطق	الهجرة	الأخر	البعض	و	بعضها
السكون	0.17	0.27	0.23	0.26	0.23	0.23	0.22
التي	0.32	0.40	0.40	0.27	0.38	0.25	0.34

Maxi=[0.46 0.45 0.43] → $\sum_i^{p-\alpha}\sum_j^{p-\alpha} mat[i][j] > 0$ *et* 8 -3 – 3 = 2

	أكثر	الهجرة	الأخر	البعض	و	بعضها
السكون	0.17	0.23	0.26	0.23	0.23	0.22

Maxi = [0.46 0.45 0.43 0.40] → $\sum_i^{p-\alpha}\sum_j^{p-\alpha} mat[i][j] > 0$ *et* 8 -3 – 4 = 1

Matrice = [ ] → $\sum_i^{p-\alpha}\sum_j^{p-\alpha} mat[i][j] > 0$ *et* 8 -3 – 5 = 0

Nous nous arrêtons là et nous calculons S(P,R) avec Wf= 0

$$S(P,R) = \frac{(5+0.91)+10+5}{2*10*5} = 0.20$$

2.2.2.2. L'approche de Mihalcea modifiée[77]:

Cette approche a été implémenté par le module « Mihalcea Modifié ». Chaque mot de la réponse modèle est comparé avec tout mot dans la réponse d'apprenant, en créant une matrice de taille N*M où N est le nombre de mots dans la réponse modèle et M étant le nombre de mots dans la réponse d'apprenant. Nous cherchons à former une matrice où chaque ligne représente un mot de la réponse modèle et chaque colonne représente un mot de la réponse d'apprenant. La valeur d'une case Mat[i j] représente la similarité entre le mot i et le mot j obtenue en appliquant le cosinus aux vecteur des deux mots

Apres avoir construit la matrice de similarité entre mots, nous appliquons la formule suivante pour obtenir une similarité entre la réponse modèle et la réponse d'apprenant :

$$Sim(P,R) = \frac{1}{2}\left(\frac{\sum_{w\in\{P\}}(maxSim(w,R)*idf(w))}{\sum_{w\in P} idf(w)} + \frac{\sum_{w\in\{R\}}(maxSim(w,P)*idf(w))}{\sum_{w\in R} idf(w)}\right)$$

Où maxSim(w,R) représente la similarité max entre le mot w de P et tout les mots de R. La pondération idf est celle déjà présentée dans les sections précédentes.

- Exemple explicatif pour le couple de réponse :

P= « بعضها يلجأ إلى البيات الشتوى و البعض الأخر إلى الهجرة لمناطق أكثر ملائمة »
R= « تلجأ إلى فترة السكون التى تسمى البيات الشتوى »

- Création de la matrice :

Matrice [0 0] = Cosinus (Vecteur (بعضها) , Vecteur (الشتوى))

Vecteur(بعضها) = [0.07 0.07 0.11 -0.28 …] = V1
Vecteur (الشتوى) = [-0.79 -0.06 0.12 0.44 …] = V2
Cosinus (V1,V2) = 0.37

Mots	الشتوى	البيات	تسمى	التى	السكون	فترة	إلى	تلجأ
بعضها	0.37	0.39	0.39	0.23	0.39	0.43	0.29	0.46
يلجأ	0.72	0.32	0.33	0.17	0.32	0.34	0.21	0.33
إلى	0.33	1	0.43	0.23	0.38	0.34	0.20	0.49
البيات	0.20	0.20	0.24	0.31	0.15	0.30	1	0.19
الشتوى	0.39	0.49	0.50	0.34	0.50	0.37	0.17	1
و	0.30	0.52	0.42	0.23	0.40	0.38	0.21	0.45
البعض	0.45	0.40	0.43	0.17	0.40	0.39	0.28	0.55
الأخر	0.27	0.38	0.33	0.26	0.27	0.26	0.23	0.43
إلى	0.33	1	0.43	0.23	0.25	0.28	0.25	0.33
الهجرة	0.22	0.37	0.28	0.23	0.25	0.28	0.25	0.33
لمناطق	030	0.39	0.42	0.10	0.24	0.31	0.23	0.38
أكثر	0.32	0.2	0.46	0.19	0.33	0.31	0.1	0.57
ملائمة	0.42	0.45	0.40	0.22	0.34	0.37	0.16	0.41

- Calcul :

Pour obtenir maxSim(w, R) on prend le mot w avec toute la phrase R, en cherchant le maximum comme suit :

w= بعضها →maxSim(w,R) = 0.37 → idf(بعضها) = 0.32

$$Sim(P, R)= \frac{1}{2} * \left(\frac{0.46*0.32+0.72*0.4+1*0.24 +\cdots+ *0.3+1.45*0.5}{(0.3+0.4+0.24+\cdots+0.3+0.5)} + \frac{0.72*0.4+1*0.5+0.43*0.3+\cdots+1*0.1+1*0.3}{(0.4+0.5+0.3+\cdots+0.1+0.3)}\right)$$

Et ainsi nous obtenons la similarité : 0.51

Nous avons modifié cette approche en remplaçant dans la formule de calcul de la similarité Sim(P,R), la pondération IDF par les nouvelles pondérations TFMinMAX et TF-IDF.

2.2.3. Hybridation des mesures

Pour pouvoir estimer la flexibilité des Word Embeddings et mesurer l'impact de leur hybridation avec d'autres méthodes, nous avons effectué une suite de tests combinatoires sur 2 niveaux :

2.2.3.1. Combinaison des approches développées dans ce travail

Nous avons combiné les similarités générées par plusieurs de nos modèles de calcul de similarité afin d'améliorer les résultats obtenus. Les critères de combinaison considérés sont les suivants :

- Pour la première combinaison locale « Combine-ALL » : Nous avons pour chaque approche, et pour chaque modèle de Zahran (CBOW ou SG) combiné toutes les méthodes ayant en résultat une corrélation de Pearson supérieure à 75% sur le dataset de GOMAA, soit une corrélation supérieure à 65% de réussite sur le dataset STS 250 AR.
- Pour la 2eme combinaison locale « Combine-Best » : Nous avons combiné les deux voir trois méthodes de chaque approche et chaque modèle ayant présenté les meilleurs résultats.

2.2.3.2. Combinaison avec d'autres mesures développées sans recours aux WE

Une opportunité s'est offerte à nous pendant la réalisation de ce travail, à savoir, que deux binômes de notre spécialité travaillent aussi dans le même projet que nous sur l'évaluation automatique des réponses courtes appliquées à la langue arabe en utilisant des mesures de similarité syntaxiques et sémantiques n'utilisant pas les Word Embedding.

- Benayad Asma et Atoub Yasmine, ayant pour thème : Mesures de similarité sémantique pour un système d'évaluation automatique des réponses courtes : Application à la langue arabe
- Garoudja Khadidja et Abdallah Amina, ayant pour thème : Mesures de similarité syntaxique pour un système d'évaluation automatique des réponses courtes : Application à la langue arabe

Nous avons en premier lieu uniformisé nos sorties par rapport à leurs sorties générés de leurs modules de calcul de notes respectifs. Nous avons ensuite combiné notre meilleur Combine-Best avec le leur pour estimer l'amélioration apportée par une telle hybridation.

Les résultats de cette hybridation sont présentés dans le chapitre 4.

2.3. Calcul des notes et évaluation :

Le module « Calcul des notes et évaluation » est le dernier module de notre système. Il contient 4 sous modules, 3 pour la génération de score et un 4eme pour l'évaluation des résultats ; une évaluation faite en lot sur l'ensemble du dataset.

2.3.1. Le module Kmeans :

Le modules Kmeans est l'un des trois modules que nous avons mis en place pour passer des similarités obtenues vers la note finale. Il est comme son nom l'indique implémenté par l'algorithme de classification non supervisé K-means[78][79].

Traitant les datasets présentés auparavant, où les notes varient de 1 à 5, nous avons choisi la valeur K=11 pour permettre un pas de 0.5 dans cet intervalle. De ce fait les 11 classes se font affectées une note en partant de 0 pour la classe contenant les similarités les plus base à 5. Ce qui revient à noter sur une échelle de 11 notes (0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5).

2.3.2. Le module X5 :

L'implémentation de ce module se résume en une opération mathématique simple: la multiplication. En effet, tous les modules de calcul de similarité que nous avons développé revoient une valeur similarité comprise dans l'intervalle 0 et 1, les deux valeurs incluses. Pour obtenir une note sur une échelle de 0 à 5, il suffit de multiplier cette similarité par 5.

2.3.3. Le module KNN :

Contrairement aux deux autres modules chargés de la même tâche, celui là implémente l'algorithme de classification supervisé des K plus proches voisins (K-NN ou K-PPV) [80] qui doit être entrainé sur un jeu de données pour pouvoir par la suite prédire la classe d'une nouvelle entrée. Nous avons utilisé le dataset « MSRvid 368 AR » comme donnée d'apprentissage (d'entrainement).

2.3.4. Le module d'évaluation :

L'évaluation d'un système implémenté ou d'une approche proposée est indispensable pour estimer le succès d'une recherche [81]. Il devient primordial d'accorder un rôle central aux métriques d'évaluation qui consiste à comparer un résultat produit avec des résultats corrects attendus. L'analyse de plusieurs situations d'évaluation dans notre cas, illustre l'importance d'un choix cohérent des métriques et de l'utilisation conjointe de plusieurs métriques. En essayant d'analyser les résultats de ce travail, nous avons été confrontés à la détermination de la métrique à utiliser pour évaluer les scores obtenus par rapport aux scores manuels fournis.

Notre décision de choix de métriques a été influencée par les datasets et les travaux connexes qui ont utilisé ces mêmes datasets. La corrélation de Pearson [82] est la métrique la plus fréquemment utilisée par les recherches dans ce domaine. C'est le cas aussi des différents datasets utilisés dans ce travail. Bien qu'elle ne soit pas citée et utilisée dans la majorité des travaux connexes, nous avons choisi d'inclure conjointement au coefficient de Pearson, l'erreur quadratique moyenne (Root Mean Squared Error (RMSE)[81]) pour quantifier la différence (ou le décalage) entre le résultat(score) obtenu par le système et celui obtenu par l'expert humain.

- **Coefficient de Pearson(r) :**

En statistiques, étudier la corrélation entre deux ou plusieurs variables statistiques numériques, c'est étudier l'intensité de la liaison ("proportionnalité") qui peut exister entre ces variables. La mesure de la corrélation linéaire entre les deux se fait alors par le calcul du coefficient de corrélation linéaire, noté r. Ce coefficient est égal au rapport de leur covariance et du produit non nul de leurs écarts types. Le coefficient de corrélation est compris entre -1 et 1

Corrélation	Négative	Positive
Faible	de -0,5 à 0,0	de 0,0 à 0,5
Forte	de −1,0 à −0,5	de 0,5 à 1,0

Tableau III-10 Signification des valeurs la corrélation de pearson

Plus le coefficient est proche des valeurs extrêmes -1 et 1, plus la corrélation linéaire entre les variables est forte ; on emploie simplement l'expression « fortement corrélées » pour qualifier les deux variables. Une corrélation égale à 0 signifie que les variables ne sont pas corrélées linéairement. Le coefficient de corrélation est multiplié par 100 pour exprimer un pourcentage de corrélation. Dans notre cas les variables statistiques à considérer sont celles définies dans deux vecteurs l'un contenant les valeurs de scores entre les couples de réponses du dataset (réponse de l'étudiant, réponse modèle de l'enseignant) calculés automatiquement, le deuxième vecteur contient les scores, pour les mêmes couples de réponses, calculées par l'expert humain. L'objectif dans notre travail revient à maximiser ce coefficient.

- **Erreur quadratique RMSE (Root Mean Squared Error (RMSE)):**

L'erreur quadratique moyenne permet de quantifier une mesure synthétique de l'erreur globale commise. Pour calculer l'erreur quadratique moyenne RMSE, les erreurs individuelles sont tout d'abord élevées au carré, puis additionnées les unes aux autres. On divise ensuite le résultat obtenu par le nombre total d'erreurs individuelles, puis on en prend la racine carrée. L'erreur quadratique est probablement le critère quantitatif le plus utilisé pour comparer des valeurs calculées (ici les scores ou notes automatiques) et valeurs observées (scores manuels attribués par l'expert humain. C'est cette fonction que nous tentons de minimiser dans le cadre de ce travail.

En conclusion, l'évaluation de nos approches correspond à trouver la meilleure minimisation de l'erreur quadratique avec une maximisation du coefficient de corrélation.

Conclusion

Nous venons de mettre en revue notre approche de développement du système d'évaluation automatique basé sur les Word Embedding. Nous avons décrit l'architecture fonctionnelle du système et détaillé les différents modules liées aux approches adoptées. Nous avons terminé le chapitre par décrire les métriques d'évaluation de notre système qui fait l'objet du chapitre prochain qui inclut les résultats ainsi que les interprétations relatives à l'analyse expérimentale menées pour les différents modèles de similarité proposés et implémentés.

IV. Synthèse expérimentale et Résultats

i. **Démarche expérimentale**
ii. **Outils développés**
iii. **Ressources matérielles et logicielles**
iv. **Synthèse expérimentale**
v. **Résultats**

L'objectif de ce chapitre est de présenter les expérimentations menées, les résultats obtenus ainsi que les interprétations correspondantes. Après avoir fait nos pré-études sur la disponibilité des outils de traitement de la langue arabe d'un point de vue outils de traitement de texte ou de calcul de similarité, nous avons constaté que la langue arabe souffre d'un manque flagrant de ressources qui sont nécessaires lors d'un travail de recherche comme le notre. C'est pour cela que nous avons doublé nos efforts afin de développer un système qui englobe quatre outils qui seront d'une grande utilité dans la continuité des recherches futures en terme de gain de temps et de facilité des prétraitements de textes qui ne peuvent être menés manuellement particulièrement avec de grands corpus.

Nous présentons dans la suite, une description des étapes menées dans le processus d'expérimentation et les outils développés, une description des ressources matérielles et logicielles utilisées ainsi qu'une description des résultats obtenus.

i. Démarche expérimentale

Après avoir défini les grandes lignes de notre approche méthodologique dans le chapitre 3, nous allons voir à présent plus en détails le processus d'application de cette dernière :

1. Prétraitement des textes

Une fois l'acquisition des ressources terminée, nous avons entamé le traitement de ces dernières :

1.1. Les datasets

Le dataset de Gomaa est sous format XML initialement. Par contre, ceux issus de la compétition SemEval sont sous forme TXT. Nous avons choisi de décomposer chaque dataset en trois fichiers TXT:

- Un fichier pour les réponses modèles : contenant une réponse par ligne.
- Un fichier pour les réponses des apprenants : de la même façon, une réponse par ligne.
- Un fichier de note : chaque ligne correspond à une note attribuée manuellement à un couple de réponses.

Pour les datasets de STS 250 AR et MSRvid 368 AR, il s'agit de paraphrases et non pas de couples de réponses ce qui fait qu'une phase peut être considérée comme réponse modèle ou réponse d'un apprenant. Cependant, l'ordre est très important pour le dataset de Gomaa, surtout que chaque réponse modèle a 10 réponses d'étudiant rattachées à elle.

Adaptation des datasets terminé, nous avons effectué un stemming de ces derniers, une fois avec Khoja et une fois avec Stanford, générant ainsi pour chaque dataset deux autres copies.

1.2. Les corpus :

Ayant fait recours aux méthodes statistiques, nous avons effectué plusieurs manipulations sur le corpus « Osac » pour garantir des valeurs statistiques fiables, notamment :

- Un nettoyage de contenu : en supprimant de chaque fichier les caractères latins, les caractères de ponctuation, les chiffres arabes et latins et les signes diacritiques.
- Un stemming des documents : en appliquant pour une copie du corpus un stemming léger (avec Stanford stemmer) et pour une autre copie un stemming lourd (avec Khodja stemmer), nous avons donc, en final, 3 copies du corpus : un corpus brut et deux corpus stemmés.
- Une normalisation des textes : en replaçant les caractères « أ - إ - آ » par « ا » et « ة » par « ه » dans les 3 copies du corpus.

Concernant le choix du corpus, il sera justifié dans la partie suivante.

2. Génération des fréquences des mots :

Les méthodes statistiques permettent d'identifier les mots les plus importants d'un texte, nous avons décris le principe de ces derniers dans le chapitre 3 ainsi que les pondérations que nous avons utilisées, à savoir IDF, TFlog, TFminmax, TF-IDFminmax.

Comme on à pu le voir en expliquant le fonctionnement de notre module « calcul statistique » au chapitre 3, la génération des fréquences est basée sur des formules mathématiques qui nécessitent des données numériques, en s'aidant du package « GenSim » de pyhton, nous avons:

- Reconstitué le corpus :
  - Pour chaque document, un vecteur contenant les mots de celui-ci a été créé
  - Chaque vecteur créé est ajouté à un autre vecteur corpus formant une matrice du corpus au complet.
- Créé un dictionnaire: chaque mot unique de la matrice se voit attribuer un identificateur numérique unique. Nous avons fait appel à la méthode « Dictionary ». A partir de là, nous travaillons avec les identificateurs des mots au lieu des mots eux mêmes.
- Calculé les IDFs : les IDFs ont été récupérés depuis l'instance Dictionary créé à l'étape précédent par la méthode « dfs » de cette dernière.
- Calculé le nombre d'apparition de chaque élément : une représentation « sac à mot » où « bag of words » du corpus a été effectué grâce à la méthode « doc2bow » afin de faciliter les calculs.

Une fois ces calculs terminés, nous avons pu implémenter les formules mathématiques, les appliquer pour notre corpus numérisé, et enfin, exporter les résultats avec les formats décris dans la description du l'outil calcul de fréquence.

Concernant le choix du corpus, nous avons commencé par évaluer les corpus en notre disposition (Osac, BBC, CNN, Watan), le processus d'évaluation étant comme suit:

- Chaque corpus à été nettoyé, stemmé (uniquement avec Khodja) puis normalisé.
- Par la suite nous avons généré les fréquences des mots de chaque corpus
- Puis nous avons testé ces fréquences avec la méthode SomVec (décrite plus loin) sur nos datasets.

Les résultats obtenus indiquant clairement Osac comme meilleur corpus, nous l'avons donc retenu pour la suite de notre expérimentation.

3. Calcul de similarité :

Etant le noyau de notre système, nous avons développé plusieurs méthodes de calcul de similarité que nous avons représenté par des modules dans la Figure III.9 : SOMVEC, Calcul matriciel, Mehalcea modifié.

- SOMVEC : Ce module regroupe toutes nos méthodes dérivées de la somme des vecteurs qui sont :
  - SomVec : Somme des vecteur unitaire où $\beta = 1$
  - SomVecIDF, SomVecTFMINMAX, SomVecTFLOG, SomVecTFIDFminmax, SomVecPOS, SomVecMIXTE où β varie selon les mesures générées par les méthodes dont le nom est porté par la méthode.
- Calcul matriciel: Ce module comporte deux méthodes, l'une dérivé de l'autre :
  - MatSim : Calcul matriciel mot à mot où $\beta = 1$
  - MatSimP : Calcul matriciel mot à mot où β varie selon les mesures générées par la combinaison des méthodes TFMINMAX et POS qu'on notera pondération Mixte.
- Mehalcea modifiée : Ce module regroupe toutes nos méthodes dérivées de l'approche de Mahalcea
  - Mahalcea où β varie selon IDF, qui est la formule de base.
  - MahalceaIDF, MahalceaTFMINMAX, MahalceaTFLOG, MahalceaTFIDFMINMAX, MahalceaPM où β varie selon les mesures générées par les méthodes dont le nom est porté par la méthode.

L'implémentation des modules a été faite en python, deux points importants :

- La récupération des vecteurs des mots :

Toutes nos méthodes utilisent les modèles Word Embeddings. Pour faciliter cette manipulation nous avons utilisé plusieurs méthodes de gensim :

  - KeyedVectors.load_word2vec_format: pour charger les modèles de Zahran en mémoire.

- gensim.models.Word2Vec.load : pour charger les modèles des WE d'araVec en mémoire.
- Une fois le modèle chargé, nous pouvons récupérer le vecteur d'un mot directement par l'instruction : vecteur = modele[mot].
- model.similarity(x,y): qui calcul la similarité cosinus entre deux mots passés en paramètres.

Notons que pour utiliser les modules de Zahran, nous avons eu besoin de la configuration minimale décrite dans le titre juste avant.

- La récupération des pondérations : Nécessite de passer en paramètre de la méthode le dictionnaire du vocabulaire et le dictionnaire de pondération, par exemple IDF.p générés après le calcule des fréquences. Le dictionnaire étant de la forme { ID : mot } a besoin d'être inversé, une fois passé à la forme { Mot : ID } nous pouvons chercher l'ID d'un mot dans le dictionnaire puis la pondération de ce mot grâce a son ID dans le dictionnaire de pondération, qui est quand à lui, de la forme { ID : Valeur }

Une fois tous nos modèles de calcul de similarité implémentés, nous avons réalisé une série de tests sur chaque modèle. Considérons par exemple le modèle SomVec : nous avons évalué nos 3 dataset par ce modules, une fois avec les copies non stemmés, une 2eme fois avec les copies stemmés avec Khodja et une 3eme fois avec les copies stemmés avec Stanford. Les résultats sont, après conversion en notes, rapportés dans la partie suivante.

4. Calcul des notes et évaluation :

Les modèles de calcul de similarité renvoient des valeurs entre 0 et 1 les deux valeurs inclues. Ces valeurs doivent être converti en note afin d'évaluer correctement les modèles. Nous avons fait recours à deux algorithmes de classification :

- cKmeans (k-means Clustering in One Dimension) : une implémentation du Kmeans classique pour des données d'une seul dimension.
- KNN : nous avons utilisé le KNN du package « Sklearn » qu'on a entrainé avec les dataset « MSRvid 368 AR » en calculant la similarité entre les couples de phrases du dataset puis nous avons attaché à chaque similarité calculée la note manuelle accordée au couple de phrase d'où elle a été calculée. Les données générées ont été introduites comme bases d'apprentissage pour ce module. Par la suite. Nous avons réalisé des tests sur les deux autres datasets en notre possession en variant le K de 2 à 30. Les

meilleures résultats (figure x) ont été obtenue par un K au alentour du 20, cette valeur a été choisi comme paramètre.

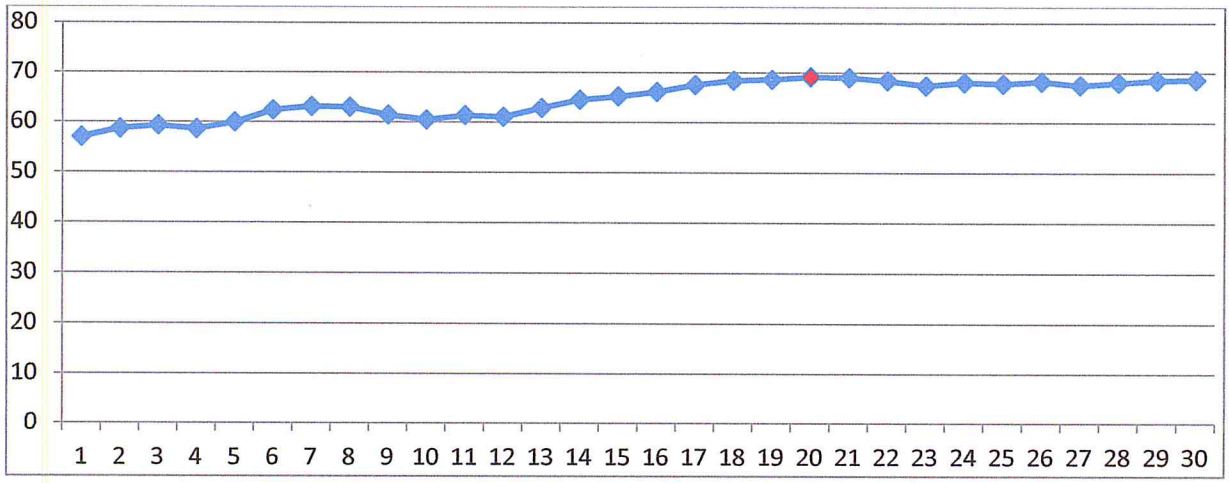

Figure IV.1 Variation du K pour les tests du KNN

Hormis les algorithmes de classification, nous avons implémenté une fonction « X5 » basique réalisant une multiplication de la similarité par 5 pour avoir une note.

Une fois nos modules de notation prêts, nous avons converti toutes les similarités en notes avec les deux modules Kmeans et X5, le KNN quand à lui, a été rapidement ignoré aux vues de ces résultats exposé dans la partie suivante.

ii. Outils développés

Le travail effectué expliqué, nous allons voir les outils qui ont été développés pour faire :

1. Un outil de nettoyage et de normalisation de texte

Cet outil réalise un nettoyage des documents textuels et une normalisation de ces derniers selon les paramètres sélectionnés. Même si les la normalisation que nous avons utilisé pour notre approche est suffisante pour d'autre cas d'utilisation, nous avons ajouté des fonctionnalisé à cet outils pour le rendre flexible et générique. Nous pouvons, entre autre, appliqué les opérations suivante :

- Supprimer les chiffres arabes et latins
- Supprimer les caractères latins
- Supprimer la ponctuation

- Supprimer les signes diacritiques
- Supprimer un ensemble de caractères introduit par fichier TXT
- Remplacer n'importe quel caractère par un autre
- Traiter des fichiers encodés avec UTF-8 ou UTF-16
- Traiter du texte depuis l'interface de l'application ou depuis un fichier.

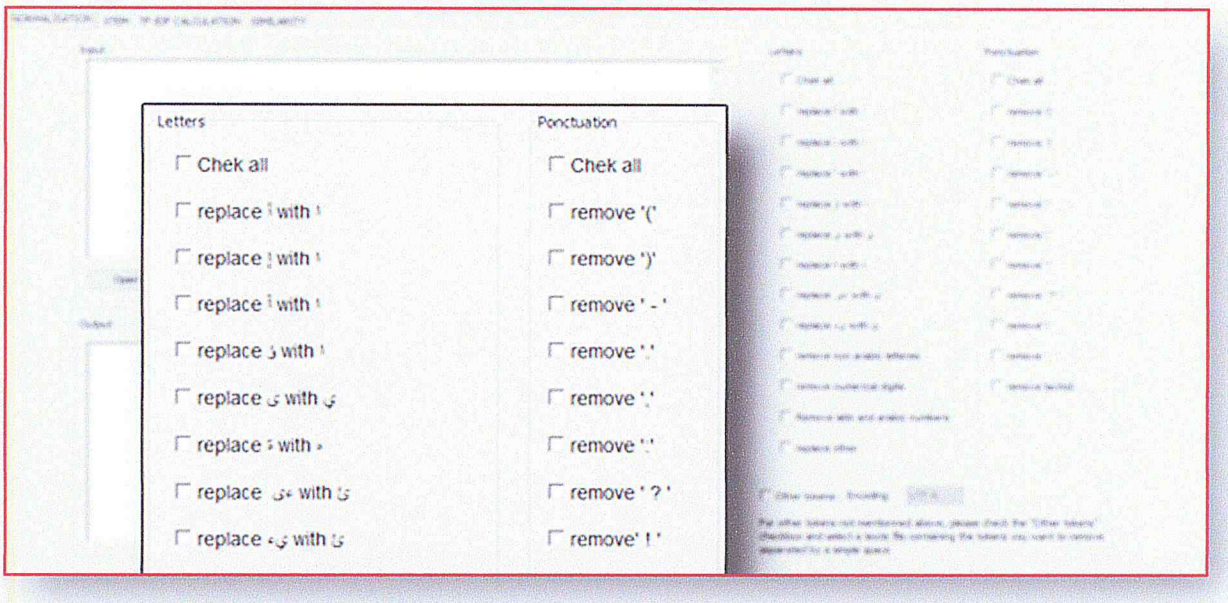

Figure IV.2 Outil de nettoyage et de normalisation de texte

2. Un outil de stemming

Cet outil regroupe les deux stemmers que nous avons utilisé, leurs formats d'affichage initial ne nous convenons pas, nous avons du apporter des modifications à ces derniers. Aussi, seul un fichier pouvait être traité à la fois et les problèmes d'encodage été assez fréquents. Cet outils essaye de répondre aux problèmes qu'on vient de citer en permettant notamment :

- Un stem léger ou lourd depuis la même application
- Traiter un ensemble de fichier à la fois, très pratique les corpus.
- Traiter des fichiers encodés avec UTF-8 ou UTF-16
- Traiter du texte depuis l'interface de l'application ou depuis un fichier.

Notons tous de même que cet outil est fonctionnel grâce aux deux stemmers que nous avons utilisés dans notre travail à savoir : Khoja et stanfordNLP, nous ne prétendons nullement avoir développé un stemmer au complet.

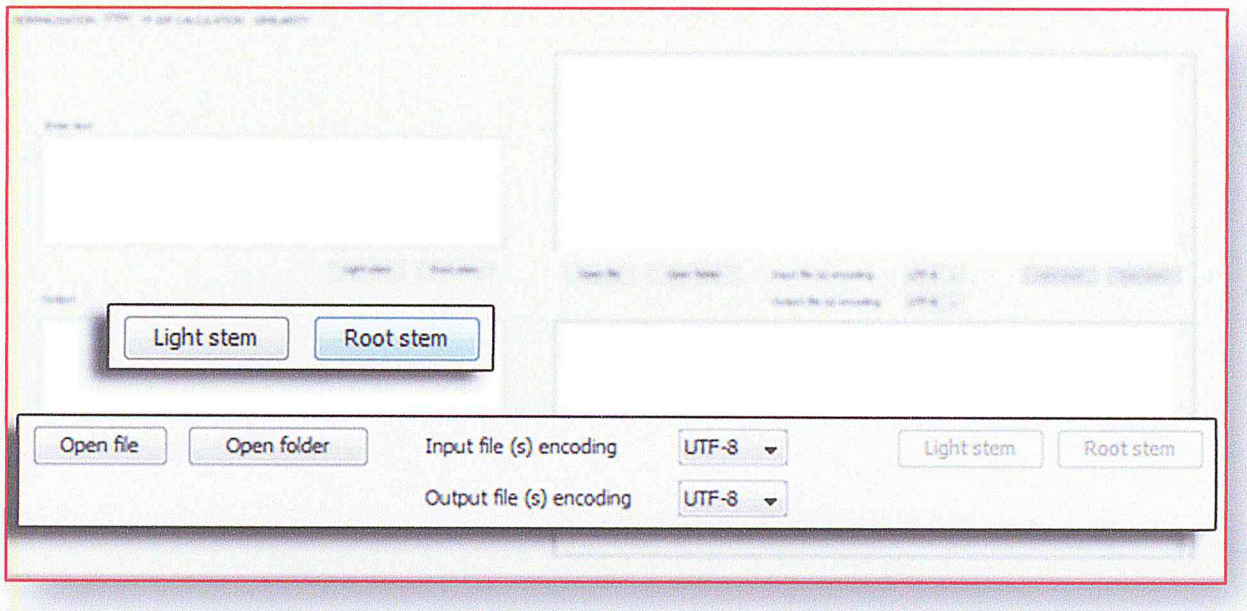

Figure IV.3 Outil de stemming

3. Outil de calcul de fréquence de mots

A l'exception de l'outil « KHAWAS »[83] , que nous considérons non fiable suite aux testes qu'on a réalisé, les outils dédié à cette tâche sont extrêmement rare dans la littérature. Notre programme permet de calculer la fréquence des mots selon les formules décrites précédemment dans ce chapitre.

Pour pouvoir effectuer ces calculs, il faut disposer d'un corpus significatifs, c'est-à-dire comportant un nombre important de mots (des milliards). Utiliser un corpus léger ou inapproprié par rapport au domaine engendre des statistiques peu fiables.

Le fonctionnement de ces outils est simple, il suffit de sélectionner le dossier à traiter et l'encodage des ces fichiers. Une fois les calculs terminés, les résultats sont retournés sous deux formats :

- Un fichier texte contenant :
  - Le nombre de fichiers traités
  - Le nombre de mot unique du corpus
  - Le nombre de mot total du corpus
  - Pour chaque mot : son ID, DF, IDF, TFlog, TFminmax, TF-IDFminmax

- Un ensemble de dictionnaire python :
  - dictionnaire.p de la forme {ID : Mot}
  - DF.p , IDF.p , TFlog.p , TFminmax.p , TF-IDFminmax.p de la forme { ID : Valeurs du mot selon la méthode}

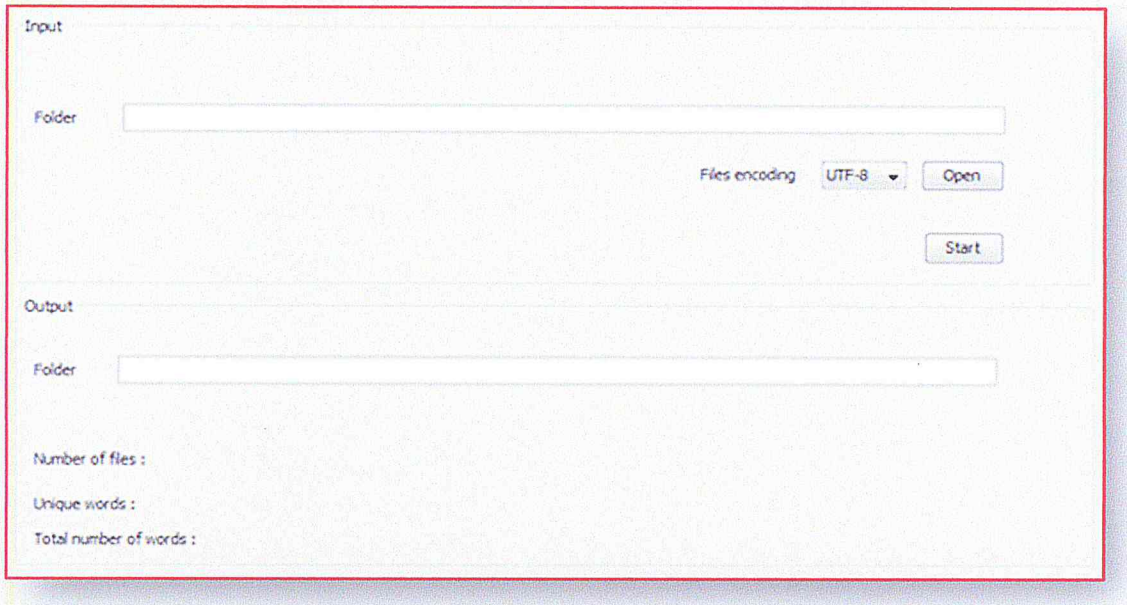

Figure IV.4 Outil de calcul des fréquences de mots

4. Outil d'évaluation automatique de réponses courtes

Cet outil est le noyau de notre travail, il représente la concrétisation de l'objectif escompté. Plusieurs fonctionnalités sont offertes par le biais de l'interface qui permet selon le cas de traiter soit :

- Le calcul de similarité et donc de score entre deux réponses entrées manuellement
- Le calcul de similarité et de score en lot pour deux fichiers textes en entrée contenant les réponses modèles et les réponses d'apprenants d'un examen, en spécifiant le nombre d'apprenant à évaluer.

Une fois les entrées saisies, l'outil apporte la possibilité du choix sur les options suivantes :

- *Modèle de Word Embedding :* L'application fournit l'utilisateur de quatre modèles au choix entre Zahran (CBOW, SkipGram) et Aravec (CBOW, SkipGram) ainsi que la possibilité de limiter ce modèle en cas d'utilisation d'une machine de faible puissance.

- *Stem :* par défaut établi à « sans stem », l'utilisateur peut sélectionner Heavy Stem ou Light Stem.
- *Format de sortie :* l'utilisateur a le choix entre avoir les résultats par leurs similarités ou par un score obtenu par la méthode 11-Means ou par la méthode X5.
- *Approche de calcul de similarité :* celle-ci est le noyau de l'application, nous avons donné le choix à l'utilisateur entre les différentes approches développées et entre une approche de combinaison recommandée :
    - **Recommandé :** approche qui calcule la similarité en utilisant la combinaison entre la matrice d'ordre pondéré et la formule de Mehalecea.
    - Somme de vecteurs à pondération unitaire.
    - Somme de vecteurs pondéra par TFminmax.
    - Somme de vecteurs pondéré par étiquetage morphosyntaxique.
    - Somme de vecteurs à pondération mixte (TFminmax + POS).
    - Mehalecea pondérée par IDF(méthode originale).
    - Mehalecea modifiée pondérée par TFminmax.
    - Mehalecea modifiée pondérée par TF-IDF.
    - Matrice d'ordre.
    - Matrice d'ordre pondéré.

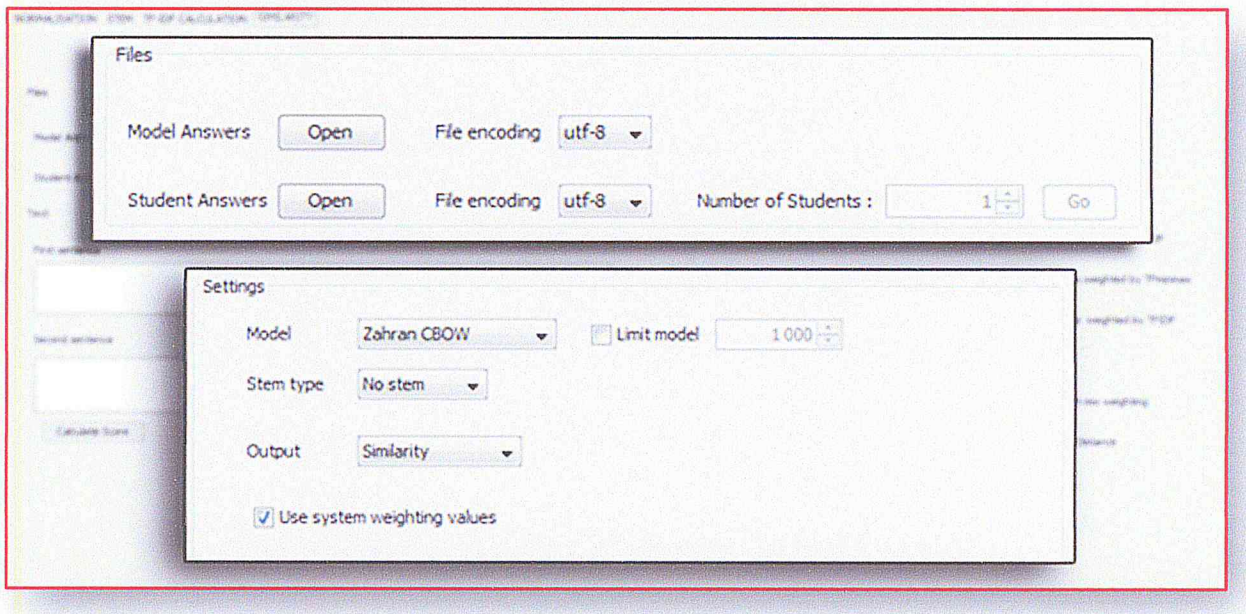

Figure IV.5 Outil d'évaluation automatique de réponses courtes

iii. Ressources matérielles et logicielles :

L'application a été développée en utilisant Python 2.6 d'une façon d'être adéquate à Python 3.6 en même temps, et en Java.

Nous avons rencontré une contrainte matérielle majeure puisque nos machines (4 GO RAM) ne pouvaient trainer des fichiers de WE de l'ordre de 15GO. Notre problème a été réglé une fois nous avons pu avoir l'accès à distance, par le biais d'un client SSH BitVise, à un serveur plus puissant dont les caractéristiques suivantes peuvent être considérées comme configuration minimale pour le bon fonctionnement de nos outils :

Serveur :	HP 470065-652 ProLiant ML350p Gen8 Intel Xeon
Processeur :	E5-2620 / 2 GHz
RAM:	16GO
OS :	Linux

iv. Synthèse expérimentale

1. Résultats de nos approches

L'analyse expérimentale quantitative a été menée pour chaque modèle de calcul de similarité dans chacune des 3 approches. Pour chaque modèle nous avons généré autant de « RUN » que de couples de réponses à évaluer. Les mêmes « RUN » sont exécutés sur les 3 DataSets.

Les notes obtenues automatiquement sont confrontées aux notes attribuées manuellement afin de calculer la corrélation de Pearson et l'erreur quadratique moyenne.

1.1.Méthode de base « Baselines »

Nous considérons les résultats de la méthode de la Somme des vecteurs sans pondération (SOMVEC) comme repère d'évaluation par rapport aux autres modèles proposés. Ce modèle sera appelé Système de base(ou méthode de base; nos propres baselines). Les résultats obtenus par l'application des 3 DataSets nous permettent d'évaluer l'amélioration enregistrée par une nouvelle approche.

Les tableaux qui suivent reportent les résultats de SOMVEC en utilisant les WE de Zahran (commentaires et interprétation : Tableau IV 8) :

Approche	Modèle	Datasets	Baseline(SomVec)	
			Kmeans	X5
Sans stem	CBOW	Gomaa	CP: 72,75 EQ: 1,15	CP: 73,04 EQ: 1,79
		STS 250 AR	CP: 55,87 EQ: 1,39	CP: 56,59 EQ: 1,88
		MSRvid 368 AR	CP: 75.70 EQ: 1.33	CP: 76.19 EQ: 1,63
	SKIPGRAM	Gomaa	CP: 73,60 EQ: 1,13	CP: 72,52 EQ: 2,17
		STS 250 AR	CP: 57,20 EQ: 1,34	CP: 55,36 EQ: 2,28
		MSRvid 368 AR	CP: 76.72 EQ: 1,4	CP: 75,38 EQ: 2,12

Tableau IV-1 Baseline Zahran sans stem

Approche	Modèle	Datasets	Baseline(SomVec)	
			Kmeans	X5
Heavy Stem	CBOW	Gomaa	CP: 75,34 EQ: 1,09	CP: 75,00 EQ: 1,54
		STS 250 AR	**CP: 63,88 EQ: 1,28**	CP: 64,49 EQ: 1,63
		MSRvid 368 AR	CP: 76,27 EQ: 1,33	CP: 76,35 EQ: 1,54
	SKIPGRAM	Gomaa	CP: 72,70 EQ: 1,16	CP: 72,13 EQ: 1,94
		STS 250 AR	**CP: 64,26 EQ: 1,31**	CP: 62,89 EQ: 2,11
		MSRvid 368 AR	CP: 76,00 EQ: 1,33	CP: 75,76 EQ: 2,06

Tableau IV-2 Baseline Zahran avec un stem lourd

Approche	Modèle	Datasets	Baseline(SomVec)	
			Kmeans	X5
Light Stem	Z-CBOW	Gomaa	**CP: 77,43 EQ: 1,04**	CP: 76,89 EQ: 1,66
		STS 250 AR	CP: 59,04 EQ: 1,79	CP: 59,72 EQ: 1,35
		MSRvid 368 AR	**CP: 76,98 EQ: 1,34**	CP: 77,39 EQ: 1,6
	SKIPGRAM	Gomaa	**CP: 77,84 EQ: 1,03**	CP: 75,98 EQ: 2,06
		STS 250 AR	CP: 57,90 EQ: 1,38	CP: 59,21 EQ: 2,18
		MSRvid 368 AR	**CP:78,00 EQ: 1,29**	CP: 76,96 EQ: 2,08

Tableau IV-3 Baseline Zahran avec un stem léger

Les tableaux qui suivent reportent les résultats de SOMVEC en utilisant les WE de AraVec (commentaires et interprétation : Tableau IV 8) :

Approche	Modèle	Datasets	Baseline(SomVec)	
			Kmeans	X5
Sans stem	CBOW	Gomaa	CP: 67.19 EQ: 1.24	CP: 66.14 EQ: 1.51
		STS 250 AR	CP: 53.3 EQ: 1.54	CP: 52.7 EQ: 1.52
		MSRvid 368 AR	CP: 41.84 EQ: 1.77	CP: 42.93 EQ: 1.74
	SKIPGRAM	Gomaa	CP: 74.38 EQ: 1.12	CP: 74.24 EQ: 1.83
		STS 250 AR	CP: 57.52 EQ: 1.4	CP: 54.5 EQ: 1.85
		MSRvid 368 AR	CP: 68.39 EQ: 1.37	CP: 68.54 EQ: 1.79

Tableau IV-4 Baseline araVec sans stem

Approche	Modèle	Datasets	Baseline(SomVec)	
			Kmeans	X5
Heavy Stem	CBOW	Gomaa	CP: 66.72 EQ: 1.24	CP: 67.12 EQ: 1.24
		STS 250 AR	CP: 52.68 EQ: 1.47	CP: 51.52 EQ: 1.55
		MSRvid 368 AR	**CP: 50.54 EQ: 1.78**	CP: 51.49 EQ: 1.69
	SKIPGRAM	Gomaa	CP: 75.79 EQ: 1.08	CP: 75.34 EQ: 1.62
		STS 250 AR	**CP: 60.44 EQ: 1.44**	CP: 58.47 EQ: 1.71
		MSRvid 368 AR	**CP: 71.72 EQ: 1.35**	CP: 69.86 EQ: 1.71

Tableau IV-5 Baseline araVec avec un stem lourd

Approche	Modèle	Datasets	Baseline(SomVec)	
			Kmeans	X5
Light Stem	Z-CBOW	Gomaa	**CP: 69.66 EQ: 1.21**	CP: 68.24 EQ: 1.46
		STS 250 AR	**CP: 54.72 EQ: 1.44**	CP: 54.27 EQ: 1.5
		MSRvid 368 AR	CP: 42.55 EQ: 1.74	CP: 42.93 EQ: 1.76
	SKIPGRAM	Gomaa	**CP: 77.24 EQ: 1.06**	CP: 76.53 EQ: 1.73
		STS 250 AR	CP: 57.91 EQ: 1.39	CP: 56.23 EQ: 1.77
		MSRvid 368 AR	CP: 68.12 EQ: 1.35	CP: 67.77 EQ: 1.76

Tableau IV-6 Baseline araVec avec un stem léger

Pour faciliter la lecture des résultats nous reportons dans un même tableau et pour chaque modèle de calcul de similarité, **les meilleurs résultats** obtenus par rapport aux WE (Zahran ou AraVec) et pour les deux modèles CBOW ou SKIPGRAM et dans les 3 approches : Sans Stem, Heavy Stem et Light Stem.

Nous utilisons la légende suivante dans la suite :

Abréviation	Description
LS	Light Stem
HS	Heavy Stem
SS	Sans Stem
Z-SG	Zahran SKIPGRAM
Z-CBOW	Zahran CBOW
AV-CBOW	AraVec CBOW
AV-SG	AraVec SKIPGRAM
CP	C. Pearson
EQ	Erreur Quadratique

Tableau IV-7 La légende des résultats

- **SOMVEC (Méthode de base) :**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	CP: 77,43 / EQ: 1,04/LS	**CP: 77,84 / EQ: 1,03/ LS**
	AraVec	CP: 69.66 / EQ: 1.21/ LS	CP: 77.24 / EQ: 1.06 / LS
STS 250 AR	Zahran	CP: 63,88 / EQ: 1,28 / HS	**CP: 64,26 / EQ: 1,31 /HS**
	AraVec	CP: 54.72 /EQ: 1.44/ LS	CP: 60.44 /EQ: 1.44/ HS
MSRvid 368 AR	Zahran	CP: 76,98 /EQ: 1,34/LS	**CP:78,00 / EQ: 1,29/ LS**
	AraVec	CP: 50.54 /EQ: 1.78/HS	CP: 71.72 EQ: 1.35/HS

Constat :

1. *AraVec présente des performances moindres dans les différents modèles et sur les différentes approches de Stems. Ceci est du au nombre de mots manquants dans AraVec ainsi que la taille du corpus et le nombre de mots << celui de Zahran. La qualité des WE a une influence sur le résultat de similarité.*
2. *Z-SG présente aussi les meilleurs résultats (tous) par rapport à CBOW.*
3. *La SOMVEC se prête bien avec un stemming plutôt light.*
4. *STS 250 se prête bien à un HS. Ceci est du à la nature du dataset qui peut contenir plusieurs mots de mêmes racines dans une phrase contrairement à de vraie réponse où plusieurs mots de même racine ne sont pas fréquents dans la même réponse.*

Tableau IV-8 Modèle de base : SOMVEC (baselines)

1.2. Les autres résultats comparés aux baselines

Nous présentons dans la suite (du Tableau 4.2 au tableau 4.14) les résultats obtenus en calculant les notes ensuite les la corrélation de Pearson(CP) et l'Erreur quadratique moyenne(EQ) pour chaque Dataset :

- **SomVecIDF :**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	CP: 71.71 / EQ: 1.15 / LS	CP: 76,29 / EQ: 1,06 / LS
	AraVec	CP: 72.52 / EQ: 1.15 / LS	**CP: 77.68 / EQ: 1.04 / LS**
STS 250 AR	Zahran	CP: 63,03 / EQ: 1,37 / HS	**CP: 64,74 / EQ: 1,59 / HS**
	AraVec	CP: 58.25 / EQ: 1.38 / HS	CP: 56.58 / EQ: 1.63 / HS
MSRvid 368 AR	Zahran	**CP: 78.25 / EQ: 1.31 / HS**	CP: 78.21 / EQ: 1.78 / SS
	AraVec	CP: 63.96 / EQ: 1.51 / HS	CP: 73.69 / EQ: 1.49 / SS
Constat : La pondération IDF ne donne pas de bons résultats <baselines pour l'erreur quadratique qui a augmenté. Nous pouvons expliquer ceci par la nature du corpus sur lequel les IDF ont été calculés. Ce corpus n'est pas lié directement au domaine.			

Tableau IV-9 Le modèle de similarité : SomVecIDF

- **SomVecTFLOG**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	CP: 75,87 / EQ: 1,59 / LS	CP: 76,23 / EQ: 1,07 / LS
	AraVec	CP: 71.71 / EQ: 1.15	**CP: 77.81 / EQ: 1.04 / LS**
STS 250 AR	Zahran	**CP: 64,73 / EQ: 1,59 / HS**	CP: 63,58 / EQ: 1,38 / HS
	AraVec	CP: 56.37 / EQ: 1.52 / HS	CP: 60.99 / EQ: 1.37 / HS
MSRvid 368 AR	Zahran	**CP: 78.15 / EQ: 1.46 / SS**	CP: 77.65 / EQ: 1.28 / LS
	AraVec	CP: 56.75 / EQ: 1.64 / HS	CP: 72.65 / EQ: 1.52 / HS
Constat : La pondération TFLOG ne donne pas de bons résultats <baselines pour l'erreur quadratique qui a augmenté. Nous pouvons expliquer ceci par la nature du corpus sur lequel les TFLog ont été calculés. Ce corpus n'est pas lié directement au domaine.			

Tableau IV-10 Le modèle de similarité : SomVecTFLOG

- **SomVecTFMINMAX :**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	CP: 78,43 / EQ: 1,61 / LS	**CP: 79,17 / EQ: 1,02 / LS**
	AraVec	CP: 71.61 / EQ: 1.17 / LS	CP: 78.41 / EQ: 1.04 LS
STS 250 AR	Zahran	CP: 60,32 / EQ: 1,92 / HS	**CP: 63,44 / EQ: 1,37 / HS**
	AraVec	CP: 56.41 / EQ: 1.52 / HS	CP: 61.75 / EQ: 1.41 / HS
MSRvid 368 AR	Zahran	**CP: 78.82 / EQ: 1.46 / LS**	CP: 78.25 / EQ: 1.25 / LS
	AraVec	CP: 57.11 / EQ: 1.58 / HS	CP: 72.59 / EQ: 1.65 / SS
Constat : La pondération TFMinMax améliore les baselines pour l'erreur quadratique et le CP. Elle est meilleure que TFlog et IDF			

Tableau IV-11 Le modèle de similarité : SomVecTFMINMAX

- **SomVecTFIDF**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	CP: 74,75 / EQ: 1,10 / LS	CP: 73,79 / EQ: 1,12 / LS
	AraVec	CP: 72.68 / EQ: 1.15 / LS	**CP: 76.18 / EQ: 1.07 / LS**
STS 250 AR	Zahran	**CP: 59,49 / EQ: 1,53 /HS**	CP: 58,20 / EQ: / HS
	AraVec	CP: 56.02 / EQ: 1.57 / LS	CP: 55.19 / EQ: 1.55 / HS
MSRvid 368 AR	Zahran	**CP: 77.43 / EQ: 1.21 / SS**	CP: 77.33 / EQ: 1.69 / SS
	AraVec	CP: 65.54 / EQ: 1.44 / LS	CP: 72.74 / EQ: 1.21 / SS
*Constat : Baisse au dessous des baselines prévisible puisqu'il s'agit du produit de deux mauvaises pondérations TF*IDF. Le résultat est prévisible.*			

Tableau IV-12 Le modèle de similarité : SomVecTFIDF

- **SomVecPOS**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	CP: 78.01 / EQ: 1.04 / LS	**CP: 78.16 / EQ: 1.04 / LS**
	AraVec	CP: 69.12 / EQ: 1.22 / HS	CP: 74.51 / EQ: 1.12 / SS
STS 250 AR	Zahran	**CP: 61.89 / EQ: 1.3 / LS**	CP: 61,25 / EQ: 1,29 / SS
	AraVec	CP: 55.75 / EQ: 1.47 / SS	CP: 55.8 / EQ: 1.47 / LS
MSRvid 368 AR	Zahran	**CP: 76.54 / EQ: 1.18 / SS**	CP: 76.53 / EQ: 1.3 / SS
	AraVec	CP: 40.93 / EQ: 1.81 / LS	CP: 66.8 / EQ: 1.75 / HS
Constat : La pondération par POS a amélioré l'erreur quadratique. Le morphologie du mot a son importance dans le sens.			

Tableau IV-13 Le modèle de similarité : SomVecPOS

- **SomVecPM**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	**CP: 79,09 / EQ:1,01 / SS**	CP: 78,86 / EQ: 1,03 / SS
	AraVec	CP: 68.97 / EQ: 1.21 / SS	CP: 75.57 / EQ: 1.08 / LS
STS 250 AR	Zahran	**CP: 61,84 / EQ: 1,30 / LS**	CP: 60,40 / EQ: 2,11 / LS
	AraVec	CP: 55.91 / EQ: 1.57 /SS	CP: 56.94 / EQ: 1.38 / HS
MSRvid 368 AR	Zahran	**CP: 77.98 / EQ: 1.49 / SS**	CP: 77.15 / EQ: 1.97 / ss
	AraVec	CP: 53.12 / EQ: 1.61 / HS	CP: 70.22 / EQ: 1.66/ SS

Constat :

1. *la pondération mixte (pos+TFMinMax) a amélioré les baselines.*
2. *Cette pondération se prête mieux avec CBOW et un SS ou LS*

Tableau IV-14 Le modèle de similarité : SomVecPM

- **MehalceaMoPM**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	**CP: 71,43 EQ: 1,31 / HS**	CP: 68,39 EQ: 1,75 / HS
	AraVec	CP: 65.86 / EQ: 1.35 / HS	CP: 65.53 / EQ: 1.35 / HS
STS 250 AR	Zahran	**CP: 57,15 EQ: 1,27 / HS**	CP: 53,56 EQ: 1,39 / HS
	AraVec	CP: 53.85 / EQ: 1.31 / HS	CP: 49.07 / EQ: 1.38 / LS
MSRvid 368 AR	Zahran	**CP: 68.56 EQ: 1.14 / HS**	CP: 65.25 EQ: 1.35 / HS
	AraVec	CP: 64.15 / EQ: 1.16 /SS	CP: 64.38 / EQ: 1.16 / HS

Constat :

1. *CBOW se comporte mieux dans les modèles de similarité mot à mot.*
2. *La formule ne se prette pas à un découpage morphosyntaxique de la langue.*

Tableau IV-15 Le modèle de similarité : MehalceaMoPM

- **MatSim**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	CP: 78,67 / EQ: 1,27/ LS	**CP: 79,51 / EQ: 1,15/LS**
	AraVec	CP: 77.76 / EQ: 1.32 / LS	CP: 77.9 / EQ: 1.15 /HV
STS 250 AR	Zahran	CP: 71,29 / EQ: 1,14 /HS	CP: 70,70 / EQ: 1,29 /HS
	AraVec	CP: 71.43 / EQ: 1.09 /HS	**CP: 71.46 /EQ: 1.09 / HS**
MSRvid 368 AR	Zahran	**CP: 75.39 / EQ: 1.06/ HS**	CP: 75.0 / EQ: 1.07 /HS
	AraVec	CP: 73.31 / EQ: 1.17 / HS	CP: 75.05 / EQ: 1.18 / HS

Constat global :

1. *Nette amélioration pour les GomaaDataset(avec LS encore) et STS250 avec HS encore.*
2. *Tous les meilleurs résultats des deux datasets SEMeval sont enregistrés dans une approche HS.*

Tableau IV-16 modèle de similarité : MatSim

- **MatSimP**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	**CP: 78,91 / EQ: 1,26 / LS**	CP: 78,84 / EQ: 1,20 /LS
	AraVec	CP: 77.72 / EQ: 1.3 / LS	CP: 77.55/ EQ: 1.38 / LS
STS 250 AR	Zahran	**CP: 72,40 EQ: 1,06/HS**	CP: 70,95 / EQ: 1,23 /HS
	AraVec	CP: 71.43 EQ: 1.09	**CP: 71.67 / EQ: 1.11 /HS**
MSRvid 368 AR	Zahran	CP: 75.2 / EQ: 1.01 /HS	**CP: 75.37 / EQ: 1.02 /HS**
	AraVec	CP: 73.08 / EQ: 1.14 / HS	CP: 74.49 / EQ: 1.16 / HS
Constat global : Considérée seule la pondération donne de meilleurs résultats par rapport aux baselines spécialement pour STS 250 avec un HS encore.			

Tableau IV-17 modèle de similarité : MatSimP

- **MehalceaMoTFminmax**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	**CP: 79,42 / EQ: 1,33 /LS**	CP: 79,31 / EQ: 1,34
	AraVec	CP: 74.41 / EQ: 1.15 / LS	CP: 74.0 / EQ: 1.17 / LS
STS 250 AR	Zahran	**CP: 70,44 / EQ: 1,24 /HS**	CP: 70,25 / EQ: 1,16 /HS
	AraVec	CP: 62.26 / EQ: 1.25 /HS	CP: 62.78 / EQ: 1.25 /HS
MSRvid 368 AR	Zahran	**CP: 78.48 / EQ: 1.01 / HS**	CP: 78.25 / EQ: 1.1 /HS
	AraVec	CP: 66.91 / EQ: 1.24 / HS	CP: 68.91 / EQ: 1.2 / HS

Constat global :

1. *Nette amélioration/baselines avec le modèle CBOW*
2. *Avec une pondération TFMINMAX c'est le modèle CBOW qui donnent les meilleurs résultats*
3. *LS donne les meilleurs résultats pour GommaDaset*
4. *HS donne les meilleurs résultats pour les deux datasets de SemEval*
5. *Pour Mehalcea, TFMinMAX est la meilleure pondération.*

Tableau IV-18 Le modèle de similarité : MehalceaMoTFminmax

- **MehalceaMoTFLOG**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	CP: 78,38 / EQ: 1,11 / LS	**CP: 79,10 / EQ: 1,34 / LS**
	AraVec	CP: 77.04 / EQ: 1.12 / LS	CP: 76.84 / EQ: 1.14 / LS
STS 250 AR	Zahran	**CP: 70,31 / EQ: 1,25 / HS**	CP: 70,14 / EQ: 1,47 / HS
	AraVec	CP: 61.08 / EQ: 1.25 / LS	CP: 63.63 / EQ: 1.25 / HS
MSRvid 368 AR	Zahran	**CP: 78.27 / EQ: 1.05 / HS**	CP: 77.34 / EQ: 1.16 /HS
	AraVec	CP: 77.34 / EQ: 1.16 / HS	CP: 69.28 / EQ: 1.21 / HS
Constat global : Bien que l'amélioration soit importante par rapport aux baselines, la pondération TFLOG ne donne pas de meilleurs résultats / la pondération TF-MinMax pour le modèle de Mihalcea.			

Tableau IV-19 Le modèle de similarité : MehalceaMoTFLOG

- **MehalceaMoTFIDF**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	CP: 78,22 / EQ: 1,11 / LS	**CP: 79,07 / EQ: 1,33 / LS**
	AraVec	CP: 75.44 / EQ: 1.14 / LS	CP: 75.12 / EQ: 1.16 / LS
STS 250 AR	Zahran	**CP: 68,58 / EQ: 1,28 /HS**	CP: 68,37 / EQ: 1,46 / HS
	AraVec	CP: 58.59 / EQ: 1.37 / HS	CP: 58.74 / EQ: 1.35 / HS
MSRvid 368 AR	Zahran	**CP: 78.74 / EQ: 1.12 / HS**	CP: 77.94 / EQ: 1.04 / SS
	AraVec	CP: 62.98 / EQ: 1.25 / LS	CP: 65.1 / EQ: 1.23 / SS
Constat global : Bien que l'amélioration soit importante par rapport aux baselines, la pondération TFIDF ne donne pas de meilleurs résultats / la pondération TF-MinMax pour le modèle de Mihalcea.			

Tableau IV-20 Le modèle de similarité : MehalceaMoTFIDF

- **MehalceaMoPM**

DataSet		CBOW	SKIPGRAM
Gomaa	Zahran	**CP: 71,43 EQ: 1,31 / HS**	CP: 68,39 EQ: 1,75 / HS
	AraVec	CP: 65.86 / EQ: 1.35 / HS	CP: 65.53 / EQ: 1.35 / HS
STS 250 AR	Zahran	**CP: 57,15 EQ: 1,27 / HS**	CP: 53,56 EQ: 1,39 / HS
	AraVec	CP: 53.85 / EQ: 1.31 / HS	CP: 49.07 / EQ: 1.38 / LS
MSRvid 368 AR	Zahran	**CP: 68.56 EQ: 1.14 / HS**	CP: 65.25 EQ: 1.35 / HS
	AraVec	CP: 64.15 / EQ: 1.16 /SS	CP: 64.38 / EQ: 1.16 / HS

Constat :

1. *CBOW se comporte mieux dans les modèles de similarité mot à mot.*
2. *La formule ne se prête pas à un découpage morphosyntaxique de la langue.*

Tableau IV-21 Le modèle de similarité : MehalceaMoPM

2. Hybridation de nos approches :

Nous avons combiné les similarités générées par plusieurs de nos modèles de calcul de similarité afin d'améliorer les résultats obtenus :

2.1. Combine ALL :

Nous n'avons considéré dans la combinaison que les WE de Zahran et les deux datasets celui de GOMAA et STS 250 AR, combinant toutes les méthodes ayant un CP supérieure à 75% pour GOMAA, ou supérieure à 65% pour STS 250 AR.

Combine All				
Approche	Modèles	Méthodes utilisées	Dataset	Résultats
Sans Stem	CBOW	SomVecPOS SomVecTFMINMAX SomVecPM MatSim MatSimP MehalceaMoIDF MehalceaMoTFMINMAX MehalceaMoTFLOG MehalceaMoTFIDF	Gomaa	CP : 78.23 EQ: 1.15
			STS 250	CP : 66.46 EQ: 1.22
	SkipGram		Gomaa	CP : 80.09 EQ: 1.02
			STS 250	CP : 66.77 EQ: 1.22

Tableau IV-22 Combine ALL pour l'approche sans stem

Combine All				
Approche	Modèles	Méthodes utilisées	Dataset	Résultats
Heavy Stem	CBOW	SomVec MehalceaMoIDF MehalceaMoTFMINMAX MehalceaMoTFLOG MatSim MatSimP	Gomaa	CP : 76.84 EQ: 1.19
			STS 250	CP : 70.27 EQ: 1.13
	SkipGram	MehalceaMoIDF MehalceaMoTFMINMAX MehalceaMoTFLOG MatSim MatSimP	Gomaa	CP : 76.39 EQ: 1.16
			STS 250	**CP : 70.28** **EQ: 1.16**

Tableau IV-23 Combine ALL pour l'approche Heavy Stem

Combine All				
Approche	Modèles		Dataset	Résultats
Light Stem	CBOW	SomVec SomVecPOS SomVec P.TFminmax MehalceaMoIDF MehalceaMoTFMINMAX MehalceaMoTFLOG MehalceaMoTFIDF MatSim	Gomaa	CP : 79.78 EQ: 1.06
			STS 250	CP : 65.69 EQ: 1.24
	SkipGram	SomVec SomVecPOS SomVecPM SomVec P.TFMINMAX MehalceaMoIDF MehalceaMoTFMINMAX MehalceaMoTFLOG MehalceaMoTFIDF MatSim MatSimP	Gomaa	**CP : 81.87** **EQ: 0.97**
			STS 250	CP : 68.04 EQ: 1.15

Tableau IV-24 Combine ALL pour l'approche Light Stem

Constat : pour la combinaison des approches WE développées, le meilleur résultat est atteint pour :

- Le dataset Gomaa : **CP : 81.87** **EQ: 0.0.97** atteint avec le modèle Z-SG dans l'approche LS
- Le dataset : STS 250 AR : **CP : 70.28** **EQ: 1.16** avec le modèle Z-SG dans l'approche LS

Le résultat atteint donne une nette amélioration des baselines et est prévisible qu'il soit atteint avec le SG et dans l'approche LS pour GOMAA et HS pour STS 250AR puisque les

approches testées individuellement ont donné les meilleurs résultats généralement avec ce modèle et ces approches.

Il est à noter ici la nette amélioration de l'erreur quadratique qui descend au dessous de 1.

2.2. Combine Best :

Il s'agit de combiner les approches dont la combinaison a donné de meilleurs résultats :

Combine Best				
Approche	Modèles		Dataset	Résultats
Sans Stem	CBOW	MatSim (0.4) SomVecPM (0.6)	Gomaa	CP : 81.05 EQ: 1.02
			STS 250	CP : 65.14 EQ: 1.24
	SkipGram	SomVecPM (0.6) MatSim (0.2) MehalceaMoTFMINMAX (0.2)	Gomaa	CP : 81.5 EQ: 0.98
			STS 250	CP : 66.29 EQ: 1.28

Tableau IV-25 Combine Best pour l'approche sans stem

Combine Best				
Approche	Modèles		Dataset	Résultats
Heavy Stem	CBOW	SomVecTFMINMAX *(0.4) MatSimP *(0.6)	Gomaa	CP : 78.56 EQ: 1.09
			STS 250	CP : 69.73 EQ: 1.12
	SkipGram	MatSim *(0.5) MatSimP *(0.5)	Gomaa	CP : 77.31 EQ: 1.18
			STS 250	**CP : 70.21 EQ: 1.12**

Tableau IV-26 Combine Best pour l'approche Heavy Stem

Combine Best				
Approche	Modèles		Dataset	Résultats
Light Stem	CBOW	Somme des vecteur MIXTE (0.59) MatSimP(0.41)	Gomaa	CP : 81.65 EQ: 0.97
			STS 250	CP : 65.20 EQ: 1.28
	SkipGram	**SomVecTFMINMAX (0.71) MatSimP (0.29)**	Gomaa	**CP : 83.08 EQ: 0.93**
			STS 250	CP : 65.32 EQ: 1.18

Tableau IV-27 Combine Best l'approche Light Stem

Constat :

- la combinaison best = **(SomVecTFMINMAX (0.71), MatSimP (0.29))** avec une approche LS et utilisant le modèle SG.
- Le résultat obtenu : CP **: 83.08 EQ: 0.93** donne une amélioration remarquable des résultats selon le CP et EQ qui sont améliorées les deux.

3. Hybridation avec d'autres mesures développées sans recours aux WE

Pour améliorer encore les résultats, nous avons combiné notre Combine-Best avec le combine-best des deux autres binômes travaillant sur la même thématique. En premier lieu uniformisé nos sorties par rapport à leurs sorties générés de leurs modules de calcul de notes respectifs. Nous avons ensuite combiné notre meilleur Combine-Best avec le leur pour estimer l'amélioration apportée par une telle hybridation.

Dans le tableau nous résumons les meilleurs résultats obtenus de cette combinaison pour les deux datasets :

Combinaison	**Résultats**
Notre Combine-best(0.81), mesure syntaxique(0.19)	Gomaa Dataset **CP : 84.2313 EQ : 0.93**
Notre Combine-best, mesure sémantique	STS Dataset **CP : 73.018 EQ :1,11**

Tableau IV-28 Résultats d'hybridation avec d'autres mesures développées sans recours aux WE

Constat:

- La combinaison avec l'approche syntaxique particulièrement a permis de surpasser les résultats obtenus par les travaux connexes de manière remarquable pour le dataset GOMAA.
- La combinaison avec l'approche sémantique développée a permis par contre d'améliorer nettement les résultats du dataset STS 250 AR

4. Discussion des résultats

4.1. Discussion par rapport aux travaux connexes/Comparaison des résultats

Pour Gomaa Dataset les données sont notées par deux juges humains, de sorte que l'erreur est rapportée par rapport à la moyenne des deux scores. L'IAA (Inter Annotator Agreement : Accord inter-annotateurs) est calculé entre les deux scores fournis par les juges. La corrélation signalée est la corrélation de Pearson. Pour le dataset GOMAA nous rapportons les résultats rapportés de son travail sur les données traduites en anglais et ceux de son travail sur les données arabes brutes. Les résultats obtenus par combinaison d'un modèle de similarité syntaxique (TableauIV.29) surpassent les travaux connexes et de plus se rapprochent de beaucoup de l'IAA particulièrement concernant la corrélation. C'est un objectif même que d'atteindre que de se rapprocher des performances humaines car il est clair qu'un outil arrive exceptionnellement à sur-dépasser l'évaluation humaine en général.

Pour le STS 250 AR MSRVid 368 arabic on enregistre une amélioration très importante avec la combinaison sémantique (Tableau IV-30 et Tableau IV-31)

	Erreur Quadratique Moyenne	Corrélation de Pearson
(Gomaa 2014) IAA [40]	0.69	0.86
(Gomaa 2014)English	0.75	0.83
(Gomaa 2014)Arabic [40]	1.07	0.73
Vectorized – Arabic[43]	**0.89**	**0.84**
Basic System - Arabic (notre baseslines: SOMVEC)	**1,03**	**0,78 (77,84%)**
Notre combine Best + syntaxique	**0.93**	**0.84 (84.23%)**

Tableau IV-29 Résultats accomplis pour Gomaa Dataset

	Erreur Quadratique Moyenne	Corrélation de Pearson
Semeval baselines (cosine)	-	60.45
([44]) semeval 2017	-	74.63
Basic System - Arabic (nos baseslines : SOMVEC)	**1,31**	**64,26**
Combine best WE (matrice d'ordre pondérée)	1.06	72.40
Notre combine Best avec STS sémantique	**1,1**	**73.018**

Tableau IV-30 Résultats accomplis pour STS 250 AR

	Erreur Quadratique Moyenne	Corrélation de Pearson
Nos baseslines	**1,29**	**78,00**
[Nagoudi] Semeval 2017	**-**	**79.69**
Best+syntaxique SomVecTFMINMAX(0,56) + Dice(0,44)	**1,08**	**79.76**

Tableau IV-31 Résultats accomplis pour MSRvid368 AR

4.2. Discussion globale

Rappelons que l'interprétation des résultats la plus réaliste c'est celle faite sur le dataset Gomaa qui a été crée dans le contexte d'une évaluation automatique des réponses courtes (l'objet même de notre thématique). Les datasets du SemEval ont été considérés à titre indicatif pour mesurer le degré de généralisation de l'approche pour d'autres domaines. Au terme de cette synthèse expérimentale, nous aboutissons à plusieurs constatations que nous allons aborder selon leurs aspects suivants :

a) Impact de l'approche de Stemming :

En considérant les résultats obtenus sans considérer le Stem(SS) nous pouvons apprécier l'impact de l'approche du stem sur les résultats. L'approche LS se présente catégoriquement meilleure pour GOMAA(et donc pour nos travaux). L'approche HS se prête mieux pour les datasets du SemEval. Une explication peut être donnée ici : STS 250 se prête bien à un HS. Ceci est du à la nature du dataset qui peut contenir plusieurs mots de mêmes racines dans une phrase contrairement à de vraies réponses où plusieurs mots de même racine ne sont pas fréquents dans la même réponse.

b) CBOW ou SKIPGRAM ?

En général SkipGram présente de meilleurs résultats particulièrement dans les approches qui calcule la similarité basée sur la somme des vecteurs. On peut expliquer ceci par la nature même du modèle qui à partir d'un mot génère le contexte et donc la sommation des vecteurs de mots capture mieux les contextes.

c) Les pondérations :

Une des constatations sûres tirée de cette synthèse est que la pondération, par une approche statistique ou linguistique, est un bon moyen pour améliorer les résultats. La qualité d'une

pondération statistique est influencée par la nature du corpus d'apprentissage ; plus les thématiques du corpus sont proches du sujet traité, plus la pondération est fiable. La pondération TFminmax que nous avons proposé capture efficacement l'importance des mots par rapport à la langue, elle se prête bien avec le modèle CBOW aussi. D'un autre coté, la pondération POS est situationnelle, elle ne peut être utilisée avec tous les modules de calcul de similarité.

d) Génération de notes (de la similarité au score) :

Nous constatons clairement que les approches non supervisés (Kmeans ici) sont nettement plus performantes que les approches supervisés (KNN pris en exemple). Ceci peut s'expliquer par la qualité des données d'entrainement et aussi, comme par exemple pour le KNN, par le fait qu'il n'est pas adapté aux problèmes dis mal posés comme celui de générer des notes.

e) L'approche hybride :

Nous sommes arrivés à la conclusion que l'hybridation permet d'atteindre des résultats nettement meilleurs. Dans notre cas nous avons combiné manuellement les poids affectés à chaque approche. Toutefois il serait intéressant dans l'avenir d'appliquer un algorithme de régression linéaire pour trouver la meilleure pondération possible des poids des approches à hybrider.

Une autre constatation est la flexibilité des modules de calcul des similarité implémentés avec les WE à être combinés avec d'autre modules issues d'autres approches tels que syntaxique ou sémantique avec une meilleure compatibilité avec les méthodes syntaxique. Ceci peut s'expliquer par l'importance qui doit être accordée à la syntaxe pour une langue aussi flexionnelle que la langue arabe.

f) Importance du contexte :

La différence de résultats entre les datasets est conséquente, ceci s'explique principalement par le contexte pour lequel ils sont été préparés, Même si un module d'évaluation automatique peut, jusqu'un un certain point, répondre à la problématique de détection de paraphrase, celui-ci excelle beaucoup plus dans son contexte où l'apprenant essaye de formuler une réponse similaire aux énoncés vus en cours. De ce fait, certaines mesures de similarité sont plus adaptées à la problématique de l'évaluation automatique des réponses courtes comme le mesure syntaxique DICE qui se base sur les mots en communs des deux propositions à comparer. L'approche de stemming lourd et le modèle CBOW semble donner de meilleurs

résultats dès qu'on s'éloigne du contexte. Cette constatation pourrait probablement être exploitée dans la problématique de détection de paraphrase et appliquée à la détection de plagiat par exemple.

g) Qualité des Word Embedding :

AraVec présente des performances moindres dans les différents modèles et sur les différentes approches de Stems. Ceci est du au nombre de mots manquants dans AraVec ainsi que la taille du corpus et le nombre de mots << celui de Zahran. La qualité des WE a une influence sur le résultat de similarité. Le tableau ci-dessous donne une idée du nombre de mots manquant lors des « RUN » de quelques approches par rapport aux WE Aravec et Zahran : (L'annexe A présente la liste de tous les mots manquants dans les différentes approches)

Datasets	Sans stem	
	Zahran	Aravec
GOMAA	Nombre de mots manquants: 5	Nombre de mots manquant: 100
STS 250 AR	Nombre de mots manquants: 9	Nombre de mots manquants: 133
MSRvid 368 AR	Nombre de mots manquants: 1	Nombre de mots manquants: 103

Tableau IV-32 Comparaison des mots manquants entre Zahran et araVec WE modèles

Les Word Embeddings sont très susceptibles aux corpus d'apprentissage. Plus ces derniers sont volumineux, plus les modèles générés sont de meilleure qualité et représentatifs de la langue. C'est d'ailleurs ce qui explique que les WE de Zahran sont nettement plus performants que ceux d'araVec.

En conclusion :

L'utilisation des word embeddings permet d'atteindre des résultats importants voir meilleurs dans la littérature. C'est une approche révolutionnaire pour la capture du sens sémantique et de l'importance des mots d'une langue donnée. Son indépendance de la langue et des ressources linguistique apportera sûrement beaucoup au domaine du TAL dans les prochaines années.

V. Conclusion et Perspectives

Les approches vectorielles ont prouvé leur validité au cours des dernières années en tant que techniques prometteuses pour la représentation des mots et des phrases. L'évaluation automatique des questions à réponses courtes est un problème difficile dans le traitement du langage naturel qui peut réduire beaucoup l'effort humain, en conséquence nous avons axé notre recherche sur l'exploitation des Word Embeddings pour résoudre ce problème dans le contexte de la langue arabe.

Notre travail se situe à la conjoncture de plusieurs domaines, ce qui nous a permis de présenter, dans ce mémoire, un état de l'art sur les systèmes d'évaluation automatique des réponses courtes, sur les approches de similarité des textes courts ainsi qu'une analyse de la représentation vectorielle en utilisant les Word Embedding. L'analyse faite nous a permis de mettre l'accent sur les enjeux de l'utilisation de la langue arabe dans le domaine de l'évaluation automatique et des approches de similarité.

Nous avons ensuite proposé et comparé une large gamme de mesures de similarité dont certaines sont existantes dans la littérature et d'autres proposées ou modifiées et finalement un système d'évaluation à réponses courtes est développé autours de plusieurs modules de calcul de similarité utilisant les Word Embeddings (Zahran et AraVec ; les seuls disponibles pour la langue arabe). Le système a été combiné à d'autres approches syntaxiques et sémantiques et il retourne une nette amélioration des résultats déjà obtenus avec les WE.

Evalué sur 3 DataSets, le système surpasse les systèmes des travaux connexes sur le Gomaa Dataset. A partir des résultats obtenus sur les SemEval Datasets, nous pouvons déduire que le système proposé peut très bien se généraliser par rapport aux autres langues qui sont dans le même défi du manque de ressources que la langue arabe car le système a obtenu de très comparables résultats dans les deux ensembles de tests génériques. Il convient de noter dans ce même ordre d'idées, que les stemmers sont les seuls outils dépendants de la langue utilisés dans notre approche ce qui la rend facilement généralisable. Les outils NLP pour la langue arabe développés autours du système tentent aussi de combler l'enjeu du manque de ressources.

En perspectives, nous prévoyons d'étendre le système proposé sur plusieurs aspects :

✓ Couvrir le système davantage de jeux de données arabes. Ce manque flagrant de jeux de données en langue arabe nous incite à penser sérieusement à la création de notre propre dataset. Ce qui nous permet de mener une évaluation plus objective et surtout enrichir l'ère d'évaluation qui même si elle est indépendante des méthodes et modèles proposés, elle devient encore plus cruciale. Il ne suffit plus au fait de s'arrêter à développer un système d'évaluation automatique pour la langue arabe, il va falloir œuvrer à le rendre concurrentiel.

✓ Le système développé doit être évaluée de manière réaliste sur une plateforme de e-Learning pour en mesurer l'impact et la réalisation dans un contexte réel. De ce fait, le développement d'un PLUGIN pour faciliter l'intégration de notre système à n'importe quelle plateforme d'apprentissage en ligne est en perspective particulièrement dans le contexte de la plateforme MOODLE.

✓ Etudier l'impact du domaine et la nature du corpus sur les pondérations que nous avons proposées en variant les corpus et introduire les corpus nommés pour considérer les entités nommées que nous avons ignorées dans le contexte de ce travail.

✓ Tester notre système avec ses différentes approches sur un dataset d'une autre langue. Nous pensons en perspective à le faire avec le dataSet crée par l'approche de Mihalcea pour l'anglais vu la disponibilité du dataset et son utilisation large. Ceci va nous permettre d'un côté de vérifier la généralisation de notre approche sur d'autres langues et d'un autre coté comparer les performances de notre système avec celui de Mehalcea pour les nouvelles pondérations proposées pour cette même approche.

✓ Enrichir l'évaluation à la prise en compte de la spécification des remarques de l'enseignant et du retour de feedback à l'apprenant permettant de lui spécifier les parties de sa réponse qui nécessitent plus d'investissements de sa part.

VI. Bibliographie :

[1] http://ec.europa.eu/transparency/regdoc/rep/1/2001/FR/1-2001-172-FR-F1-1.Pdf (Dernier accès le 23/06/2018)

[2] Abu Bakr Soliman, Kareem Eissa, Samhaa R. El-Beltagy, AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP, Procedia Computer Science,Volume 117 ,Pages 256-265, 2017.

[3]Steven Burrows, Iryna Gurevych, and Benno Stein. The Eras and Trends of Automatic Short Answer Grading. International Journal of Artificial Intelligence in Education, 25.1. 2015.

[4] R. Siddiqi, C. J. Harrison and R. Siddiqi, "Improving Teaching and Learning through Automated Short-Answer Marking," in IEEE Transactions on Learning Technologies, vol. 3, no. 3, pp. 237-249, July-Sept. 2010.

[5] Wing, J. M. Computational thinking. Communications of the ACM, 49(3), 33–33. doi:10.1145/1118178.1118215, (2006).

[6] Bloom taxonomie : A taxonomy for learning, teaching, and assessing : a revision of Bloom's taxonomy of educational objectives. New York : Longman, c2001

[7] Wachsmuth, H., Stein, B., and Engels, G. Constructing Efficient Information Extraction Pipelines. In B. Berendt, A. de Vries, W. Fan, C. MacDonald, I. Ounis, and I. Ruthven, editors, Proceedings of the Twentieth ACM International Conference on Information and Knowledge Management, CIKM '11, pages 2237–2240, Glasgow, Scotland. ACM, (2011).

[8] Wachsmuth, H., Stein, B., and Engels, G. Information Extraction as a Filtering Task. In Q. He and A. Iyengar, editors, Proceedings of the Twenty-Second ACM International Conference on Information and Knowledge Management, CIKM '13, pages 2049–2058, San Francisco, California. ACM, (2013).

[9] Valenti, S., Neri, F., and Cucchiarelli, AAn. Overview of Current Research on Automated Essay Grading.Journal of Information Technology Education, 2, 319–330, (2003).

[10] Pérez-Marín, D., Pascual-Nieto, I., and Rodríguez, P. Computer-Assisted Assessment of Free-Text Answers. The Knowledge Engineering Review, 24(4), 353–374, (2009).

[11] Burstein, J., Kaplan, R., Wolff, S., and Lu, C. Using Lexical Semantic Techniques to Classify Free Responses. In E. Viegas, editor, Proceedings of the ACL SIGLEX Workshop on Breadth and Depth of Semantic Lexicons, pages 20–29, Santa Cruz, California. Association for Computational Linguistics, (1996).

[12] Nielsen, R. D., Ward, W., Martin, J. H., and Palmer, M. Annotating Students' Understanding of Science Concepts. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, and D. Tapias, editors, Proceedings of the Sixth International Conference on Language Resources and Evaluation, pages 1–8, Marrakech, Morocco. European Language Resources Association, (2008a).

[13] Leacock, C. and Chodorow, M. C-rater: Automated Scoring of Short-Answer Questions. Computers and the Humanities, 37(4), 389–405, (2003).

[14] Jordan, S. Investigating the Use of Short Free Text Questions in Online Assessment. Final project report,Centre for the Open Learning of Mathematics, Science, Computing and Technology, The Open University,Milton Keynes, United Kingdom, (2009).

[15] Sukkarieh, J. Z. and Blackmore, J. c-rater: Automatic Content Scoring for Short Constructed Responses. In H. C. Lane and H. W. Guesgen, editors, Proceedings of the Twenty-Second International Conference of the Florida Artificial Intelligence Research Society, pages 290–295, Sanibel Island, Florida. AAAI Press, (2009).

[16] Sukkarieh, J. Z. and Stoyanchev, S .Automating Model Building in c-rater. In C. Callison-Burch and F. M. Zanzotto, editors, Proceedings of the First ACL/IJCNLP Workshop on Applied Textual Inference,TextInfer '09, pages 61–69, Suntec, Singapore. Association for Computational Linguistics, (2009).

[17] Myroslava Dzikovska, Peter Bell, Amy Isard, and Johanna D. Moore. Evaluating language understanding accuracy with respect to objective outcomes in a dialogue system. In EACL, pages 471–481. The Association for Computer Linguistics, 2012.

[18] Mitchell, T., Russell, T., Broomhead, P., and Aldridge, N. Towards Robust Computerised Marking of Free Text Responses. In Proceedings of the Sixth Computer Assisted Assessment Conference, pages 233–249, Loughborough, United Kingdom, (2002).

[19] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. Indexing by latent semantic analysis. JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE, 41(6) :391407, (1990).

[20] Bukai, O., Pokorny, R., and Haynes, J. An Automated Short-Free-Text Scoring System: Development and Assessment. In Proceedings of the Twentieth Interservice/Industry Training, Simulation, and Education Conference, pages 1–11. National Training and Simulation Association, (2006).

[21] Bailey, S. and Meurers, D.. Diagnosing Meaning Errors in Short Answers to Reading Comprehension Questions. In J. Tetreault, J. Burstein, and R. De Felice, editors, Proceedings of the Third ACL Workshop on Innovative Use of NLP for Building Educational Applications, pages 107–115, Columbus, Ohio. Association for Computational Linguistics, (2008)

[22] Dzikovska, M. O., Nielsen, R. D., Brew, C., Leacock, C., Giampiccolo, D., Bentivogli, L., Clark, P., Dagan, I., and Dang, H. T. SemEval-2013 Task 7: The Joint Student Response Analysis and Eighth Recognizing Textual Entailment Challenge. In M. Diab, T. Baldwin, and M. Baroni, editors, Proceedings of the Second Joint Conference on Lexical and Computational Semantics, pages 1–12, Atlanta, Georgia, (2013).

[23] Prettenhofer, P. and Stein, B. Cross-Lingual Adaptation using Structural Correspondence Learning. Transactions on Intelligent Systems and Technology, 3, 13:1–13:22, (2011).

[24] Negre, Elsa. Comparaison de textes: quelques approches, (2013).

[25] Huang, A.Y. Similarity Measures for Text Document Clustering, (2008).

[26] Wael H Gomaa and Aly A Fahmy. Article: A Survey of Text Similarity Approaches. International Journal of Computer Applications 68(13):13-18, April 2013.

[27] Mihalcea, R., Corley, C. & Strapparava, C. Corpus based and knowledge-based measures of text semantic similarity. In Proceedings of the American Association for Artificial Intelligence.(Boston, MA), (2006)

[28] Islam, A., & Inkpen, D. Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data, 2(2), 1–25, (2008).

[29] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representation of Words and Phrases and their Compositionality. In: NIPS: Proceedings of Neural Information Processing Systems Nevada, United States, pp. 3111–3119 (2013)

[30] M. Naili, A. H. Chaibi, dan H. H. Ben Ghezala, "Comparative Study of Word Embedding Methods in Topic Segmentation," Procedia Computer Science, Vol. 112, hal. 340–349, 2017.

[31] Efficient Estimation of Word Representations in. Vector Space. Tomas Mikolov. Google Inc., Mountain View, CA

[32] L. Ouahrani, String similarity for Arabic short answer grading, internal report, LIMPAF/118, the LIMPAF Laboratory, Bouira University, January 2018.

[33] M. M. A. Alqahtani and E. Atwell, "A Review of Semantic Search Methods to Retrieve Information from the Qur'an Corpus," 2015.

[34] K. F. Shaalan, M. Attia, P. Pecina, Y. Samih, and J. van Genabith, "Arabic Word Generation and Modelling for Spell Checking," in LREC, pp. 719-725, 2012.

[35] K. F. Shaalan, M. Attia, P. Pecina, Y. Samih, and J. van Genabith, "Arabic Word Generation and Modelling for Spell Checking," in LREC, pp. 719-725, 2012.

[36] I. Hajeer, "Comparison on the Effectiveness of Different Statistical Similarity Measures," International Journal of Computer Applications, vol. 53, no. 8, 2012.

[37] H. Khafajeh, N. Yousef, and G. Kanaan, "Automatic query expansion for arabic text retrieval based on association and similarity thesaurus," in Proceedings he European, Mediterranean & Middle Eastern Conference on Information Systems (EMCIS), Abu Dhabi, UAE, 2010.

[38] A. O. Al-Thubaity, "A 700M+ Arabic corpus: KACST Arabic corpus design and construction," Language Resources and Evaluation, vol. 49, no. 3, pp. 721-751, 2015.

[39] Gomaa, W. H., & Fahmy, Automatic scoring for answers to Arabic test questions. Computer Speech & Language. Elsevier Ltd 2013.

[40] Gomaa, W. H., & Fahmy, Arabic Short Answer Scoring with Effective Feedback for Students. International Journal of Computer Applications (0975 – 8887) Volume 86 – No 2, January 2014.

[41] P. Kolb, "Disco: A multilingual database of distributionally similar words," Proceedings of KONVENS-2008, Berlin, 2008.

[42] D. Cera, M. Diabb, E. Agirrec, I. Lopez-Gazpioc, and L. Speciad, SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Cross-lingual Focused Evaluation, Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 1–14, Vancouver, Canada, August 3 - 4, 2017.

[43] A. Magooda, M.. Zahran , M. Rashwan, H. Raafat, M. Fayek, Vector Based Techniques for Short Answer Grading, Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, Key Largo, Florida, May 16-18 , 2016.

[44] E. Nagoudi, D. Schwab, Semantic Similarity of Arabic Sentences withWord Embeddings, Proceedings of The Third Arabic Natural Language Processing Workshop (WANLP), pages 18–24, Valencia, Spain, April 3, 2017

[45] Lovins, Julie B. Development of a stemming algorithm.Cambridge: MIT Information Processing Group, Electronic Systems Laboratory, 1968.

[46] D. Cera, M. Diabb, E. Agirrec, I. Lopez-Gazpioc, and L. Speciad, SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Cross-lingual Focused Evaluation, Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 1–14, Vancouver, Canada, August 3 - 4, 2017.

[47] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Proceedings of EMNLP 2015.

[48] Zahran, Mohamed & Magooda, Ahmed & Mahgoub, Ashraf & Raafat, Hazem & Rashwan, Mohsen & Atyia, Amir. Word Representations in Vector Space and their Applications for Arabic, (2015).

[49] Mohammad, Abu Bakr & Eissa, Kareem & El-Beltagy, Samhaa. (2017). AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. Procedia Computer Science, 117. 256-265.

[50] https://drive.google.com/drive/folders/1lpzvYO8K8CEu-e69nVKWRWVmzT8yZE0A (Dernier accès le 23/06/2018)

[51] https://github.com/bakrianoo/aravec/tree/master/AraVec%201.0 (Dernier accès le 23/06/2018)

[52] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. In: ICLR: Proceeding of the International Conference on Learning Representations Workshop Track, Arizona, USA, pp. 1301–3781 (2013)

[53] http://opus.lingfil.uu.se/ (Dernier accès le 23/06/2018)

[54] Tiedemann, J.: Parallel Data, Tools and Interfaces in OPUS. In: LREC: Proceedings of the 8th International Conference on Language Resources and Evaluation, Istanbul, Turkey, pp. 2214–2218 (2012)

[55] Raafat, H., Zahran, M., Rashwan, M.: Arabase A Database Combining Different Arabic Resources with Lexical and Semantic Information. In: Proceeding of KDIR is part ofIC3K, The International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, Portugal, pp. 233–240 (2013)

[56] Eisele, A., Chen, Y.: MultiUN: A Multilingual corpus from United Nation Documents. In: LREC: Proceeding of the International Conference on Language Resources and Evaluation, Valletta, Malta, pp. 17–23 (2010)

[57] http://www.opensubtitles.org/ (Dernier accès le 23/06/2018)

[58] http://tanzil.net/download/ (Dernier accès le 23/06/2018)

[59] Tiedemann, J.: News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In (RANLP): Recent Advances in Natural Language Processing, pp. 237–248. John Benjamins, Amsterdam (2009)

[60] https://sites.google.com/site/mouradabbas9/corpora (Dernier accès le 23/06/2018)

[61] Saad, M.K., Ashour, W.: OSAC: Open Source Arabic Corpus. In: EEECS: the 6th International Symposium on Electrical and Electronics Engineering and Computer Science, European University of Lefke, Cyprus, vol. 10 (2010)

[62] https://github.com/anastaw/Meedan-Memory (Dernier accès le 23/06/2018)

[63] http://ksucorpus.ksu.edu.sa/ar/ (Dernier accès le 23/06/2018)

[64] https://en.wikipedia.org/wiki/List_of_Wikipedias (Dernier accès le 23/06/2018)

[65] https://sites.google.com/site/mouradabbas9/corpora (Dernier accès le 23/06/2018)

[66] https://sourceforge.net/projects/ar-text-mining/files/Arabic-Corpora/ (Dernier accès le 23/06/2018)

[67] https://github.com/MiladAlshomary/light10stemmer (Dernier accès le 23/06/2018)

[68] https://github.com/motazsaad/khoja-stemmer-command-line/blob/master/khoja/KhojaStem.java (Dernier accès le 23/06/2018)

[69] https://www.nltk.org/_modules/nltk/stem/isri.html (Dernier accès le 23/06/2018)

[70] https://pypi.org/project/Tashaphyne/ (Dernier accès le 23/06/2018)

[71] https://github.com/motazsaad/arabic-light-stemmer (Dernier accès le 23/06/2018)

[72] http://arabicstemmer.com/ (Dernier accès le 23/06/2018)

[73] https://stanfordnlp.github.io/CoreNLP/ (Dernier accès le 23/06/2018)

[74] Gerard Salton, Christopher Buckley,Term-weighting approaches in automatic text retrieval, Information Processing & Management, Volume 24, Issue 5, Pages 513-523, 1988.

[75] Bojan Furlan, Vuk Batanović, Boško Nikolić, Semantic similarity of short texts in languages with a deficient natural language processing support, Decision Support Systems, Volume 55, Issue 3, Pages 710-719, 2013.

[76] Nagoudi, El Moatez Billah & Schwab, DidierSemantic Similarity of Arabic Sentences with Word Embeddings, (2017).

[77] Mihalcea, R., Corley, C., Strapparava, C. Corpus-based and knowledge-based measures of text semantic similarity. In: Proceedings of the National Conference on Artificial Intelligence, July, vol. 21, no. 1. AAAI Press/MIT Press, Menlo Park, CA/Cambridge, MA/London,p. 775, 2016.

[78] J. B. MacQueen « Some Methods for classification and Analysis of Multivariate Observations » Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1: 281–297 p, (1967).

[79] https://github.com/llimllib/ckmeans (Dernier accès le 23/06/2018)

[80] Altman, N. S. "An introduction to kernel and nearest-neighbor nonparametric regression". The American Statistician. 46 (3): 175–185, (1992).

[81] Jacob Cohen, Statistical power analysis for the behavioral sciences (2nd ed.). (1988).

[82] William H Greene, Econométrie, Paris, Pearson Education, 5e éd. (ISBN 978-2-7440-7097-6), 2005.

[83] https://sourceforge.net/projects/kacst-acptool/?source=typ_redirect ? (Dernier accès le 23/06/2018)

VII. Annexes

i. Les mots manquants des modules de Zahran :

- Sans Stem :

Dataset Gomaa: 5 mots manquants
['قيقل', 'التوزان', 'يتممد', 'فيزيايئه', 'الكلورقيل']

Dataset Sts2017 250 : 9 mots manquants
['الدايتونا', 'كاحلان', 'السكسيفون', 'ويجيلز', 'ذقونهما', 'تلتقطن', 'تنزهما', 'تتصتع', 'بالشقبله']

Dataset Sts2017 368 : 1 mot manquant
['نيئتان']

- Heavy Stem:

Dataset Gomaa: 7 mots manquants
['كخذ', 'نيتروجيه', 'قيقل', 'يتممد', 'فيزيايئه', 'كلورقيل', 'جائرللحيوانات']

Dataset Sts2017 250 : 10 mots manquants
['للسباحهحالا', 'يقفزالى', 'ويجيلز', 'يقفزعلى', 'تنزهما', 'قطاريتطلع', 'تتصتع', 'شقبله', 'تسيرعلي', 'غيرمطلي']

Dataset Sts2017 368 : 4 mots manquants
['خيارالى', 'تقشرحبه', 'تقشربرتقاله', 'يرقصعلى']

- Light Stem :

Dataset Gomaa: 10 mots manquants
['وقايت', 'السلالسل', 'قيقل', 'التوزان', 'يتممد', 'التشاب', 'يزيايئه', 'احتوائ', 'الكلورقيل', 'الجائرللحيوانات']

Dataset Sts2017 250 : 14 mots manquants
['الدايتونا', 'قيثارت', 'لسباحهحالا', 'السكسيفون', 'يقفزالى', 'يجيلز', 'يقفزعلى', 'تلتقطن', 'تنزهما', 'قطاريتطلع', 'تتصتع', 'الشقبله', '
تسيرعلي', 'غيرمطلي']

Dataset Sts2017 368 : 6 mots manquants
['قيثارت', 'نيئتان', 'الخيارالى', 'تقشرحبه', 'تقشربرتقاله', 'يرقصعلى']

ii. Les mots manquants des modules d'araVec:

- Sans stem

Dataset Gomaa : 100 mots manquants

['ليتكيف', 'المقومات', 'اهدارها', 'ووقايتها', 'ئودي', 'البلانكتون', 'تتنتج', 'مترممه', 'لتحللها', 'اضراراها', 'الموقعيه', 'كالضوء', 'المؤثر', 'الاظلام', 'الاصي', 'هائمات', 'صعيره', 'العلاقت', 'فضلاته', 'يتمز', 'بالتعقيدات', 'الاستضاءه', 'فالاشعه', 'يتغلل', 'موجاته', 'فتمتص', 'لزرقه', 'بتحررها', 'ترسبها', 'الهائمات', 'البوتين', 'تكويت', 'بهيكلها', 'كالجراد', 'بولها', 'يشح', 'عرقها', 'كالزواحف', 'وتختبئ', 'باغطيه', 'تشح', 'للاستفاده', 'يطوره', 'النيتروجيه', 'لعناصرها', 'النتيروجين', 'خضريا', 'التجرثم', 'التحوصل', 'الجرائيم', 'بالبيات', 'للبيات', 'اجسماها', 'توزان', 'ازدباد', 'فتهلك', 'تتحل', 'قيقل', 'تضار', 'المستساغه', 'للتوزان', 'التوزان', 'فتعيد', 'كالكربون', 'فتظل', 'فتحمي', 'الاوكسينات', 'فينحني', 'لانحناء', 'التواقت', 'والتواقت', 'بنقصانها', 'فيطفو', 'يتمدد', 'الترابط', 'فيزيايئه', 'ليتنفسه', 'فالطحالب', 'لانفصالها', 'كالمناخ', 'ثلاجات', 'تخزنها', 'تسربها', 'فيتوفر', 'سيتحمل', 'بقدار', 'بيما', 'كاكلات', 'فتحتل', 'الكلورقيل', 'لوجدود', 'لتمتص', 'لاتها', 'فيوفر', 'جائرا', 'لحيواناتهم', 'تسئ', 'لمخلفات', 'بتلويث', 'التوزان"]

Dataset Sts2017 250 : 133 mots manquants

['الدايتونا', 'يمشين', 'متجاورات', 'يتسكعون', 'تهرول', 'وقيثاره', 'مرتديان', 'يحدقون', 'قيثارته', 'بتفحص', 'كلبان', 'فمم', 'بقطه', 'مبتسمه', 'دراجتهم', 'بالبالونات', 'كاحلان', 'مكسوران', 'بالملصقات', 'تحضران', 'بلعبته', 'يفرم', 'يصليان', 'يتعرق', 'فطورا', 'ولصديقها', 'بقبعات', 'يطاردان', 'ارنبا', 'قطتان', 'تطاردان', 'تخيط', 'يتسوق', 'تاكلان', 'يتسوقون', 'التوبا', 'السكسيفون', 'يصافح', 'صاحبهما', 'يتناولان', 'ترتري', 'تمتطي', 'لعارض', 'يتقافز', 'سروالا', 'ويجيلز', 'مقلما', 'معجبون', 'بنافوره', 'تريفي', 'متظاهره', 'مزلجه', 'يثب', 'ترامبولين', 'غيتاره', 'وبسعاده', 'وورديا', 'تسبحن', 'ينظيف', 'الطوبيه', 'اصابعهن', 'وايديهما', 'ذقونهما', 'يلوذون', 'ينقلان', 'لعضهما', 'تلتقطن', 'يلتقطان', 'تنزهما', 'غواصين', 'حافلتين', 'ترتدين', 'بستره', 'يمسكان', 'يقذفون', 'المقتربه', 'يركبان', 'يجدفون', 'يزيلون', 'غمرته', 'بالغان', 'يتقاتلان', 'يترنح', 'دهسه', 'يتدربان', 'مارثون', 'وصحبهما', 'يراقبها', 'يشيحون', 'اسيويان', 'يقظ', 'نفثت', 'لصورتها', 'تتمرن', 'بخدعه', 'يتارجحان', 'مراهقان', 'ينبش', 'نفض', 'مستعدات', 'بالشقلبه', 'يجتازون', 'تتعدوان', 'تقيات', 'بمزمار', 'وطفلان', 'يتهامس', 'يرقبهما', 'تتصتع', 'لكاميره', 'انتهيا', 'لتوهما', 'بيومها', 'خيولها', 'مزعور', 'بعصبيه', 'باناه', 'يركل', 'يغفو', 'يقومن', 'بالشقبله', 'الصغيرتان', 'تقفزان', 'يقران', 'تقران', 'تسفع', 'يسفع', 'جالسات', 'بمشروب', 'صخرتين', 'يتزلج', 'كميرات', 'زنجبيلا', 'تفرم', 'ثوما', 'تفرك', 'بفرشاه', 'بلاقط', 'تجدل', 'للمقلاه', 'قشرت', 'يكبل', 'يفرم', 'الكلبين', 'بلاستيكيا', 'بكفه', 'اليمونا', 'يلاعب', 'ارنبا', 'كاراتي', 'تعصر', 'انغمست', 'ببروني', 'البببروني', 'يغيظ', 'بالفار', 'يشطف', 'غيتارا', 'يسرن', 'تتمرن', 'طماطما', 'يدفق', 'بالشوكه', 'تغادرن', 'لريهانا', 'تسترخي', 'بصلا', 'تدلك', 'ينش', 'التورتيا', 'المخفوقه', 'ضربان', 'عصيرا', 'منزعج', 'الجايم', 'بهاتف', 'تتحدثان', 'حليبا', 'يلعق', 'يطعمن', 'فرما', 'قفصهم', 'يملء', 'بمكياج', 'بتقلبات', 'لنمله', 'تبري', 'يلدغ', 'غيتاره', 'السباحون', 'يغوصون"]

- Heavy Stem

Dataset Gomaa : 95 mots manquants

['هتم', 'خرا', 'صفف', 'خرع', 'ضور', 'قصا', 'قوع', 'صحر', 'سقر', 'ئودي', 'كلوروفيل', 'سلسل', 'بلانكتون', 'حشش', 'كخذ', 'دقق', 'جهي', 'سوع', 'قتر', 'نوخ', 'عوم', 'الاصي', 'كرث', 'صعر', 'استضاءه', 'وشع', 'لشي', 'غلل', 'فتمتص', 'حوج', 'لقع', 'لتت', 'دمس', 'غوط', 'يرابيع', 'ثعب', 'حوس', 'حود', 'سفي', 'شعع', 'ورل', 'شحح', 'باغطيه', 'نيتروجيه', 'وبا', 'ركم', 'لعناصرها', 'ختل', 'نتيروجين', 'مصص', 'لكا', 'جرثم', 'حوصل', 'خمل', 'جرائيم', 'جسا', 'لحف', 'ازدباد', 'بخر', 'سلالسل', 'تتحل', 'قيقل', 'سوغ',

'فتظل', 'اوكسينات', 'فينحني', 'دفا', 'طفا', 'فيطفو', 'يتممد', 'فيزيائيه', 'غوز', 'كسد', 'عطي', 'قوا', 'بلطيق', 'فيتوفر', 'بيما', 'كاكلات',
"فتحتل', 'كلورقيل', 'تمم', 'خرر', 'ذبل', 'جفف', 'يقت', 'لتمتص', 'لاتها', 'نتح', 'فيوفر', 'طنن', 'جائرللحيوانات', 'رهب', 'تسئ', 'خذا']

Dataset Sts2017 250 : 108 mots manquants

زلج', 'فتا', 'قمص', 'سكع', 'هرول', 'ازق', 'وسق', 'نوص', 'وقيثاره', 'قبع', 'فنن', 'يهما', 'رقب', 'دكن', 'حدق', 'قيثارته', 'زيي', "
'دغل', 'فمم', 'للسباحهحالا', 'قوط', 'سقل', 'تشف', 'نكر', 'قوا', 'بسا', 'فرم', 'وكب', 'خرا', 'رمد', 'مهي', 'سوح', 'نفض', 'ندل', 'ارنبا',
'فتت', 'ذرا', 'وبا', 'كسف', 'عجن', 'وخي', 'رقق', 'ترتري', 'مطي', 'هرا', 'يقفزالي', 'خوذ', 'سوع', 'فقع', 'طفا', 'خرر', 'ويجيلز',
'يحر', 'رهق', 'يقفزعلي', 'ترامبولين', 'حوب', 'غيتاره', 'نظف', 'صبع', 'الوذ', 'فرر', 'قما', 'نوق', 'عرك', 'شبب', 'تنزهما', 'دهي',
'نزه', 'ترتدين', 'تنانير', 'انث', 'رمث', 'جدف', 'ثنن', 'رنح', 'زجج', 'مرث', 'كيي', 'يقظ', 'قطاريتطلع', 'جهي', 'دفا', 'جمهر', 'لكا',
'شقلب', 'قيا', 'عوم', 'تتصتع', 'لكاميره', 'تلفز', 'نطل', 'زفف', 'زعر', 'يغفو', 'شقبله', 'سرر', 'يقران', 'تقران', 'سفع', 'صفف', 'ذرع',
"تسيرعلي', 'لبط', 'حوك', 'حدر', 'زمل', 'غيرمطلي']

Dataset Sts2017 368 : 101 mots manquants

تظيف', 'زلج', 'نوص', 'قبع', 'حوب', 'بطط', 'كرث', 'قيثارته', 'صفف', 'رزا', 'بروكولي', 'سوج', 'نظف', 'فتا', 'دجج', 'قوط', 'مسر', "
'قمص', 'لثم', 'خيارالي', 'رقق', 'حصا', 'ميكرفون', 'خرر', 'رصص', 'عوم', 'تقشرحبه', 'ماكياجا', 'جدف', 'خرا', 'حدق', 'رقا',
'صندوق', 'لحف', 'بطخ', 'ثنن', 'نشف', 'تقشربرتقاله', 'هرا', 'ثعب', 'ضلل', 'قنبل', 'صحر', 'سرر', 'نبح', 'كترونيه', 'غوط', 'فصي',
'جثم', 'فلت', 'باذنجان', 'وسق', 'زنجبيلا', 'رذل', 'فرم', 'ثوي', 'ذرع', 'جفف', 'ترامبولين', 'شمم', 'عجن', 'شفر', 'ظلل', 'بطق', 'ستت',
'بلاستيكيا', 'قما', 'يرقصعلي', 'رقب', 'صلص', 'ارنبا', 'فنن', 'غيتارا', 'نزه', 'سفي', 'قور', 'طماطما', 'لريهانا', 'كزبر', 'جمبري',
'ذبب', 'بروكلي', 'تورتيا', 'شعل', 'بسا', 'كتروني', 'صبع', 'زعج', 'جايم', 'جهي', 'فتت', 'رفا', 'قيثار', 'بمكياج', 'عطي', 'قنع', 'قعي',
"زيي', 'رغف', 'ورس', 'غيتاره']

- Light Stem

Dataset Gomaa : 73 mots manquants
اقي', 'لحفاظ', 'وقايت', 'لاستفاده', 'تؤدي', 'غذائ', 'البلانكتون', 'تتنتج', 'امثلت', 'مترممه', 'الموقعيه', 'لضو', 'الاظلام', 'لتكيف', "
'الاصي', 'لبحار', 'هائمات', 'يقان', 'صعيره', 'العلاقت', 'يتمز', 'الاستضاءه', 'يتغلل', 'الهائمات', 'البوتين', 'تكويت', 'لاحتفاظ', 'يشح',
'لمساهمه', 'تشح', 'اهميت', 'النيتروجيه', 'النتيروجين', 'خضريا', 'لازهار', 'التجرثم', 'التحوصل', 'الجرائيم', 'لبيات', 'اجسما', 'توزان',
'ازدباد', 'السلالسل', 'تتحل', 'قيقل', 'تضار', 'لغطاء', 'المستساغه', 'لتوزان', 'فتحمي', 'الاوكسينات', 'التواقت', 'نقصانها', 'كثافت',
'فيطفو', 'يتممد', 'التشاب', 'يزيائيه', 'ثلاجات', 'زيادت', 'قدار', 'بيما', 'احتوائ', 'كاكلات', 'فتحتل', 'الكلورقيل', 'وجدود', 'زراعت',
"جائرا', 'الجائرللحيوانات', 'سسب', 'تسئ', 'التوزان']

Dataset Sts2017 250 : 124 mots manquants
الدايتونا', 'يمشين', 'متجاورات', 'يتسكعون', 'تهرول', 'مرتديان', 'يحدقون', 'قيثارت', 'كلبان', 'لسباحهحالا', 'مبتسمه', 'دراجت', "
'احلان', 'مكسوران', 'تحضران', 'يفرم', 'يصليان', 'يتعرق', 'المطب', 'يطاردان', 'ارنبا', 'قطتان', 'تطاردان', 'تخيط', 'يتسوق',
'زملائ', 'تاكلان', 'يتسوقون', 'التوبا', 'السكسيفون', 'يصافح', 'يتناولان', 'ترتري', 'تمتطي', 'يقفزالي', 'لسباحه', 'ساعت', 'يتقافز',
'سروالا', 'يجيلز', 'مقلما', 'معجبون', 'بنافوره', 'تريفي', 'متظاهره', 'يقفزعلي', 'مزلجه', 'لنفخ', 'يثب', 'ترامبولين', 'وورديا',
'تسبحن', 'ينظيف', 'الطوبيه', 'ذقون', 'يلوذون', 'ينقلان', 'لعض', 'تلتقطان', 'يلتقطان', 'تنزهما', 'غواصين', 'حافلتين', 'ترتدين',
'بستره', 'يمسكان', 'نتان', 'صديقت', 'يقذفون', 'المقتربه', 'يركبان', 'يجدفون', 'يزيلون', 'الغان', 'يتقاتلان', 'يترنح', 'الدت', 'لجري',
'يتدربان', 'مارثون', 'يشيحون', 'خلفيت', 'اسيويان', 'يقظ', 'قطاريتطلع', 'نفثت', 'تتمرن', 'يتارجحان', 'مراهقان', 'ينبش', 'نفض',
'يرا', 'مستعدات', 'الشقلبه', 'يجتازون', 'تعدوان', 'تقيات', 'يتهامس', 'تتصتع', 'انتهيا', 'توهما', 'مزعور', 'اناه', 'يركل', 'يغفو',
'يقومن', 'الشقبله', 'الصغيرتان', 'تقفزان', 'يقران', 'تقران', 'تسفع', 'يسفع', 'جالسات', 'شاح', 'رقبت', 'تسيرعلي', 'لدراجات',
"صخرتين', 'غرفت', 'يتزلج', 'قفزا', 'ممسوكه', 'غيرمطلي']

Dataset Sts2017 368 : 96 mots manquants

['تظيف', 'متزلج', 'يتزلج', 'مفاتيحه', 'قيثارت', 'صقف', 'يرقصان', 'تتسوقان', 'يتبل', 'بيضتان', 'نينتان', 'عاء', 'يقشر', 'البروكولي', 'تمشط', 'يقرص', 'المقشره', 'تلعق', 'كفها', 'لطبق', 'يركل', 'سقطع', 'تنقر', 'الخيارالي', 'يصافح', 'تقشرحبه', 'ماكياجا', 'تجدف', 'لفلا', 'شرائحا', 'صندوق', 'رقا', 'ممسكان', 'تقشربرتقاله', 'مزمارا', 'قدورا', 'البوليود', 'زهورا', 'ليمونه', 'بغاء', 'ينبح', 'تتبل', 'فصا', 'كميرات', 'زنجبيلا', 'تفرم', 'ثوما', 'تفرك', 'لتجمد', 'بلاقط', 'تجدل', 'لمقلاه', 'قشرت', 'يكبل', 'يفرم', 'الكلبين', 'بلاستيكيا', 'ليمونا', 'يرقصعلي', 'يلاعب', 'ارنبا', 'كاراتي', 'تعصر', 'انغمست', 'البيروني', 'يغيظ', 'يشطف', 'دراجت', 'غيتارا', 'يسرن', 'تتمرن', 'طماطما', 'يدفق', 'تغادرن', 'ريهانا', 'تسترخي', 'صلا', 'تدلك', 'ينش', 'التورتيا', 'المخفوقه', 'ضربان', 'عصيرا', 'منزعج', 'الجايم', "بهاتف", 'تتحدثان', 'حليبا', 'يلعق', 'يطعمن', 'فرما', 'يملء', 'تبري', 'يلدغ', 'السباحون', 'يغوصون']

iii. Liste des stops words de Khoja :

التي | كذلك | تلك | وكان | على | أحد | وليس | به | يكون | وهو | حتى | من | في | الى | يلي | ضد | بعد | ان | وكانت | ليسب | لا | ومن | حين | أما | الذي | منذ | ليس | مساء | عن| لكن | وعلى | إن | عليها | فيها | وبين | جدا | بين | قد | تكون | أنه | هذه | ثم | فقط | والتي | هذا | له | ولكن | لكنه | مع | دون | حول | عنه | ما | أي | أن | وثي | لدى | بد | كل | اللذين | عند | لو | ذلك | فيه | فإن | هؤلاء | لم | اليوم | لأن | لهم | كان | نحو | لن | وقد | هنا | كيف | كما | عليه| علي | إذ | أو | لها | تحت | فهو | وفي | بها | منه | عنها | هو | بل | فقد | ومع | الي | لا | ما | و | او | اذا| هي | حيث | هل | إذا | إلى | منها | يوم | معه | قبل | هناك | أمام | لذلك | كانت | مافتئ | مابرح | ظل | اضحى | أضحى | امسى | أمسى | أصبح | اصبح | مايزال | لايزال | لازال | مازال | إلي | اي | ذات | وله | اول | ضمن | الحالي | ولايزال | لاسيما | لعل | ليت | كأن | إن | ليس | صار | بات | ماانفك | يمكن | اليه | الذي | بإن | أبو | مما | ستكون | فكان | الا | لهذا | هذا | والذي | وان | فانه | الذين | انه | اليها| بدلا | الذى | هن | الذي | آل | وأبو | وهي | وأن | لدي | بهذا |

iv. Les couples (note manuelle, note générée) STS 250:

Stem : Heavy stem
Model : Zahran CBOW
Modèles de calculs : MatSim P.Mixte
Calcul de score : 11-means
N°R → Numéro de la réponse, N.M → Note manuelle, N.G → Note générée par ordinateur

N°R	N.M	N.G	N°R	N.M	N.G	N°R	N.M	N.G	N°R	N.M	N.G	N°R	N.M	N.G	N°R	N.M	N.G
1	0.8	0.5	43	3.2	4.0	85	5.0	5.0	127	1.6	3.0	169	4.0	1.5	211	4.2	3.5
2	1.0	2.5	44	2.8	3.5	86	4.0	4.5	128	4.4	4.0	170	0.6	1.0	212	2.4	1.5
3	2.6	0	45	5.0	5.0	87	0.2	0.5	129	3.6	2.5	171	0.6	0.5	213	1.2	1.5

4	2.2	2.5	46	5.0	5.0	88	3.4	2.0	130	0.2	0.5	172	1.8	2.5	214	3.6	2.0
5	1.4	1.0	47	1.4	3.5	89	4.6	4.0	131	1.8	4.0	173	4.0	4.0	215	1.0	2.5
6	1.8	1.0	48	2.6	3.0	90	5.0	4.0	132	1.6	3.0	174	1.4	1.0	216	5.0	5.0
7	0.4	0	49	3.8	2.5	91	1.6	1.5	133	0.2	0.5	175	1.4	2.0	217	0.4	1.5
8	1.4	0.5	50	1.6	1.0	92	1.8	4.5	134	0.6	0.5	176	2.2	2.5	218	1.0	0.5
9	1.0	1.0	51	0.6	2.0	93	0.2	1.5	135	3.6	2.0	177	3.6	2.5	219	1.6	1.5
10	2.8	4.0	52	2.0	1.0	94	0.2	0.5	136	1.8	3.5	178	3.8	3.5	220	0.2	1.5
11	1.0	3.0	53	0.6	1.5	95	3.8	4.0	137	1.0	1.0	179	5.0	2.5	221	0.8	0
12	1.6	2.0	54	4.0	4.0	96	3.4	3.0	138	2.2	2.5	180	3.2	2.0	222	2.6	2.5
13	1.8	2.0	55	5.0	5.0	97	0.2	1.5	139	3.8	2.5	181	1.0	0.5	223	1.0	2.0
14	0.2	1.0	56	3.6	4.0	98	4.4	4.5	140	3.2	1.5	182	1.8	1.0	224	3.2	3.0
15	3.2	3.0	57	4.0	4.0	99	1.4	3.5	141	1.8	2.5	183	3.6	4.0	225	3.2	3.0
16	2.0	2.0	58	3.8	3.0	100	1.8	1.5	142	3.4	3.5	184	3.8	3.0	226	3.2	2.0
17	1.0	1.5	59	3.2	3.5	101	3.4	4.0	143	2.8	1.0	185	2.0	3.0	227	0.4	1.5
18	1.4	1.5	60	2.0	2.5	102	0.2	0	144	4.0	4.0	186	3.8	1.5	228	0.4	0.5
19	5.0	4.5	61	3.4	4.0	103	2.8	4.5	145	0.8	1.5	187	0.2	2.0	229	0.2	0.5
20	4.0	2.5	62	0.4	0	104	4.8	2.5	146	2.6	2.5	188	1.0	0.5	230	4.6	5.0
21	1.0	4.5	63	5.0	4.5	105	2.4	3.0	147	4.0	2.5	189	1.2	1.0	231	1.0	1.0
22	1.6	1.5	64	2.2	2.5	106	1.2	1.0	148	2.4	3.0	190	0.8	0	232	4.2	4.5
23	3.2	2.5	65	4.6	5.0	107	2.4	4.5	149	3.6	2.0	191	0.6	1.0	233	0.4	0.5
24	0.6	0.5	66	0.2	0.5	108	2.2	2.5	150	0.8	1.0	192	0.8	0	234	1.4	2.5
25	0.2	0.5	67	1.8	4.0	109	5.0	4.5	151	5.0	4.0	193	4.4	5.0	235	2.0	2.5
26	2.6	2.0	68	4.0	4.0	110	0.2	0.5	152	1.2	0.5	194	1.4	1.0	236	2.6	2.0
27	3.6	4.0	69	3.2	2.5	111	0.8	1.0	153	2.8	4.5	195	1.2	2.0	237	2.8	4.0
28	1.8	1.5	70	4.2	3.0	112	0.6	2.5	154	2.4	2.0	196	0.6	1.0	238	0.4	1.5
29	2.0	1.0	71	0.0	1.0	113	1.6	2.5	155	3.2	1.5	197	2.4	4.5	239	2.6	4.0
30	1.0	1.5	72	4.2	3.0	114	1.8	1.0	156	1.4	1.0	198	1.6	0.5	240	2.6	2.0
31	0.2	1.0	73	2.0	3.5	115	2.6	3.0	157	5.0	5.0	199	0.8	0.5	241	1.0	1.5
32	2.6	0	74	3.8	3.5	116	5.0	5.0	158	1.8	2.0	200	3.2	3.0	242	0.2	1.5
33	0.8	1.5	75	4.2	4.5	117	1.0	0.5	159	4.0	4.0	201	1.6	5.0	243	2.4	3.5
34	1.2	2.0	76	5.0	2.0	118	2.2	3.5	160	2.4	4.0	202	0.2	0	244	3.8	4.5
35	3.0	1.5	77	4.2	3.0	119	0.4	1.5	161	1.2	1.5	203	1.4	3.0	245	0.6	1.0
36	1.2	0	78	1.6	1.0	120	2.0	2.5	162	3.8	3.5	204	0.2	1.0	246	1.6	1.5
37	1.0	1.5	79	0.0	0.5	121	3.4	2.5	163	1.0	2.0	205	2.0	0.5	247	1.4	3.0
38	5.0	5.0	80	4.6	5.0	122	3.8	1.5	164	1.0	2.0	206	1.6	3.0	248	0.2	0.5
39	3.4	4.0	81	0.2	1.0	123	3.0	1.5	165	1.2	3.0	207	0.2	1.0	249	1.2	3.5
40	4.4	2.5	82	3.0	3.5	124	1.6	3.0	166	0.8	1.5	208	2.8	5.0	250	0.2	0
41	2.4	1.5	83	1.6	1.0	125	1.4	2.0	167	3.0	2.0	209	2.0	2.5			
42	2.2	3.0	84	0.6	1.5	126	0.4	0.5	168	3.2	2.0	210	4.4	4.5			

v. Les couples (note manuelle, note générée) Gomaa :

Notes manuelles et Notes accordées par programme

Dataset : Gomaa
Stem : Light stem
Model : SkipGram
Modèles de calculs :
(Matrice d'ordre P.Mixte* 0.29 | Somme des vecteurs P.TFMINMAX *0.71) *0.81
Dice * 0.19
Calcul de score : 11-means
N°R → Numéro de la réponse, N.M → Note manuelle, N.G → Note générée par ordinateur

N° R	N.M	N.G	N° R	N.M	N.G	N° R	N.M	N.G	N° R	N.M	N.G	N° R	N.M	N.G	N° R	N.M	N.G
1	3.5	3.5	103	4.5	3.5	205	2.0	1.5	307	2.5	3.0	409	1.5	1.0	511	2.5	2.0
2	2.5	2.0	104	2.5	3.0	206	3.5	2.5	308	2.5	2.5	410	3.5	2.5	512	3.5	3.0
3	1.5	2.0	105	3.0	3.0	207	1.5	1.5	309	0.0	1.0	411	5.0	3.5	513	3.5	3.0
4	1.0	1.5	106	4.5	4.0	208	2.5	1.5	310	0.0	1.0	412	5.0	4.5	514	4.0	4.0
5	4.5	4.0	107	2.5	2.5	209	2.0	2.0	311	5.0	4.0	413	5.0	3.0	515	2.0	1.5
6	4.5	3.5	108	2.5	2.0	210	0.0	0.5	312	5.0	4.0	414	5.0	4.5	516	2.0	1.5
7	5.0	4.5	109	4.5	3.5	211	5.0	3.0	313	5.0	4.0	415	5.0	4.5	517	4.5	3.5
8	1.5	3.5	110	2.0	1.5	212	0.5	0	314	0.0	2.0	416	5.0	4.5	518	5.0	2.5
9	1.5	1.5	111	4.5	3.0	213	2.5	3.0	315	0.0	0.5	417	5.0	4.0	519	5.0	4.5
10	0.5	1.5	112	2.5	1.5	214	5.0	4.0	316	2.0	3.5	418	5.0	3.5	520	2.0	1.5
11	1.5	1.0	113	3.0	3.0	215	5.0	3.0	317	5.0	1.5	419	5.0	5.0	521	5.0	3.5
12	3.0	2.0	114	2.0	2.0	216	2.0	1.0	318	5.0	1.5	420	2.0	1.0	522	5.0	3.5
13	3.0	2.5	115	2.0	2.5	217	2.5	2.0	319	5.0	4.0	421	4.5	4.0	523	0.0	1.0
14	3.5	4.0	116	5.0	3.0	218	0.5	1.0	320	5.0	3.5	422	5.0	4.5	524	2.0	2.0
15	5.0	5.0	117	2.5	2.5	219	3.5	2.0	321	1.5	0.5	423	5.0	5.0	525	2.0	1.5
16	2.5	3.0	118	2.5	1.5	220	5.0	2.5	322	3.5	2.5	424	0.0	0	526	4.5	4.0
17	3.5	4.0	119	2.5	2.0	221	4.5	3.5	323	2.5	2.0	425	0.0	1.0	527	4.5	4.0
18	1.5	1.5	120	3.5	2.5	222	4.5	3.5	324	1.5	1.0	426	2.5	3.0	528	5.0	3.0
19	3.0	4.0	121	3.5	3.5	223	4.5	4.0	325	0.0	1.0	427	4.0	4.0	529	0.0	1.5
20	3.5	4.0	122	3.5	3.0	224	3.5	3.0	326	0.5	1.0	428	5.0	4.0	530	4.0	3.0
21	5.0	5.0	123	4.0	3.0	225	3.5	3.0	327	3.5	3.0	429	2.0	2.5	531	1.5	0
22	5.0	5.0	124	2.0	2.5	226	2.0	2.0	328	0.0	1.0	430	2.5	2.0	532	5.0	4.5
23	5.0	5.0	125	3.5	3.5	227	2.0	2.0	329	0.5	1.5	431	2.0	1.5	533	3.5	2.5
24	5.0	4.0	126	2.0	2.5	228	0.0	0.5	330	2.0	1.5	432	3.0	2.0	534	4.0	2.5
25	3.5	3.0	127	2.0	2.5	229	0.0	0.5	331	5.0	4.0	433	1.5	1.5	535	2.5	1.0
26	1.5	2.5	128	2.5	3.0	230	2.0	1.5	332	5.0	4.5	434	2.0	1.5	536	3.0	2.0
27	3.0	3.5	129	2.0	2.0	231	3.5	2.0	333	3.5	2.5	435	2.0	2.5	537	5.0	4.5
28	5.0	4.5	130	2.0	2.0	232	3.5	3.5	334	5.0	4.0	436	3.0	3.0	538	2.5	2.5
29	3.0	3.5	131	3.5	3.0	233	3.5	2.5	335	0.0	0	437	2.0	1.5	539	0.0	1.0
30	1.5	3.5	132	3.0	2.0	234	2.0	1.0	336	0.0	1.5	438	5.0	5.0	540	4.0	3.0
31	5.0	5.0	133	3.0	2.0	235	2.5	0.5	337	5.0	4.5	439	2.5	2.0	541	0.0	1.0
32	5.0	4.5	134	0.0	1.0	236	1.0	1.0	338	0.0	0	440	2.0	1.5	542	0.0	0.5
33	2.5	4.0	135	0.0	0.5	237	2.5	1.0	339	0.0	0.5	441	5.0	4.0	543	5.0	4.0
34	2.5	2.5	136	3.5	2.5	238	2.5	1.5	340	5.0	3.5	442	2.5	2.5	544	5.0	2.5

35	0.0	0.5	137	5.0	5.0	239	2.0	0.5	341	3.0	2.5	443	5.0	4.0	545	2.5	1.5
36	0.0	0.5	138	3.5	3.0	240	2.5	1.0	342	5.0	4.5	444	1.0	1.0	546	0.0	1.0
37	2.0	4.0	139	0.0	1.0	241	3.0	2.0	343	5.0	5.0	445	1.0	0.5	547	5.0	4.5
38	4.5	4.0	140	5.0	5.0	242	4.0	3.5	344	3.5	4.0	446	5.0	3.5	548	0.0	0.5
39	0.5	0.5	141	5.0	4.5	243	1.5	1.5	345	3.5	3.0	447	2.5	2.0	549	0.0	1.0
40	3.0	4.5	142	5.0	4.0	244	1.5	2.0	346	3.0	3.0	448	2.5	2.5	550	5.0	4.5
41	5.0	4.5	143	4.0	4.0	245	1.5	2.0	347	0.0	1.5	449	4.5	4.5	551	2.0	1.0
42	5.0	4.0	144	5.0	5.0	246	2.5	2.5	348	0.0	0.5	450	1.0	0.5	552	0.5	0
43	0.5	3.5	145	1.0	1.5	247	3.0	1.5	349	0.0	0.5	451	5.0	3.5	553	0.0	0.5
44	0.5	1.0	146	0.0	1.0	248	1.0	1.5	350	3.0	3.0	452	2.0	3.0	554	2.0	1.5
45	0.5	1.0	147	2.5	3.5	249	3.0	2.5	351	1.5	1.5	453	5.0	4.0	555	2.0	2.0
46	0.5	3.5	148	3.0	3.5	250	3.0	3.0	352	1.5	1.5	454	2.0	1.5	556	4.0	4.5
47	2.5	1.5	149	3.0	3.5	251	2.5	2.0	353	1.5	1.5	455	5.0	4.0	557	4.0	4.0
48	5.0	3.0	150	5.0	4.5	252	2.5	2.0	354	4.0	4.0	456	2.5	2.5	558	5.0	5.0
49	0.5	3.5	151	5.0	4.5	253	2.5	2.0	355	4.0	4.0	457	0.0	1.0	559	3.0	2.5
50	0.5	2.0	152	5.0	4.5	254	4.5	4.0	356	2.0	1.5	458	0.0	1.0	560	3.0	3.0
51	2.5	2.0	153	3.0	3.0	255	4.5	3.0	357	5.0	4.5	459	4.5	3.5	561	5.0	3.5
52	1.0	1.0	154	2.0	1.5	256	2.5	2.5	358	3.0	3.0	460	2.0	2.0	562	5.0	3.0
53	5.0	4.5	155	2.5	2.5	257	2.5	2.5	359	2.5	2.0	461	3.5	3.5	563	5.0	3.5
54	4.5	2.5	156	2.5	3.5	258	5.0	3.5	360	3.5	3.5	462	4.5	3.0	564	5.0	5.0
55	1.0	1.5	157	3.0	3.0	259	3.5	2.5	361	2.5	2.5	463	3.5	3.5	565	2.0	1.5
56	3.5	3.0	158	3.5	3.0	260	3.0	2.5	362	2.5	2.0	464	5.0	5.0	566	0.0	0
57	1.0	2.0	159	5.0	4.5	261	5.0	3.0	363	3.5	1.5	465	2.5	2.5	567	3.0	3.5
58	0.0	1.0	160	5.0	5.0	262	5.0	5.0	364	3.5	4.0	466	1.0	1.5	568	2.5	2.5
59	5.0	5.0	161	5.0	4.5	263	5.0	5.0	365	3.0	2.5	467	1.0	2.0	569	0.0	0
60	0.0	1.0	162	5.0	5.0	264	5.0	5.0	366	2.5	1.5	468	2.5	2.5	570	5.0	4.0
61	5.0	4.0	163	4.5	3.5	265	2.0	1.0	367	5.0	5.0	469	3.0	3.5	571	3.5	3.5
62	2.0	1.5	164	0.0	1.0	266	5.0	5.0	368	5.0	5.0	470	1.0	1.0	572	5.0	3.5
63	5.0	4.0	165	1.5	2.0	267	2.5	0.5	369	3.0	2.5	471	3.5	2.5	573	0.0	0.5
64	1.5	2.5	166	3.5	3.5	268	5.0	3.0	370	2.0	1.5	472	3.0	3.0	574	0.0	0.5
65	1.5	1.5	167	5.0	5.0	269	5.0	4.5	371	2.0	2.0	473	4.0	4.5	575	5.0	5.0
66	2.0	2.5	168	5.0	5.0	270	0.0	1.5	372	5.0	4.0	474	5.0	5.0	576	5.0	4.5
67	2.5	2.5	169	3.5	2.5	271	2.0	1.5	373	2.5	3.0	475	3.5	4.0	577	5.0	3.5
68	3.5	3.5	170	2.5	2.0	272	5.0	4.0	374	2.5	2.0	476	1.5	2.0	578	0.0	0.5
69	2.5	2.5	171	5.0	3.5	273	5.0	4.5	375	2.5	2.5	477	1.5	1.5	579	5.0	4.0
70	2.0	2.5	172	5.0	2.5	274	1.0	1.0	376	5.0	3.5	478	4.0	4.5	580	4.0	4.0
71	1.5	1.0	173	4.0	3.5	275	1.0	1.0	377	2.5	1.5	479	3.0	3.5	581	5.0	2.0
72	3.5	3.5	174	3.5	1.5	276	0.0	0	378	1.0	2.0	480	2.0	3.0	582	3.5	3.5
73	5.0	4.5	175	0.0	1.0	277	2.5	2.5	379	5.0	4.5	481	3.0	3.0	583	5.0	4.0
74	5.0	4.5	176	1.0	1.0	278	0.0	2.5	380	2.0	2.0	482	3.0	3.0	584	0.0	0.5
75	5.0	5.0	177	5.0	4.5	279	2.0	2.5	381	4.5	4.0	483	4.5	3.5	585	0.0	0.5
76	5.0	5.0	178	2.5	1.0	280	1.0	0.5	382	4.5	3.5	484	3.5	2.5	586	5.0	4.5
77	3.5	2.5	179	3.5	2.5	281	5.0	4.5	383	3.0	3.0	485	1.0	2.0	587	5.0	3.5
78	2.0	2.0	180	4.0	3.0	282	5.0	4.5	384	2.5	1.5	486	4.5	3.0	588	0.0	0.5
79	2.0	2.0	181	5.0	4.0	283	5.0	4.5	385	3.5	3.5	487	1.0	1.5	589	0.0	0.5

80	4.0	4.5	182	4.5	4.5	284	5.0	3.0	386	1.0	1.5	488	2.5	2.0	590	3.5	2.0
81	4.5	4.5	183	1.5	2.5	285	5.0	4.0	387	3.5	3.5	489	0.0	1.0	591	1.5	1.0
82	4.0	3.5	184	3.5	3.5	286	5.0	4.0	388	3.0	2.5	490	5.0	3.5	592	3.0	3.0
83	4.5	4.5	185	2.5	2.5	287	0.5	0	389	3.0	3.0	491	2.5	1.0	593	2.5	1.5
84	2.0	2.5	186	2.5	2.5	288	0.5	0	390	3.0	3.0	492	2.0	1.5	594	5.0	4.5
85	4.5	5.0	187	0.0	1.0	289	0.5	0	391	2.0	1.5	493	2.0	2.0	595	4.0	4.5
86	2.0	3.0	188	2.5	2.0	290	0.5	0	392	5.0	3.5	494	0.0	1.0	596	3.0	3.0
87	3.5	3.5	189	0.5	1.5	291	2.5	2.0	393	4.5	3.0	495	0.0	0.5	597	4.0	2.0
88	2.0	2.5	190	2.5	2.5	292	2.5	2.0	394	2.0	1.5	496	4.5	2.5	598	5.0	3.0
89	1.5	2.0	191	4.0	3.5	293	2.5	2.5	395	2.0	1.5	497	5.0	3.5	599	0.0	1.0
90	4.5	4.5	192	3.5	3.5	294	4.5	4.0	396	5.0	5.0	498	2.0	2.0	600	2.5	1.0
91	5.0	5.0	193	2.0	2.5	295	3.0	2.0	397	3.5	3.5	499	4.5	1.5	601	0.0	0.5
92	5.0	5.0	194	3.0	3.0	296	4.5	4.0	398	2.0	1.5	500	0.0	2.0	602	2.5	1.0
93	5.0	4.5	195	3.5	3.5	297	2.5	2.0	399	5.0	3.5	501	4.0	3.5	603	5.0	3.5
94	5.0	4.5	196	1.5	2.5	298	2.5	3.0	400	3.0	3.0	502	4.0	3.0	604	4.0	1.5
95	0.0	1.5	197	3.5	3.5	299	5.0	3.0	401	2.0	1.5	503	4.5	4.0	605	4.5	2.5
96	1.0	1.0	198	1.0	2.0	300	0.5	0	402	3.0	3.0	504	1.0	1.5	606	4.5	2.5
97	0.0	1.5	199	1.0	2.0	301	3.0	3.5	403	3.0	1.5	505	1.0	1.5	607	2.0	1.5
98	2.5	3.0	200	1.5	2.5	302	5.0	4.5	404	3.0	2.0	506	2.0	2.0	608	5.0	4.0
99	2.5	3.5	201	4.5	3.5	303	5.0	5.0	405	2.0	3.0	507	5.0	4.0	609	4.5	2.5
100	0.0	1.0	202	2.5	2.0	304	0.0	1.0	406	0.0	1.0	508	4.5	3.5	610	4.0	1.5
101	3.5	3.5	203	2.5	2.0	305	4.0	2.5	407	1.0	1.0	509	4.0	3.0			
102	4.5	3.5	204	4.5	3.0	306	2.0	2.0	408	3.0	1.5	510	1.5	1.0			