Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE SAAD DAHLAB DE BLIDA 1

Faculté des Sciences

Département d'Informatique

MÉMOIRE DE MASTER INFORMATIQUE IL

Mesures de similarité syntaxique pour un système d'évaluation automatique des réponses courtes : Application à la langue arabe

Réalisé par :

ABDALLAH Amina
GAROUDJA Khadidja

Proposé et encadré par :

Mme OUAHRANI Leila

Composition de jury :

M.	BALA MAHFOUD	Président
M.	KAMECHE ABDALLAH HICHAM	Examinateur

Soutenu le :

26/09/2018

الملخص

أصبح التقييم التلقائي أحد أهم الأسلحة في مجال التعليم من حيث كفاءته في توفير الوقت وإمكانية تنفيذ عدة بسهولة ، اختبارات في فترة صغيرة، وكذلك التعليم عن بعد على سبيل المثال. يتم تقييم اسئلة صحيح / خطأ ،أسئلة الاختيار ومع ذلك ، نحن مهتمون بالأسئلة ذات الإجابات القصيرة التي تقيس بشكل أفضل عمل المتعلمين من حيث اكتساب المعرفة. يستند مبدأ التقييم التلقائي للإجابات القصيرة على حساب التشابه بين إجابة الطالب و الاجابة النموذجية للمعلم. في هذا العمل ، نحن مهتمون بالتشبيه النحوي. تم تطوير مجموعة من الانهجة الموجودة في الأدبيات والأبحاث المقترحة حول أداة تسمح بالتقييم التلقائي على 3 مجموعات بيانات (قواعد بيانات) تتعامل مع اللغة العربية. وقد أظهر التهجين مع الانهجة ذات التشبيه الدلالي أهمية التشبيه النحوي في عملية التقييم التلقائي

الكلمات المفتاحية : التشبيه النحوي، التقييم التلقائي للإجابات القصيرة ، جذر الكلمة ، ASAGS، قواعد بيانات

Résumé

L'évaluation automatique est devenue une des armes les plus importantes dans le champ de l'éducation, pour son efficacité en terme de gain de temps, faisabilité de lancer plusieurs tests dans une petite période, ainsi que l'apprentissage à distance par exemple. L'évaluation des questions de type vrai/faux, question à choix multiples, sont facilement évalués, néanmoins, nous nous intéressons aux questions à réponses courtes qui évaluent de mieux le travail d'apprenant en terme d'acquisition de connaissances. Le principe d'évaluation automatique des réponses courtes se base sur le calcul de la similarité entre la réponse de l'étudiant et la réponse modèle de référence de l'enseignant. Nous nous intéressons dans ce travail aux mesures de similarité syntaxiques. Une panoplie de mesures existantes dans la littérature et celles proposées sont développées autour d'un outil qui permet d'évaluer les différentes approches sur 3 jeux de données (Datasets) traitant de la langue arabe. L'hybridation avec des approches sémantiques a permis de montrer l'importance des approches syntaxiques dans le processus d'évaluation automatique.

Mots clés: Similarité syntaxique, Evaluation automatique des réponses courtes, lemmatisation, Stem, ASAGS, NLP, Datasets

Abstract

Automatic grading has become one of the most important arms in the field of education, for its efficiency of saving time, possibility of launching several tests in a small period, as well as distance learning for example. The evaluation of true / false questions, multiple-choice questions, are easily evaluated, however, we are interested in short-answer questions that evaluate the work of students in a better way to insure the transformation of knowledge. The main idea of automatic evaluation of short answers is based on measuring the similarity between the student's answer and the teacher's model answer. In this work, we are interested in string similarity measures. A range of existing measures in the literature and those proposed are developed around a tool that allows to evaluate different approaches on 3 datasets dealing with the Arabic language. Hybridization with semantic approaches has shown the importance of string approaches in the automatic evaluation process.

Keywords: String similarity, Automatic short answer grading, stemming, Stem, ASAGS, NLP, Datasets

Remerciements

Nous remercions tout d'abord ALLAH le tout puissant d'avoir nous donner le courage, la volonté et la patience de mener à terme le présent travail.

Nous exprimons toute notre reconnaissance et gratitude à notre promotrice Madame Leila Ouahrani, pour son effort à nous garantir la continuité et l'aboutissement de ce travail, pour sa patience ainsi que sa disponibilité à nous orienter et à nous guider depuis le début du travail jusqu'à la fin , pour ses remarques constructives, et son professionnalisme qui nous a mené à découvrir ce nouveau domaine, et arriver à présenter convenablement ce travail.

Nous remercions chaleureusement, nos chers parents qui nous ont soutenus depuis le début de notre parcours, leur soutien moral et leurs prières nous ont toujours été une épaule sur laquelle on peut se reposer et compter.

Nos remerciements s'adressent aussi à Monsieur Wael Hassan Gomaa de l'université d'Egypte de nous avoir donné l'accès à son data set et de pourvoir l'exploiter et l'utiliser.

Nous remercions du fond de cœur, tous les membres de la famille Abdallah, Garoudja, et Serradj pour leur soutien éternel et leur présence depuis le début, nos amis et surtout nos collègues, Asma et Yasmine, Adel et Hamza, avec eux on a pu expérimenter cette nouvelle agréable expérience de travail en groupe. Merci à vous.

Enfin, nous tenons à remercier tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Merci !

Liste des figures

Liste des tables

Tables des matières

I. Introduction générale

Ce chapitre représente le contexte de notre travail, en précisant la problématique du travail en faisant un tour sur la difficulté de développement de ce domaine dans la langue arabe ainsi que la difficulté de l'évaluation des réponses courtes, on passera par la suite aux objectifs de notre travail, ses limites, et pour finir on va détailler les enjeux de la langue arabe dans ce travail.

i. Introduction

Pendant de nombreuses années, le processus d'apprentissage a été perçu comme un cercle fermé entre les enseignants et les étudiants en termes de quiz et d'examens. L'évaluation de connaissances acquises par l'apprenant est l'un des aspects les plus importants du processus d'apprentissage.

L'évaluation automatique est utilisée dans l'enseignement pour la réalisation et la consolidation des avantages d'un système présentant les caractéristiques suivantes :

Premièrement, réduire la charge de travail des enseignants en automatisant une partie de la tâche d'évaluation des apprenants, Deuxièmement, fournir aux étudiants des informations détaillées sur leur période d'apprentissage de manière plus efficace que l'évaluation traditionnelle, et Enfin, intégrer la culture d'évaluation au travail quotidien des apprenants dans un environnement d'e-Learning.

Plusieurs systèmes d'évaluation automatique sont en pratique dans le domaine de l'enseignement particulièrement l'enseignement en ligne, les types les plus simples de systèmes de notation automatique en termes de mise en œuvre et de conception sont ceux conçus pour des 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), nature subjective de l'évaluation (multiples réponses possibles), manque de cohérence dans la notation humaine, … etc.

ii. Problématique

Le concept principal de l'évaluation automatique des réponses courtes consiste à comparer la réponse de l'apprenant à la réponse de référence de l'enseignant (réponse modèle) et à mesurer la similarité entre les deux réponses.

Les questions à réponses courtes sont le type de question le plus difficile et le plus long à évaluer. Les études récentes, dans le domaine de l'évaluation automatique de réponses courtes basées sur les types d'approches utilisées ainsi que sur l'étendue de la supervision humaine nécessaire, s'accordent à conclure que l'évaluation automatique de réponses courtes ne donne pas encore le meilleur résultat.

D'un autre côté, l'arabe est une langue répandue parlée par environ 300 millions de personnes à travers le monde alors que la plupart des recherches dans l'évaluation automatique des réponses courtes traitent de *l'anglais*. Du point de vue du langage naturel, la langue arabe se caractérise par une ambiguïté élevée et une morphologie riche et complexe [1].

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* dans la langue arabe : corpus arabes, 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.

Il existe plusieurs mesures de similarité pour mesurer la similarité entre la réponse de référence de l'enseignant et la réponse de l'étudiant(ou l'apprenant). Ces mesures sont classées dans deux approches principales ; les mesures syntaxiques (comparaison des chaines de caractères) et les mesures sémantiques (comparaison de contenu sémantique).

iii. Les objectifs

- Etudier les diverses techniques d'évaluation automatiques des réponses courtes.
- Faire une synthèse expérimentale de différentes approches de mesures de similarité syntaxiques appliquées à des ensembles de données (Data Sets) exprimées dans la langue arabe. Parmi ces approches, certaines approches existent dans la littérature et d'autre sont proposées dans le cadre de ce travail.
- Retenir les meilleures approches en vue d'une future hybridation avec des mesures sémantiques dans le système d'évaluation automatique.
- Développer un outil d'analyse d'approches et d'évaluation automatique des questions à réponses courtes pour la langue arabe.

iv. Importance du travail

L'influence de similarité syntaxique est importante dans le domaine d'évaluation automatique qui est la cible de notre recherche. Les objectifs de l'utilisation de l'évaluation automatique dans l'enseignement comprennent la réalisation et la consolidation des avantages d'un système présentant les caractéristiques suivantes :

- Réduire la charge de travail des enseignants en automatisant une partie de la tâche d'évaluation des apprenants,
- Fournir aux étudiants des informations détaillées sur leur période d'apprentissage de manière plus efficace que l'évaluation traditionnelle,
- Libérer l'enseignant de la correction manuelle subjective qui peut être influencée par son humeur (stress, joie…etc.)
- Intégrer la culture d'évaluation au travail quotidien des apprenants dans un environnement d'e-Learning.

Considérant la langue arabe, la plupart des recherches dans le domaine d'évaluation automatique sont en progrès par rapport aux différentes langues latines et en particulier l'anglais. L'arabe n'en fait pas partie le très peu de travaux la concernent ce qui valorise

encore plus cette recherche.

D'une autre part, la similarité syntaxique n'est pas négligeable dans le domaine de la similarité:

- La syntaxe est, à l'origine, la branche de la linguistique qui étudie la façon dont les mots se combinent pour former des phrases ou des énoncés dans une langue[1], ce qui donne plus d'importance aux mesures syntaxiques dans la mesure de similarité entre phrases.
- La similarité sémantique est souvent combinée avec la similarité syntaxique pour pouvoir donner un résultat meilleur.

v. Limites de notre travail

- Dans ce travail nous avons utilisé une réponse de référence (réponse modèle de l'enseignant) pour mesurer la similarité entre la réponse d'étudiant et la réponse modèle. Par conséquent, il ne convient pas pour l'évaluation de l'essai en général où l'on ne considère pas une réponse de référence.
- Le travail adopte un système de notation à réponse courte en langue arabe qui est général, et non spécifique à un certain domaine.
- Ce travail traite de la langue arabe. Ainsi, toute ressource, mot ou paraphrase d'une langue autre que l'arabe est ignoré,
- Dans la suite de notre travail nous considérons qu'une réponse est « réponse courte » si elle respecte les propriétés suivantes:
  - La réponse doit être rédigée dans un langage naturel.
  - Elle doit être issue (formulée) de connaissances externes à la question elle-même.
  - Sa longueur est de l'ordre d'une centaine de mots,
  - La réponse valorise le contenu et non pas le style d'écriture.

[1] https://fr.wikipedia.org/wiki/Syntaxe

vi. Structure du mémoire

La structure de notre travail est composée de 4 chapitres principaux :

- Chapitre 1 : Introduction générale qui englobe le contexte et problématique du travail.
- Chapitre 2 : Etat de l'art qui traite les travaux connexes à notre travail, ainsi que les approches de similarités et systèmes ASAGS, pour mieux encadrer le champ de notre travail.
- Chapitre3 : Système d'évaluation automatique des réponses courtes qui explique les démarches de la conception et l'évaluation de notre système.
- Chapitre 4 : Résultat expérimentaux et évaluation, pour évaluer notre travail et exposer les résultats obtenus.
- On termine par une conclusion et des perspectives.

II. L'état de l'art

Dans ce chapitre, nous allons encadrer notre travail, et faire le tour sur ce qui existe déjà dans la littérature, en commençant par présenter les diverses techniques d'évaluation automatique des réponses courtes, ensuite pour généraliser et comprendre le contexte de travail nous parlerons des systèmes d'évaluation automatiques des réponses courtes ASAG (Automatic Short Answer Grading), ensuite nous expliquerons les approches de similarité qui existent, et enfin nous allons exposer les travaux connexes à notre travail pour pouvoir faire une comparaison à la fin du travail entre ce qui est fait et ce que nous allons faire.

i. Les approches d'évaluation automatique des réponses courtes

Les recherches concernant l'évaluation automatique ont vu le jour depuis 1965, Mitchell et al.[2] ont classés les techniques d'évaluation automatique des réponses courtes en trois types principaux: statistique, extraction de l'information et traitement complet en langage naturel comme c'est mentionné dans la figure 1.

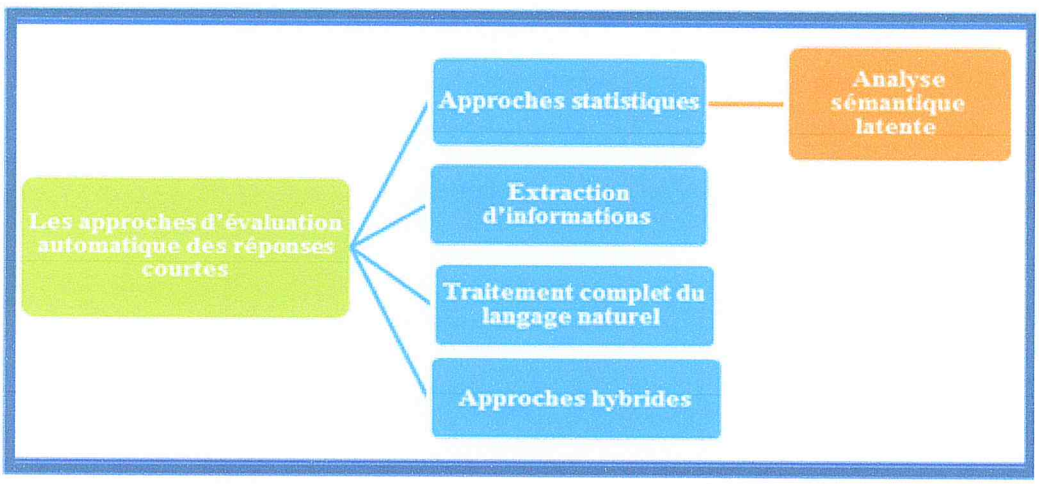

Figure 1 : classification des approches d'évaluation automatique

1. Approches statistiques

En général, tous les systèmes qui reposent sur une analyse statistique d'une ou plusieurs caractéristiques des textes devraient être considérés dans cette catégorie. Ils ont généralement besoin d'une phase d'entraînement initiale pour calculer les paramètres du système [3]. Ils n'utilisent pas de techniques NLP complexes et dans la plupart des cas, les textes sont uniquement traités avec un séparateur de phrases, Ils sont basés sur des corpus et elles ne

nécessitent pas la compréhension du vocabulaire ou de la grammaire de la langue d'un texte [4].

Parmi de telles mesures de similarité sémantique, nous présentons l'analyse sémantique latente (LSA) la plus utilisée dans cette catégorie :

- **Analyse sémantique latente (LSA)**

C'est une technique statistique complexe qui était initialement développé pour l'indexation des documents et la recherche d'information, elle peut néanmoins être appliquée à l'évaluation automatique [3]. Dans ce domaine, cette technique permet d'extraire la similarité conceptuelle entre la réponse de l'étudiant et la réponse modèle de l'enseignant. Selon Dessus et al [5] Cette approche est assez robuste et prouve son nom en trouvant les relations cachées entre des mots qui pourraient être dans des documents différents ou entre des documents qui ne partagent pas forcément les mêmes mots.

Comme exemple des approches de cette catégorie on site encore : Disco1 et Disco 2.

2. Extraction d'informations

L'extraction d'information consiste à acquérir des informations structurées à partir d'un texte libre comme identifier les entités nommées dans le texte et remplir un Template [6]. Elle peut être considérée comme une technique de NLP (Natural Language Processing) peu profonde, car elle ne nécessite généralement pas une analyse approfondie des textes. Elle peut être utilisée pour extraire les dépendances entre les concepts. Premièrement, le texte est divisé en concepts et leurs relations, ensuite, les dépendances trouvées sont comparées aux experts humains pour donner le score de l'élève.

Par exemple : les systèmes ATM (Automatic text Marking) sont basés sur cette approche.

3. Traitement complet du langage naturel

NLP (Natural Language Processing) est l'application de méthodes de calcul pour traiter le langage naturel.

Burstein et al.[7] ont cité des outils tels que les parseurs syntaxiques pour trouver la structure linguistique d'un texte [8] et des parseurs rhétoriques pour trouver la structure discursive d'un texte [9]. En outre, Williams et Dreher [10] ont utilisé le thésaurus électronique pour extraire des informations lexicales et un algorithme de segmentation spécialement conçu pour extraire les syntagmes nominaux et les clauses verbales.

Par exemple : C-rater et PS-ME [11] sont également soutenus par ces techniques. Leur combinaison améliore l'utilisation des statistiques en impliquant une analyse syntaxique en profondeur et une analyse sémantique afin de recueillir plus d'informations pour évaluer efficacement la réponse de l'étudiant. D'un autre côté, il est difficile à réaliser et plus difficile à transférer à travers les langues.

4. Approches hybrides

Il est également possible de tirer parti des meilleures caractéristiques de plusieurs techniques afin d'améliorer un système.

Par exemple : E-rater utilise VSM (Vector space model) pour capturer l'utilisation du vocabulaire et effectuer l'analyse topique, et le reste des phases est basé sur la NLP (Natural Language Processing); L'auto-marquage repose sur la NLP (Natural Language Processing) et l'appariement de motifs, et CarmelTC sur les techniques d'apprentissage automatique et une classification de réseau neuronal bayésien [3].

ii. Les systèmes d'évaluation automatiques (ASAG)

1. Vue historique

La figure 2 représente les différents systèmes d'évaluation automatique :

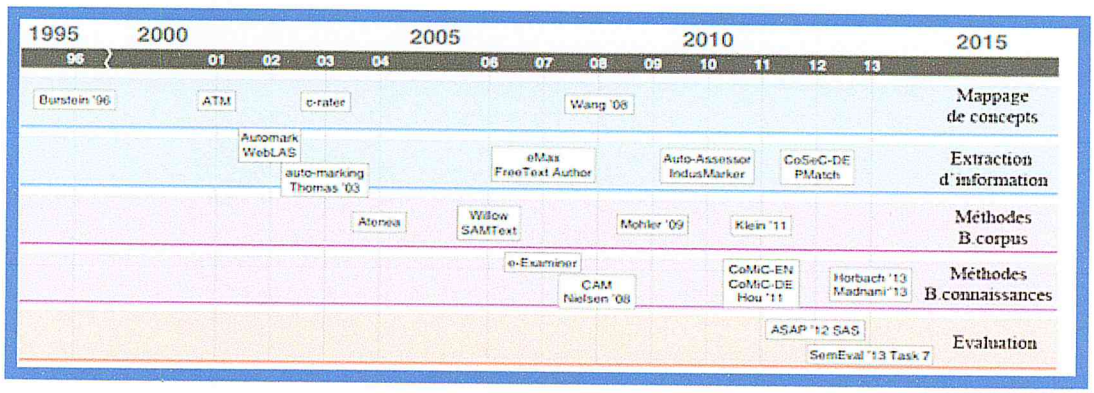

Figure 2 : Les systèmes d'évaluation automatiques (ASAG)

La recherche dans le domaine d'évaluation automatique 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 [12].

2. Fonctionnement des systèmes ASAG

Quand on considère ASAG, il faut non seulement considérer les algorithmes et la technologie, mais aussi les ensembles de données et les techniques d'évaluation qui sont utilisés pour mesurer l'efficacité. Tous ces composants peuvent être considérés comme un "pipeline" où chaque artefact ou processus alimente le suivant. La notion de pipeline est bien soutenue par plusieurs domaines de recherche sur le traitement du langage naturel, notamment l'extraction de relations et le remplissage de modèles [13] et l'extraction efficace d'informations [14].

Pour les systèmes ASAGS, la forme générale d'un pipeline de développement d'un système ASAG est constituée de 5 processus et 6 artéfacts comme le montre la figure 3 :

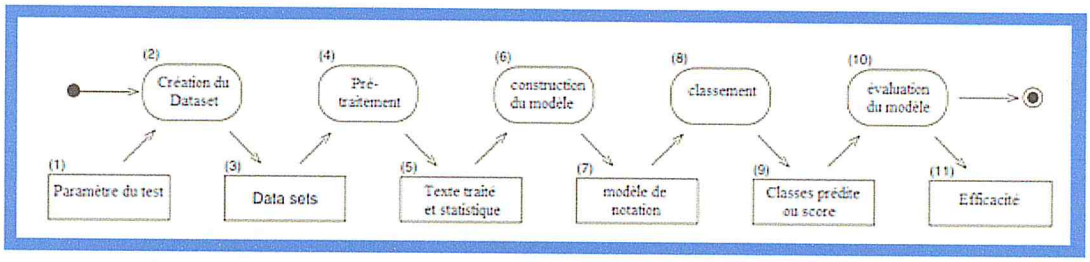

Figure 3 : Pipeline de développement des systèmes ASAGs

- **Création du data set :** Tout d'abord, un test ou examen est passé dans un domaine donnée, Ensuite, un ou plusieurs ensembles de données sont créés en rassemblant les questions, réponses de l'enseignant et réponses de l'élève. Les ensembles de données sont stockés sur un disque dans un fichier plat, XML ou un format similaire.
- **Prétraitement du data set :** un prétraitement de tâches NLP est appliqué sur le data set, pour générer un data set prétraité et prêt à être utilisé.
- **Construction de modèle :** à cette étape là, un modèle est généré pour les réponses modèles et les réponses des étudiants à fin qu'ils seront utilisables pour le calcul de similarité avec les approches d'un système donné, comme le modèle d'espace vectoriel dans la similarité syntaxique.
- **Classement :** à cette étape un ensemble des approches de similarité sont appliquées pour donner la similarité entre chaque couple de data set (entre la réponse modèle et la réponse de l'étudiant) qui est donnée entre 0 et 1, ensuite cette valeur sera passé au score, pour avoir la note finale que le professeur veut accorder à une question donné du data set.
- **Evaluation du modèle :** à cette dernière étape, on va estimer et valoriser notre

système, pour évaluer son efficacité, il existe plusieurs méthodes comme le Coefficient de corrélation de Pearson et l'erreur quadratique RMSE.

3. Quelques exemples des systèmes ASAG

- **Burstein** [15] considèrent des questions de type hypothèse où plusieurs explications doivent être données pour une hypothèse donnée, chacune pouvant correspondre ou non à l'une des réponses de l'enseignant. Chaque réponse peut être considérée comme un concept distinct. La technique appliquée est la représentation de la structure conceptuelle lexicale [16] selon laquelle un lexique conceptuel et une grammaire conceptuelle doivent être développés à partir d'un ensemble d'apprentissage avant de classer les hypothèses dans les réponses des élèves.
- **ATM** (marqueur de texte automatique) [17] répartit les réponses des enseignants et des élèves en listes de concepts minimaux ne comprenant que quelques mots chacun, et compte le nombre de concepts communs pour fournir une évaluation mais, Chaque concept est essentiellement la plus petite unité possible dans une réponse à laquelle on peut attribuer un poids pour les fins du classement. Les poids sont additionnés pour produire le score global.
- **eMax** [18] demande à l'enseignant de baliser les éléments sémantiques requise des réponses de l'enseignant, d'accepter ou de rejeter les synonymes de ces éléments et d'attribuer des poids à chaque élément pour calculer la note finale [19]. L'approche de la notation est une approche combinatoire, où toutes les formulations possibles sont prises en compte lors de l'appariement de modèles. Les scores attribués reçoivent également une cote de confiance, de sorte que les cas difficiles peuvent être transmis pour un examen manuel.

iii. Les approches de similarité

Évaluer la similarité entre des documents textuels est une des problématiques importantes de plusieurs disciplines comme l'analyse de données textuelles, la recherche d'information ou l'extraction de connaissances à partir de données textuelles, ainsi que l'évaluation automatique.

Les techniques mises en œuvre pour calculer les similarités varient bien évidemment selon les contextes, mais on peut cependant l'intégrer dans trois catégories comme c'est montré dans la figure 4 :

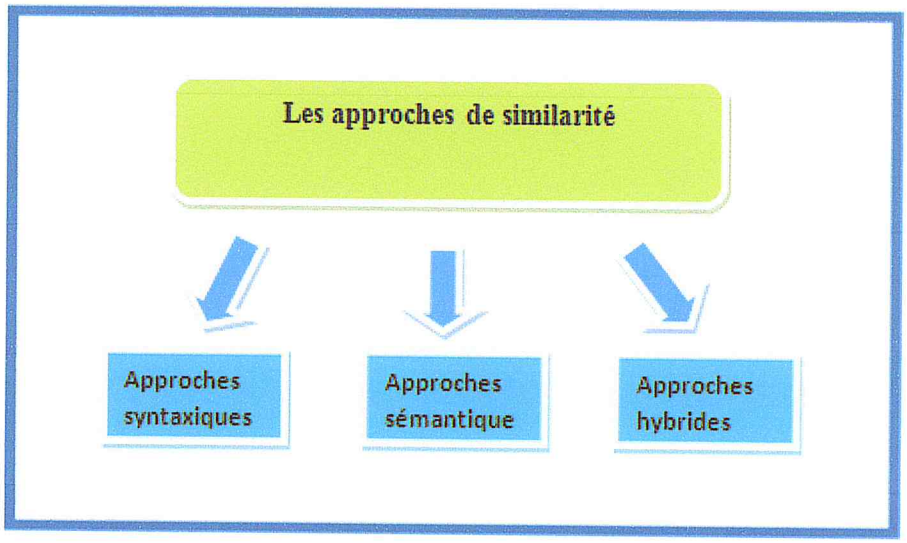

Figure 4: Les approches de similarité

1. La similarité syntaxique

Une mesure de similarité syntaxique permet de comparer des documents textuels en se basant sur les chaines de caractères qui les composent. Par exemple, les chaînes de caractères"جلس" et "جالس" peuvent être considérées comme très proches, alors que "جلس" et "قعد" pourront être considérées comme très différents.

Parmi de telles mesures de similarité syntaxique, citons par exemple, la distance de Levenshtein (ou distance d'édition), le coefficient de Dice, l'indice de Jaccard, la distance euclidienne, le cosinus…, [7]qui seront bien détaillées et expliquées dans le prochain chapitre

Comme la similarité syntaxique est une tâche NLP, les réponses devront être représentées d'une manière mathématique compréhensible par la machine, comme le modèle d'espace vectoriel.

A. Modèle d'espace vectoriel : La figure 5 représente le modèle d'espace vectoriel pour la similarité syntaxique :

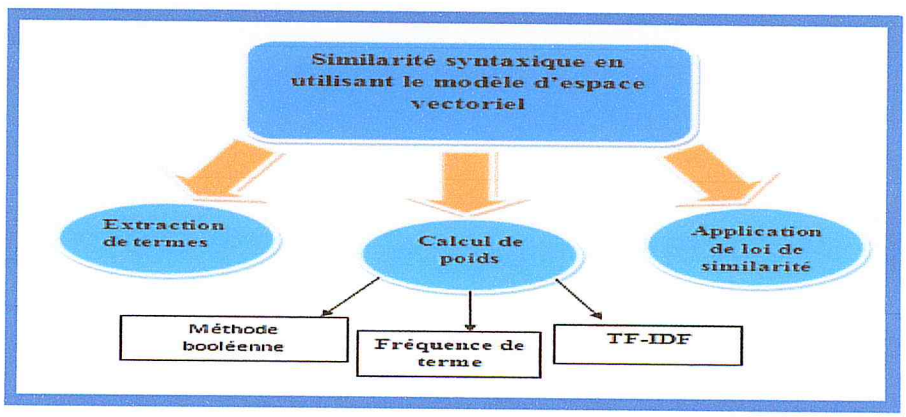

Figure 5 : Similarité syntaxique en utilisant le modèle vectoriel

Afin de réduire la complexité des documents et de faciliter leur manipulation, il faut transformer chaque document, i.e. sa version textuelle intégrale, en un vecteur qui décrit le contenu du document. La représentation d'un ensemble de documents sous forme de vecteurs dans un espace vectoriel commun est connue sous le nom de modèle d'espace vectoriel (Vector Space Model).

La représentation d'un document sous forme vectorielle se déroule en 2 étapes :

1. **Extraction des termes pertinents**

Il s'agit de prétraiter le texte des documents textuels en supprimant les mots-vides, la ponctuation et les éventuels 'retours-chariots', de lemmatiser le texte et de le segmenter.

2. **Calcul des poids**

Le poids de chaque terme dans un document peut être obtenu de différentes manières : booléenne, fréquence des termes, Tf-Idf (Term frequency - Inverse Document Frequency).

❖ **Méthode booléenne**

De manière booléenne, si un terme existe dans un document alors la valeur qui lui correspond vaut 1, sinon 0. L'approche booléenne est utilisée lorsque chaque terme est d'égale importance et s'emploie uniquement lorsque les documents sont de petites tailles.

❖ **Fréquence des termes**

Pour la fréquence des termes, le poids d'un terme est obtenu en comptant les occurrences du terme dans le document : Tfij représente donc la fréquence du terme i dans le document j [7].

❖ **Tf-Idf**

Elle vise à donner un poids plus important aux termes les moins fréquents, considérés comme les plus discriminants.

Il s'agit de calculer le logarithme de l'inverse de la proportion de documents qui contiennent le terme : **Idfi = log (|D| / | {dj : ti appartient à dj}|)** où :

|D| est le nombre total de documents

| {Dj : ti appartient à dj}| est le nombre de documents où le terme ti apparait.

Finalement, le poids s'obtient en multipliant les deux mesures :

Tfidf i, j = tf i,j * idfi.

B. Exemple de modèle d'espace vectoriel

D1=البيئة التى يشترك فيها الإنسان مع الكائنات الحية و الغير حية

D2=البيئة التى يشترك فيها الإنسان مع باقى البشر

Après avoir supprimé les mots vides, et avec lemmatisation avec le lemme Khoja on obtient les termes et la représentation vectorielle en utilisant la méthode booléenne comme c'est montré dans le tableau 1 :

D1=بيئة التى شرك أنس كون حيا غور حيا

D2=بيئة التى شرك أنس بقي بشر

Terme	بيئة	التى	شرك	أنس	كون	حيا	غور	حيا	بقي	بشر
D1	1	1	1	1	1	1	1	1	0	0
D2	1	1	1	1	0	0	0	0	1	1

Tableau 1 : représentation booléenne des phrases

$$SIMcosine(D1, D2) = \frac{\overrightarrow{D1}.\overrightarrow{D2}}{\left\|\overrightarrow{D1}\overrightarrow{D2}\right\|} = \frac{4}{\sqrt{8}\sqrt{6}} = 0.57$$

2. La similarité sémantique

Une mesure de 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 / contenu sémantique [4].

Nous distinguons 3 sortes de similarités sémantiques : les approches vectorielles, les approches topologiques et les approches statistiques (voir la figure 6).

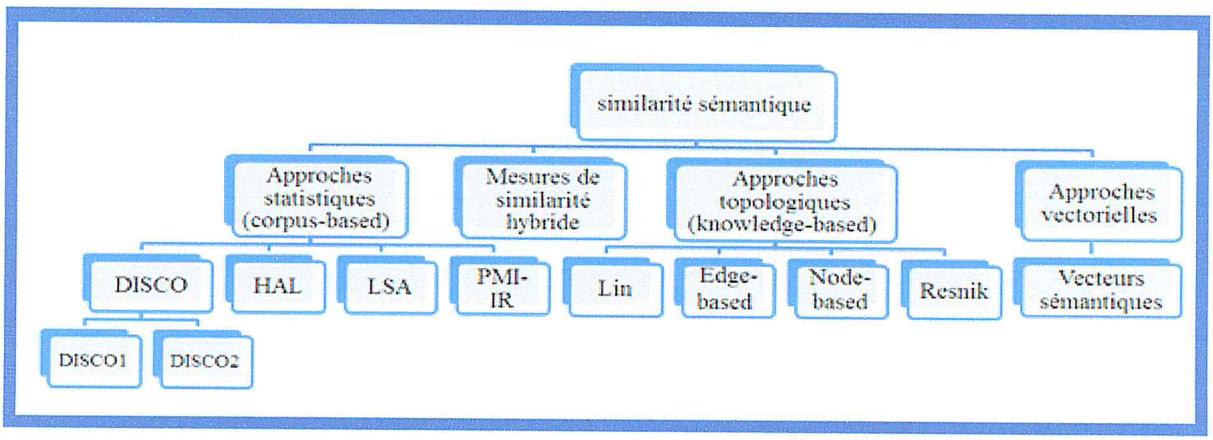

Figure 6 : La similarité sémantique

A. Approches vectorielles

➢ **Vecteurs sémantiques**

L'idée consiste à déterminer la sémantique d'un mot en consultant les autres termes utilisés à ses côtés dans des phrases. Une manière simple de le faire est d'utiliser des vecteurs pour représenter le sens des mots, et d'utiliser ensuite des mesures de similarité vectorielles (comme pour la similarité syntaxique). 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 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.

B. Approches topologiques (ou knowledge-based)

Les approches de similarité de mots basées sur la connaissance s'appuient sur un réseau sémantique de mots, tel que WordNet [20]. Etant donnés deux mots, leur similarité peut être estimée à partir de leurs positions relatives dans la hiérarchie de la base de connaissances. En effet, la structure de la base est un arbre où chaque nœud est un concept (par exemple, un chat), ses enfants sont les hyponymes du concept (i.e., 'X' est un hyponyme de 'Y' si 'X est un Y' est vrai), et ses parents sont ses hyperonymes (i.e., 'X' est un hyperonyme de 'Y' si 'Y est un X' est vrai). Les concepts peuvent être des noms, des verbes ou des adjectifs.

Les mots ont des "synsets", qui sont des ensembles de concepts pour lesquels le mot peut correspondre (i.e., les concepts pour lesquels le mot peut être synonyme). Enfin, il faut noter que les concepts sont de plus en plus abstraits et généraux lorsqu'on va vers la racine et qu'ils sont plus spécifiques lorsqu'on va vers les feuilles [4].

Remarque : Wordnet est une base de connaissances ou taxonomie dont les concepts sont en anglais. Cependant, une base similaire a été créée pour la langue française : WOLF (WordNet Libre du Français) [21].

Exemples :

Edge-based

L'approche basée sur les arcs est une manière naturelle et directe d'évaluer la similarité sémantique dans une taxonomie. Il s'agit d'estimer la distance (e.g. longueur des arcs) entre les nœuds correspondants aux concepts / classes à comparer. Compte tenu de l'espace multidimensionnel des concepts, la distance conceptuelle peut facilement être mesurée par la distance géométrique entre les nœuds représentants les concepts. Évidemment, plus le chemin d'un nœud à l'autre est court, plus ils sont similaires.

Il existe aussi les mesures : **Leacock et Chodorow** [22], **Wu et Palmer** [23], **Node-based (ou information content-based) Resnik** [24], **Lin** [25], **Jiang and Conrath** [26].

C. Approches corpus-based

Les mesures basées sur des corpus différents des mesures présentées précédemment car elles ne nécessitent pas la compréhension du vocabulaire ou de la grammaire de la langue d'un texte. La similarité basée sur le corpus détecte la similarité entre les mots en fonction de l'information obtenue à partir d'un corpus important. Il existe de nombreuses techniques de similarité basées sur les corpus, telles que l'analyse sémantique latente (LSA), l'analyse sémantique explicite (ESA) [27], la recherche d'informations point par point (PMI-IR) [28], et enfin, Extraction de mots similaires sur le plan de la distribution à l'aide de Cooccurrences (Disco). DISCO prend en charge neuf langues, à savoir l'arabe, le tchèque, le néerlandais, l'anglais, le français, l'allemand, l'italien, le russe et l'espagnol. L'outil Disco calcule la similarité distributionnelle entre les mots tout en considérant que les mots qui sont similaires dans le sens se produisent dans un contexte similaire. La similarité est basée sur l'analyse statistique de très grandes collections de textes. Pour déterminer la similarité entre deux mots, la mesure Lin [29] est appliquée aux mots qui ont été récupérés à partir des vecteurs des données indexées.

Parmi les approches Corpus based avec lesquels on va combiner nos approches syntaxiques de notre travail, c'est les approches qui utilisent les WORD EMBEDDINGS(WE) :

Les word embedding peuvent constituer une alternative efficace aux bases de données linguistiques [30]. Aussi appelé représentation distribué 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.

- **Modèles de word embedding**

Les représentations de mots en tant que vecteurs dans un espace multidimensionnel permettent de capturer les propriétés sémantiques et syntaxiques du langage [30]. Ces représentations peuvent servir d'unité de construction fondamentale à de nombreuses applications du traitement automatique du langage naturel. Dans la littérature, plusieurs techniques sont proposées pour construire des représentations spatiales vectorisées.

Mnih et Hinton [31] ont proposé une autre forme pour représenter les mots dans l'espace vectoriel, appelée modèle hiérarchique log-bilinéaire (HLBL). Comme pratiquement tous les modèles de langage neuronal, le modèle HLBL représente chaque mot avec un vecteur d'entité à valeur réelle. Pour les mots de n-gramme, HLBL concatène les n-1 premiers mots d'incorporation (w1..wn-1) et apprend un modèle linéaire neuronal pour prédire le dernier mot wn.

Mikolov et al.[32] ont utilisé un réseau neuronal récurrent (RNN) [33] pour construire un modèle de langage neuronal. Le RNN code le mot par mot et prévoit le mot suivant. Les poids du réseau formé sont utilisés en tant que vecteurs de words embeddings.

Mikolov et al. [34] ont proposé deux autres approches pour construire des représentations de mots dans un espace vectoriel. En utilisant une version simplifiée de Bengio et al. [35] mode de langage neuronal. Ils ont remplacé la couche cachée par une simple couche de projection afin d'améliorer les performances. Dans leurs travaux, deux modèles sont présentés: le modèle continu de sacs de mots (CBOW)[32] et le modèle de base (SKIP-G) [34].

Dans le premier, le modèle continu de mot CBOW, prédit un mot pivot selon le contexte en utilisant une fenêtre de mots contextuels autour de lui. Étant donné une suite de mots S = w1, w2, ..., wi, le modèle CBOW apprend à prédire tous les mots wk à partir des mots qui les entourent (wk − l, ..., wk − 1, wk + 1, ... , wk + l).

Le deuxième modèle SKIP-G prédit les mots environnants du mot-clé actuel wk [34].

Pennington et al. [36] ont proposé un Global Vectors (GloVe) pour construire un modèle de représentation de mots, GloVe utilise les statistiques globales de co-occurrence de mot-mot pour construire la matrice de co-occurrence M. Alors, M est utilisé pour calculer la probabilité d'apparition du mot wi dans le contexte d'un autre mot w j, cette probabilité P (i / j) représente la relation entre les mots.

D. Mesures de similarité hybride

Les mesures de similarité hybride ont été couvertes dans de nombreuses études où de multiples mesures de similarité ont été utilisées, elles utilisent les approches statistiques et topologiques.

Prenons comme exemple : Mihalcea et al. [37] huit mesures de similarité sémantique ont été évaluées, individuellement et combinées; la meilleure performance a été atteinte avec des mesures de similarité combinées. Une méthode pour mesurer la similarité sémantique entre des phrases ou des textes très courts, basée sur des informations sémantiques et l'ordre des mots, a été présentée dans Li et al. [38].les auteurs de [39] ont présenté une méthode appelée similitude de texte sémantique, qui détermine la similarité entre deux textes en fonction d'informations sémantiques et syntaxiques d'où on va inspirer notre méthode de similarité syntaxique.

iv. Les travaux sur la similarité des textes en utilisant la langue arabe

1. Les similarités syntaxiques

La similarité syntaxique en langue arabe, de nombreux chercheurs ont utilisé l'algorithme de distance de Levenshtein dont [40] 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 [40] 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. [41] 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.

[42] 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 Les 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. [43] 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'utilité du corpus et une difficulté à comparer les langues 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,…

Pour conclure en ce qui concerne la similarité syntaxique et sémantique, ce tableau 2 montre les avantages ainsi que les inconvénients de chacune d'elles :

	approches syntaxiques	Approches sémantiques
Avantages	-Les techniques basées sur l'approche syntaxique ne laissent pas de place aux exceptions. -Elles sont donc facilement automatisables.	-Ont montré de bons résultats en s'attaquant aux problèmes de la similarité lexicale (LSA).
Inconvénients	- ne prennent pas en compte la sémantique. Par exemple, il est difficile de trouver une forte similarité entre "Je possède un chien" et "J'ai un animal". -Les relations syntaxiques sont ignorées. Par exemple, aucune différence n'est faite entre "Amina invite Khadija" et "Khadija invite Amina". -De même, les rôles sémantiques sont ignorés. Par exemple, dans "La société A achète la société B" et "La société B a été achetée par la société A", seule la forme verbale change. Cela peut engendrer des problèmes de pertinence. -Les problèmes liés aux négations (par exemple, "Je suis malade" et "Je ne suis pas malade") semblent encore difficiles à pallier.	-les approches sémantiques basées sur les corpus ou la connaissance posent des problèmes de stockage et de complexité et sont souvent spécifiques à un domaine donné. -les problèmes liés à la négation sont encore mal pris en charge.

Tableau 2 : comparaison entre l'approche syntaxique et sémantique

v. Les enjeux de la langue arabe dans le contexte de l'évaluation automatique

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 [44]. 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 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 défi est 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 [45][46].

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 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.

vi. Les tâches NLP considérées dans notre travail

1. La lemmatisation :

La lemmatisation est le processus de cartographie et de transformation de toutes les formes fléchies de ce mot en une forme commune, partagée et canonique et, par conséquent, cette forme canonique serait la forme la plus appropriée pour l'indexation et pour la recherche, aussi bien. En d'autres termes, la lemmatisation transforme différentes formes et variantes d'un certain mot en un seul mot canonique [46].

2. La normalisation:

La normalisation est le processus de production de la forme canonique d'un mot afin de maximiser la correspondance entre un mot de requête et autre de collection de mots dans un document. Dans sa forme simple, la normalisation prétraite les mots en une seule forme, mais très légèrement. Cela se fait souvent en plusieurs étapes de prétraitement afin de rendre différentes formes d'une lettre particulière à une seule représentation Unicode, par exemple, en remplaçant la lettre arabe non pointillée par une lettre finale en pointillé, lorsque cette lettre apparaît à la fin d'un mot arabe [47].

La figure 7 représente certaines normalisations appliquée en langue arabe :

MSA	Variant	Gloss	Typographical Occurrence
امتحان	إمتحان	Exam	The final bare ALIF is changed to ALIF HAMZA below
صفاء	صفا	Purity	The final HAMZA is dropped
قرآن	قران	The Quran	ALIF MADDA in the middle is altered to bare ALIF
علاء	علا	A proper noun	They compute (plural feminine)
نافذه	نافذة	Window	The final letter HAA is altered to a different letter, which is TAA MARBOOTA
زراعي	زراعى	Agricultural	The final dotted YAA is changed to un-dotted YAA

Figure 7 : un exemple de normalisation

3. La lemmatisation en langue arabe

Il existe deux majeures approches de lemmatisation en langue arabe :

La lemmatisation lourde

L'unité de base en arabe est la racine, la technique de racine basée sur les racines tente d'effectuer une analyse morphologique heuristique et linguistique afin d'extraire la racine d'un mot. Par exemple, les algorithmes basés sur la racine produisent la racine عمل pour le mot وأعمالهم (ce qui signifie: et leurs œuvres) parce que tous les affixes sont enlevés. Pour atteindre cet objectif d'enracinement, les chercheurs utilisent les analyseurs morphologiques arabes.

Parmi les stemmers lourds, on mentionne celui que nous avons utilisé :

Khoja stemmer : est l'un des stemmers à base de racines les plus célèbres. L'algorithme a été largement utilisé en arabe dans la recherche d'information. Il transforme les mots en leurs racines en supprimant leurs préfixes et leurs suffixes les plus longs au début. Par exemple, le préfixe ي et le suffixe ون sont d'abord supprimés, si le mot d'entrée est يلاعبون (ce qui signifie: ils jouent avec). Le mot qui en résulte (dans ce cas, le mot لاعب) est ensuite associé à des motifs prédéfinis et à des racines basées sur des listes. Le motif sélectionné dépend de la longueur du mot extrait. Par exemple, pour le mot لاعب dans notre exemple, le

motif فاعل peut être choisi. Par ce processus d'appariement la racine est produite comme لعب (signifiant: jouer) puisque le modèle فاعل est déjà prédéfini dans le langage qui a une lettre nue ALIF (ا) ajoutée médialement au motif trilatéral فعل. Enfin, dans l'algorithme, la racine extraite est comparée à une liste de racines pour vérifier sa validité [48].

Un avantage de Khoja stemmer est qu'il a la capacité de détecter les lettres qui ont été supprimées pendant le processus de dérivation des mots. Par exemple, la dernière lettre YAA est supprimée dans un mot comme (امشي signifiant: aller), résultant en, امش si elle apparaît sous une forme impérative. Comme autre exemple, la dernière lettre ALIF dans la racine (نما signifiant: développé) sera modifiée en WAW sous la forme actuelle de cette racine et sera donc نمو au lieu de .نما En utilisant Khoja stemmer, il est possible de traiter de tels cas.

On mentionne aussi :

Darwish (Sebawai à root-based) [49],

Buckwalter [50]

Abdelali [51]

Ghwanmeh, et al [52]

Al-Kabi, et al, [53].

La lemmatisation légère

Pour atténuer l'impact de l'inconvénient majeur des algorithmes basés sur la racine, qui fait perdre la notion de sémantique, une lemmatisation légère provenant de l'arabe a également été proposée. Les stemmers légers coupent quelques affixes tels que les terminaisons plurielles en anglais à partir des mots et sans effectuer d'analyse linguistique profonde. De ce point de vue, la majorité des approches tentent de supprimer les préfixes les plus fréquents (c'est-à-dire articles), les suffixes (c'est-à-dire les pronoms possessifs) et tous les préfixes ou suffixes cela peut être attaché au début ou à la fin des mots. Par exemple, les stemmers légers génèrent أعمال (ce qui signifie: œuvres) de اعمالهم parce que seuls les préfixes (y compris les antéfixes) et les suffixes (y compris les post fixes) sont supprimés. La décision de retirer cependant, les affixes sont généralement contrôlés par des règles heuristiques dérivées de l'utilisation commune de ces antéfixes [47].

Comme exemple d'un stemmer léger :

Al-stem : est un stemmer léger, présenté par Darwish et Oard [54], qui légèrement coupe les préfixes suivants mais dans l'ordre de droite à gauche,(بت, وال،فال،بال،
يت،لت،مت،وت،ست،نت،بم،لم،وم،كم،فم،ال،لل،في،وا،وا،فا،لا،با),ainsi que les suffixes commençant de droite à gauche(تك،نا،ين،يه،ة،ه،ي،ا.ها, ،ات،وا،ون،وه،ان،تي،ته،تم،كم،هم،هن,), Al-Stem était comparé à light8 stemmer, résultats conclu qu'il n'y a presque pas de différence statistiquement entre les deux stemmers quand ils ont été testés en utilisant les données TREC 2001. Plus tard, Al-Stem a été modifié par David Graff du Linguistic Data Consortium (LDC) pour supprimer les suffixes (تا et ا) et les préfixes (سي et تت) de la liste des suffixes dans Al-Tige.

On mentionne aussi : le light stemmer que nous avons utilisé :

Stanford stemmer :

C'est un ensemble d'outils d'analyse et technologiques en langage humain, il traite de différentes tâches complexes comme la reconnaissance des noms propre, baliser la structure des phrases en termes de phrases et de dépendances syntaxiques (Part-Of speach), effectuer un stem léger… De même il est conçu pour des finalités NLP et disponible pour plusieurs langues. Il existe aussi le Light10.

vii. Les travaux connexes à notre recherche

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). Dans ce contexte notre travail est connexe aux travaux menés par Gomaa & al [55]. 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 les approches de distance de n-gramme. Dans la deuxième étape, ils ont mesuré la similarité en utilisant des mesures de similarité basées sur le corpus [56]: DISCO1 (Calcule la similarité du premier ordre entre deux mots basés sur leurs ensembles de collocation) et 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 mélangeant n-gramme avec les techniques de similarité DISCO1. En utilisant le SemEval Dataset nous allons avoir une indication sur la généralisation de nos approches dans des domaines connexes en les comparants aux résultats de la compétition 2017 fournis dans SEMEval 2017 [57].

Le travail de Vectorized [58] et Nagoudi [59] sont tous aussi intéressants en considérant leurs résultats par rapport à une approche basée sur le calcul vectoriel et les word Embedding. Vectorized ont évalué leur approche sur Gomaa Dataset alors que Nagoudi 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.

Wael Hassan Gomaa et Aly Fahmyont traité trois types de similarité à base de chaînes, LCS, N-gramme, damerau levenshtein. L'approche N-gramme a permis d'obtenir de meilleurs résultats r et RMSE.

La distance DL détermine la distance entre deux chaines de caractères S1 et S2 selon le nombre minimum d'opérations qui sont nécessaires pour transformer une chaîne de caractères à une autre. L'opération peut être soit insertion, suppression ou substitution d'un seul caractère, ou la transposition de deux caractères adjacents [60].

Le LCS considéré dans ce travail se base sur la similitude entre les deux chaînes selon la longueur de la chaîne de caractères commune continue qui existe dans les deux chaînes, ce qu'on va critiquer par la suite dans notre expérience.

De plus, leur résultats ont montré que plusieurs mesures semblent être meilleures lors de l'évaluation avec la corrélation de Pearson, tandis que d'autres semblent être meilleures lors de la mesure avec le RMSE. La meilleure valeur de la corrélation r était de 0,73, résultat de l'approche basée sur les caractères de Bi-gram utilisant la méthode d'échelle de Clust11.

Dans le travail de H.Gomaa et A.Fahmy [1], les algorithmes syntaxiques expérimentés tels que Les méthodes de Ngrammes, LCS, ils ont été expérimentés en appliquant plusieurs méthodes, en premier , ils ont utilisé ce qu'ils ont appelé la méthode Raw , celle-ci veut dire aucun outil NLP n'est appliqué, la méthode Stop qui applique la suppression des mots d'arrêt en utilisant la liste d'arrêt qui contient 387 mots, et puis en utilisant (ISRI) l'outil de stem arabe, utilisé pour remplacer chaque mot qui n'est pas un mot d'arrêt avec son stem sans

enlever les mots d'arrêts, et puis à la fin, ils ont utilisés une méthode qui supprime les mots d'arrêts et au même temps une lemmatisation, les résultats obtenus était en faveur d'utiliser la dernière méthode, c'est-à-dire en appliquant une lemmatisation et en enlevant les mots d'arrêts, et c'est toujours le ngram qui a donné le meilleur résultat dans ce travail [55].

En considérant ces travaux nous tentons d'améliorer les résultats obtenus dans les différentes étapes (syntaxiques dans le cadre de ce travail) ensuite atteindre une meilleure hybridation en termes de maximisation du coefficient de Pearson et de minimisation de l'erreur quadratique.

Conclusion

Une vue générale et encore détaillé sur les travaux qui ont été exploités dans la littérature ont été présentés dans ce chapitre, y compris les différents concepts à connaitre pour mieux comprendre la suite de travail.

III. Système d'évaluation automatique des réponses courtes

Dans ce chapitre, nous allons présenter notre système, on commence par expliquer son fonctionnement, c'est-à-dire le prétraitement des données, les approches syntaxiques utilisées y compris, les approches qui existent dans la littérature et nos approches syntaxiques proposées, on passera par la suite à expliquer le processus d'évaluation de système, c'est-à-dire les data set utilisés, ainsi que les méthodes utilisées pour l'évaluer et on termine par une explication détaillé de notre système implémenté et un aperçu des interfaces de ce dernier.

i. Ressources matérielles et logicielles utilisées

Nous avons travaillé avec le langage de programmation *Python 3.6*, et en utilisant Sublime text 3 comme éditeur de texte version 3.0, Build 3170.

Pour les interfaces, nous avons utilisé l'environnement de développement Netbeans 8.2.

ii. Fonctionnement de notre outil d'évaluation automatique

Le professeur va introduire sa réponse modèle ou bien l'ensemble de ses réponses modèles à une question donnée, l'étudiant va introduire par la suite sa réponse à cette question, à cette étape là , le premier processus qui se passera, est le processus de prétraitement des réponses, en utilisant la lemmatisation et normalisation, ensuite un ensemble des approches de similarités vont être appliqués sur les deux réponses, en hybridant plusieurs approches ou bien en utilisant qu'une seule pour avoir une similarité entre 0 et 1, à la fin cette similarité est passé à la note que le professeur veut accorder à cette question soit en utilisant le classifieur de Kmeans, ou bien en utilisant la multiplication (voir la figure 8)

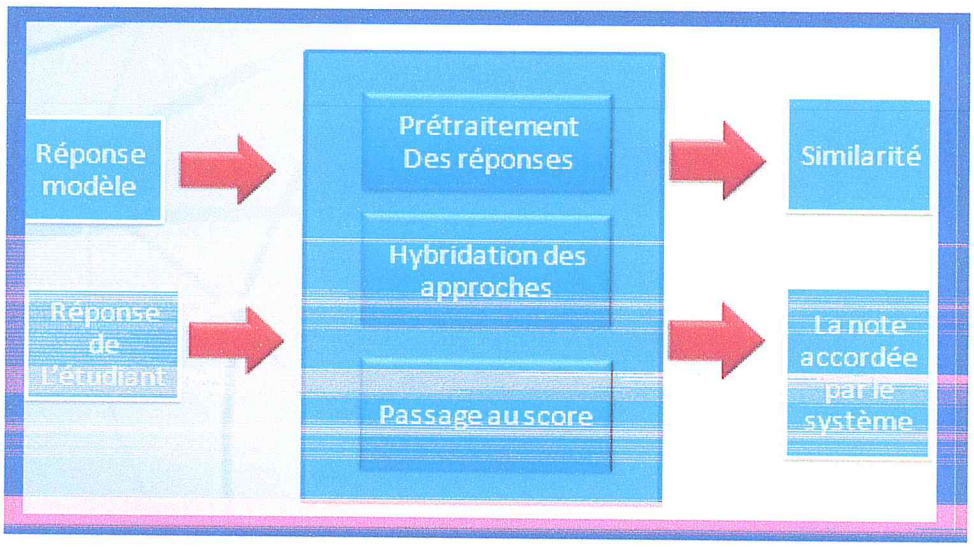

Figure 8 : Fonctionnement de notre système

iii. Prétraitement des données

Les réponses qui vont être introduites dans notre système, vont passer par plusieurs processus, le premier processus est le prétraitement de ces dernières, c'est-à-dire la lemmatisation et la normalisation

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 ambigue 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 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 [61].

Pour toutes les approches de similarités développées nous avons considéré les 3 cas suivants :

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,
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.

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.

Le stemming consiste en général à réaliser les actions suivantes pour chaque couple de réponses à comparer. :

- Suppression des nombres des deux réponses.
- Suppression des signes diacritiques des deux réponses.
- Suppression de toutes les lettres d'autres langues.
- Supprimer les mots d'arrêt. Une liste de mots d'arrêt est disponible dans la base de données
- Enlever le (ال : AL), et ses Dérivés, (كال, وال, تال, وبالفبال, لبال, فال, ال, لل, بال, ...)
- Normaliser les mots en remplaçant des lettres similaires : le détail de notre normalisation utilisée est dans la figue 9 :

```
# supp lettre non arabe
for x in range(len(alpha)):
    if alpha[x] in txt:
        txt = txt.replace(alpha[x],"")
# normalisation
for x in range(len(beta)):
    if beta[x] in txt:
        txt = txt.replace(beta[x],"ا")
txt = txt.replace("ة","ه")
txt = txt.replace("ى","ي")
txt = txt.replace("ؤ","ؤ")
txt = txt.replace("ئ","ئ")
name+=1
with c.open("StemmedCorpus\\"+str(name)+".txt","w","utf-8")as f:
    f.write(txt)
```

Figure 9 : la normalisation utilisée

- 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 le stemming est lourd ou léger.

Il est à noter 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 :

- Khoja Stemmer[2] [61] pour un stemming lourd.
- Standford Stemmer[3] pour un stemming léger.

Les deux figures 10 et 11 suivantes montrent un exemple d'approche de stemming léger et lourd :

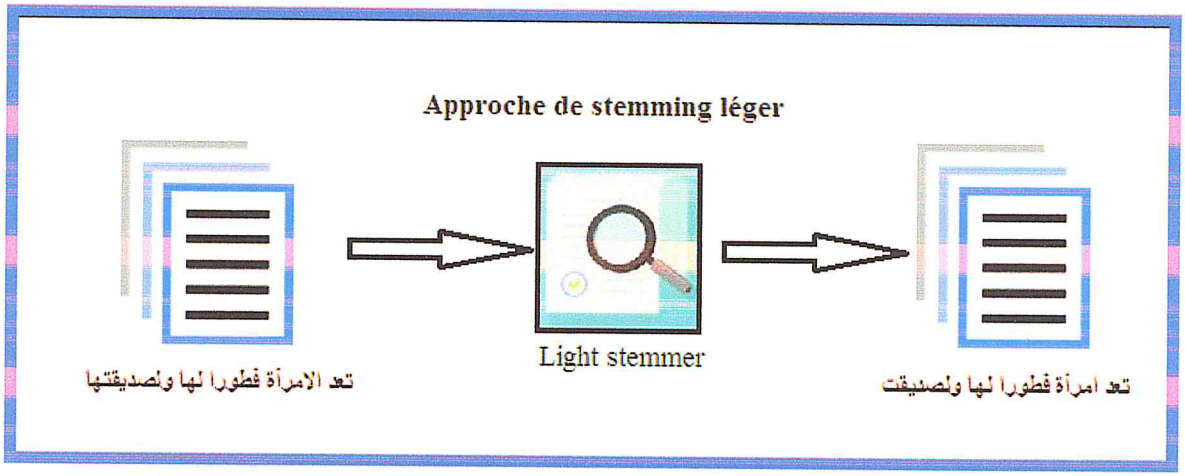

Figure 10: La lemmatisation légère

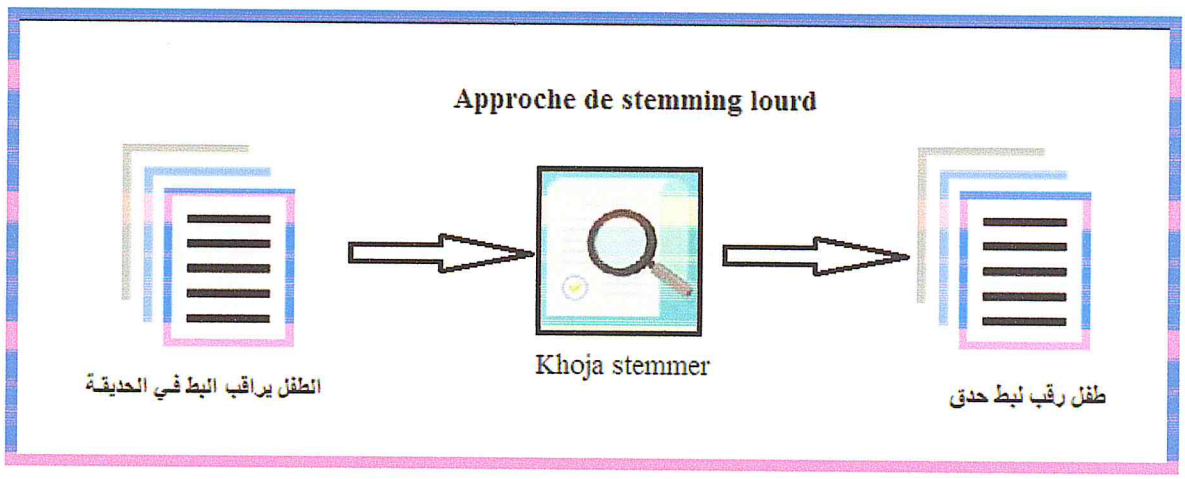

Figure 11 : La lemmatisation lourde

[2] https://github.com/motazsaad/khoja-stemmer-command-line/blob/master/khoja/KhojaStem.java
[3] https://stanfordnlp.github.io/CoreNLP/

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

- Light10 Stemmer[4]
- Khoja Stemmer[5]
- ISRI Stemmer[6]
- Tashaphyne Stemmer[7]
- Motaz Stemmer[8]
- Assem Stemmer[9]

Le tableau 3 en dessous montre l'impact des différents stemmers sur la même phrase :

	Word	Khoja	Light 10	ISRI	Tashaphyne	Motaz	Assem
الدراسة التي تتناول جوانب الطبيعة بما يحدد حياة الكائن و كيفية استخدامه لمكونات البيئة.	الدراسة	درس	دراس	درس	دراس	دراس	دراس
	التي	التي	ت	لتي	ى	تي	التي
	تتناول	نول	ناول	نول	ناول	تتناول	تتناول
	جوانب	جنب	جوانب	جنب	جوانب	جوانب	جوانب
	الطبيعة	طبع	طبيع	طبع	طبيع	طبيع	طبيع
	بما	ما	ا	بما	م	بما	بما
	يحدده	حدد	يحدد	حدد	حدد	يحدد	يحدد
	حياة	حيا	حيا	حية	حيا	حيا	حيا
	الكائن	كون	كا	كئن	كائن	كائن	كاء
	و	و	و	و		و	و
	كيفية	كيف	كيف	كيف	يف	كيف	كيف
	استخدامه	خدم	استخدام	استخدامه	خدام	استخدام	استخدام
	لمكونات	كون	كون	مكو	كونا	لمك	مكونا
	البيئة	بيئة	بيا	ئة.	بيئة.	بيئه.	بيء

Tableau 3: un apperçu de l'imapct des différents stemmers sur une phrase

L'appréciation de groupe était basé sur le sens du mot après l'avoir stemmé, on donne un exemple pour l'évaluation et le choix entre deux stemmers lourds ISRI et Khoja dans les

[4] http://zeus.cs.pacificu.edu/shereen/research.htm#stemming

[5] http://arabic.emi.ac.ma:8080/SafarWeb_V2/faces/safar/morphology/stemmer.xhtml

[6] http://www.nltk.org/_modules/nltk/stem/isri.html

[7] https://pypi.org/project/Tashaphyne/

[8] https://github.com/motazsaad/arabic-light-stemmer

[9] http://www.arabicstemmer.com/

tableaux 4 et 5, on voit bien que Isri, sa lemmatisation change beaucoup le sens du mot contrairement à Khoja, même si khoja n'a pas atteint le degré de perfection.

Word	Khoudja	Type	Evaluation
الإطار	طور	root	0
الذى	الذى	stop word	1
يحيا	حيا	root	1
فيه	فيه	stop word	1
الإنسان	أنس	root	1
مع	مع	stop word	1
غيره	غور	root	1
من	من	stop word	0
الكائنات	كون	root	0
الحية	حيا	root	0
و	و	stop word	1
يحصل	حصل	root	1
منها	منها	stop word	1
على	على	stop word	1
مقومات	قوم	root	0
حياته	حيا	root	1
			11

Tableau 4: Exemple d'évaluation de stem Khoja

Word	ISRI	Type	Evaluation
الإطار	اطر	root	0
الذى	الذى	stop word	1
يحيا	يحا	root	0
فيه	فيه	stop word	1
الإنسان	سان	root	0
مع	مع	stop word	1
غيره	غير	root	1
من	من	stop word	1
الكائنات	كئن	root	0
الحية	لحي	root	0
و	و	stop word	1
يحصل	حصل	root	1
منها	منها	stop word	1
على	على	stop word	1
مقومات	قوم	root	0
حياته	حياته.	root	1
			10

Tableau 5: Exemple d'évaluation de stem Isri

iv. Hybridation des approches

Après avoir effectué le prétraitement des réponses, un ensemble des approches de similarité syntaxique vont êtres appliquées pour pourvoir mesurer la similarité entre les réponses, nous allons exposer en premier, les approches syntaxiques qui existent dans la littérature et puis nos approches syntaxiques proposées, notre outil permet d'hybrider n'importe quel approche avec autre, ou bien hybrider plusieurs approches pour obtenir le meilleur résultat possible.

1. Notions importantes pour l'application des approches syntaxiques

A. La représentation d'union et d'intersection

La représentation des vecteurs dans le cas de similarité Cosine, Manhattan, ou l'euclidienne nécessite ces deux représentations, union et intersection pour avoir des vecteurs de même taille, car l'application des formules de ces trois similarités nécessite des vecteurs de même taille, dans le cas ou les deux phrases à comparer sont de différentes tailles on va passer par :

Une représentation d'union : La longueur de chaque vecteur est égale au nombre des mots des deux phrases sans répétition.

Une représentation d'intersection : La longueur de chaque vecteur est égale au nombre des mots en commun entre les deux phrase

Remarque : La représentation par union est la plus utilisée.

B. Le passage au pourcentage :

Certaines similarités donnent une estimation supérieure à 1 comme la distance de levenshtein et Neeleman ce qui exige un passage au pourcentage pour donner une similarité entre 0 et 1.

Le passage au pourcentage pour une similarité entre une phrase string1 et string2 nécessite la formule suivante :

$$1 - \frac{sim(string_1, string_2)}{\max\left(len(string_1), len(string_2)\right)}$$

Où ($len(string_1)$ (resp.$len(string_2)$), Est la longueur de la chaine.

2. Les similarités syntaxiques

La figure 12 représente Les approches de similarités syntaxiques :

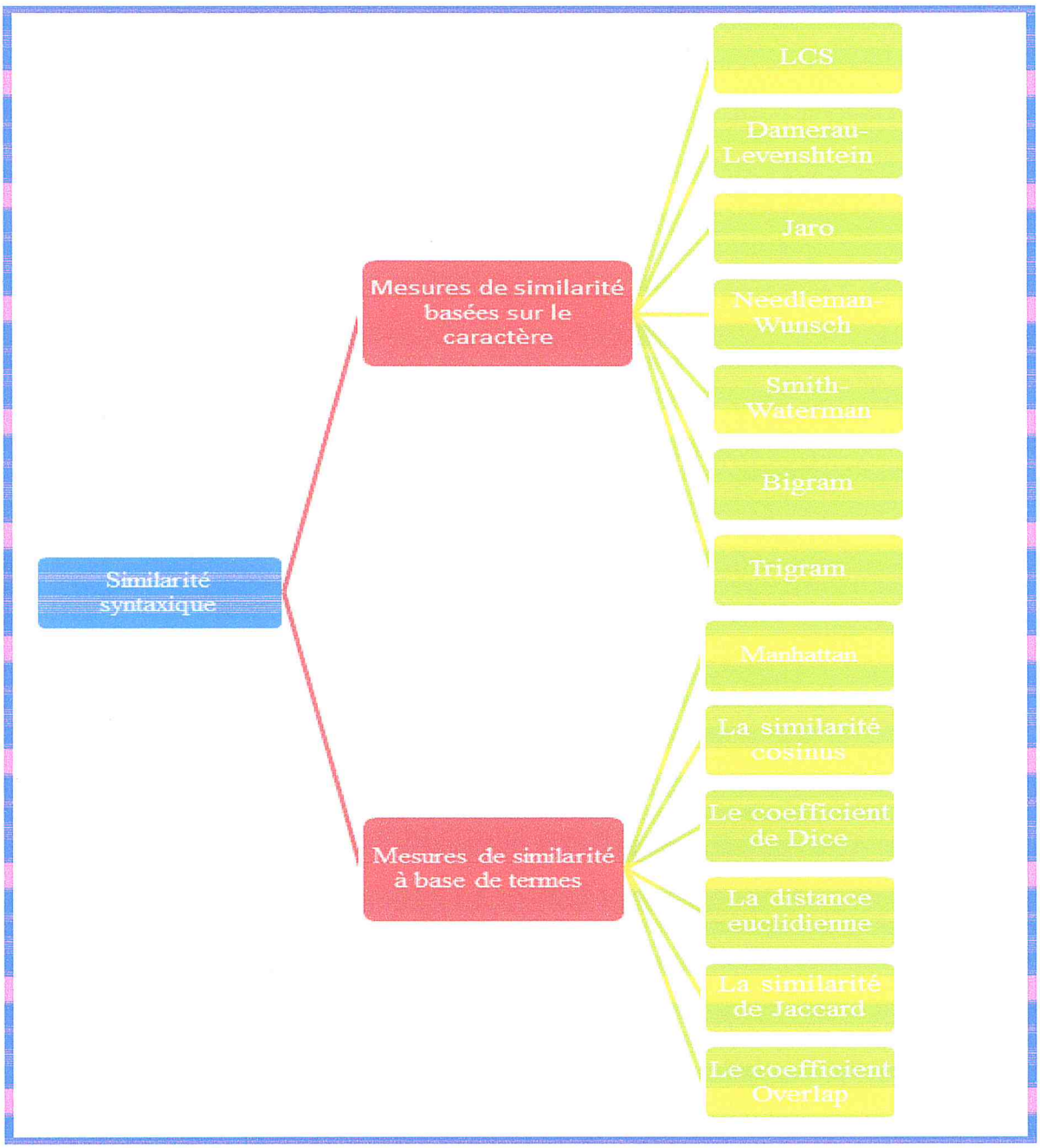

Figure 12 : Les approches de similarité syntaxique

A. Mesures de similarité à base de caractère

➢ **LCS (Longest Common SubString)** : est un algorithme qui considère la chaine commune la plus longue. La plus longue sous-séquence commune à deux suites, ou deux chaînes de caractère, est une séquence étant sous-suite des deux suites, et étant de taille maximum. La résolution de ce problème peut être obtenue par programmation dynamique.

Exemple :

Soit :

P= جالس, et R= يجلسان deux mots.

Où : n est la taille de mot P, n = 4 et m la taille de mot R, m = 6.

Soit LCS la plus longue sous-séquence commune à P et R :

LCS= جلس, et la longueur LCS =3 :

$$sim_{LCS}(P,R) = \frac{len(LCS)}{Max(len(P),len(R))}$$

$$= \frac{3}{6} = 0.5$$

➢ **Damerau-Levenshtein** : aussi connue sous le terme *distance d'édition,* il considère la distance entre deux chaînes en comptant le nombre minimum d'opérations nécessaires pour transformer une chaîne en une autre, l'opération est définie comme une insertion, une suppression ou une substitution d'un seul caractère[10].

Levenshtein distance entre deux chaînes a et b :

$$lev_{a,b}(i,j) =$$

$$\begin{cases} \max(i,j) \; if \; \min(i,j) = 0, \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \\ lev_{a,b}(i-2,j-2)+1 \end{cases} \quad if \; a,b > 1 \; and \; a_i = b_{j-1} and \; a_{i-1} = b_j \\ min \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} \quad autrement. \end{cases}$$

Où $1_{(a_i \neq b_j)}$

[10] https://en.wikipedia.org/wiki/Damerau%E2%80%93Levenshtein_distance

est la fonction d'indicateur égale à 0 quand $a_i = b_j$ et égal à 1 sinon, et $lev_{a,b}(i,j)$ est la distance entre les premiers caractères i de a et les premiers caractères j de$b$.

Exemple :

Soit :

P= الرجل, et R= رجلان, deux mots.

Où : n est la taille de mot P, n = 5 et m la taille de mot R, m = 5.

- **Construction de la matrice de Levenshtein :**

ل	ج	ر	ل	ا		
5	4	3	2	1	0	
4	3	2	2	1	1	ر
3	2	3	2	2	2	ج
2	3	3	2	3	3	ل
3	4	3	3	3	4	ا
4	4	4	4	4	5	ن

On a : Edit distance entre P et R = 4

Passage au pourcentage :

$$\text{sim}_{Levenshtein}(P,R) = 1 - \frac{Edit\ distance}{Max\big(len(P), len(R)\big)}$$

$$= 1 - \frac{4}{Max(5,5)} = 0.2$$

➢ **Jaro** est basé sur le nombre et l'ordre des caractères communs entre deux chaînes; il prend en compte les directives orthographiques typiques et est principalement utilisé dans le domaine du couplage d'enregistrements [62][63].

La distance de Jaro entre chaînes s_1 ets_2 est définie par :

$$d_{Jaro} = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right)$$

Où :

- $|s_i|$est la longueur de la chaîne de caractèress_i.
- Mest le nombre de caractères correspondants.
- $t$est le nombre de transpositions.

Deux caractères identiques de s_1 et de $s_2$sont considérés comme correspondants si leur éloignement (i.e. la différence entre leurs positions dans leurs chaînes respectives) ne dépasse pas :

$$\left\lfloor \frac{\max(|s_1|, |s_2|)}{2} \right\rfloor - 1$$

Le nombre de transpositions est obtenu en comparant le i-ème caractère correspondant de $s_1$avec le i-ème caractère correspondant de s_2.Le nombrede fois où ces caractères sont différents, divisé par deux, donne le nombre de transpositions.

Exemple :

Soit :

P=الوسط, et R=الإنسان, deux mots.

Où : n est la taille de mot P, n = 5 et m la taille de mot R, m = 7.

m est le nombre de caractères correspondants.

t est le nombre de transpositions.

$$d_{Jaro} = \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right)$$

On a : $m = 3$ et t= 0

$$d_{Jaro} = \frac{1}{3}\left(\frac{3}{5} + \frac{3}{7} + \frac{3-0}{3}\right)$$

d_{Jaro}= 0.676

➢ **Needleman-Wunsch** Il effectue un alignement global pour trouver le meilleur alignement sur l'ensemble des deux séquences. Il est approprié lorsque les deux séquences

sont de longueur similaire, avec un degré significatif de similitude tout au long [64].

Exemple :

Soit :

P= الحياة, et R= الحية, deux mots.

Où : n est la taille de mot P, n = 6 et m la taille de mot R, m = 5.

Où : Match = +2, Incompatibilité = -1, Indel = -1

- **Construction de la matrice de notation :**

Le principe de remplissage de notre matrice est simple, si on a deux caractères identiques, alors la case associé est égale à 1, si les caractères sont différents, on va soustraire un 1 de chaque case qui entoure la case courante, puis on prend en considération la case ayant la valeur la plus grande.

ة	ا	ي	ح	ل	ا		
-6	-5	-4	-3	-2	-1	0	
-4	-3	-2	-1	0	1	-1	ا
-4	-3	-2	-1	2	0	-2	ل
0	1	2	3	1	-1	-3	ح
2	3	4	2	0	-2	-4	ي
4	3	3	1	-1	-3	-5	ة

Score obtenu de la matrice = 4

Meilleur alignement : Alignement = الحية

Longueur de meilleur alignement = 5

Longueur d'alignement: 6

$$sim_{Needleman} = \frac{Longueur\ de\ l'alignement}{\text{Longueur d'alignement}} = \frac{5}{6} = 0.833$$

➢ **Smith-Waterman** Il effectue un alignement local pour trouver le meilleur alignement sur le domaine conservé de deux séquences. Il est utile pour les séquences dissemblables qui sont suspectées de contenir des régions de similarité ou des motifs de séquence similaires dans leur contexte de séquence plus grand [65].

Exemple :

Soit :

P=مشاركة , et R=تشترك , deux mots.

Où : n est la taille de mot P, n = 6 et m la taille de mot R, m = 5.

Où: Match = +2, Mismatch = -1, gap= -1

- **Construction de la matrice de notation :**

  Le principe de son remplissage est le même que needlman, sauf pour les caractères qui sont différents au lieu de soustraire un 1, on va soustraire un 2, et prendre la valeur la plus grande.

ة	ك	ر	ا	ش	م		
0	0	0	0	0	0	0	
2	-1	-1	-1	-1	-1	0	ت
1	-2	-1	0	1	-1	0	ش
0	-2	-1	0	0	-1	0	ت
0	1	2	-1	-1	-1	0	ر
3	4	1	-2	-2	-1	0	ك

Score obtenu de la matrice = 3

Meilleur alignement : Alignement = شرك

Longueur de meilleur alignement = 3

Longueur d'alignement: 4

$$sim_{Smith-Waterman} = \frac{Longueur\ de\ l'alignement}{\text{Longueur d'alignement}} = \frac{3}{4} = 0.75$$

➢ **Bigram** est une séquence de deux éléments adjacents d'une chaîne de jetons, qui sont généralement des lettres, des syllabes ou des mots. Un bigram est un n-gramme pour n = 2. La distribution de fréquence de chaque bigram dans une chaîne est couramment utilisée pour l'analyse statistique simple du texte dans de nombreuses applications, y compris en linguistique computationnelle, cryptographie, reconnaissance vocale, et ainsi de suite. dans le cas de similarité, on considère les bigrams en commun entre les termes[11].

Exemple :

Soit :

P= يظهر, et R= الظهر, deux mots.

Où : n est la taille de mot P, n = 4 et m la taille de mot R, m = 5.

Avec n=2 on aura :

P= {يظ, ظه, هر}

R= {ال, لظ, ظه, هر}

donc :

Termes en commun = {ظه, هر}

Longueur des termes en commun = 2

P union R = {يظ, ظه, هر, ال, لظ, ظه, هر }

Longueur de P union R = 7

[11] https://en.wikipedia.org/wiki/Bigram

$$sim_{Bigram} = 1 - \frac{longueur(termes\ en\ commun)}{longueur(tous\ les\ termes)}$$

$$sim_{Bigram} = 1 - \frac{2}{7} = 0.714$$

➢ **Trigram** est une séquence de trois éléments adjacents d'une chaîne de jetons, qui sont généralement des lettres, des syllabes ou des mots. Un trigram est un n-gramme pour n = 3. La distribution de fréquence de chaque trigram dans une chaîne est couramment utilisée pour l'analyse statistique simple du texte dans de nombreuses applications, y compris en linguistique computationnelle, cryptographie, reconnaissance vocale, et ainsi de suite. dans le cas de similarité, on considère les trigrams en commun entre les termes.

Exemple :

Soit :

P= امرأة, et R= المرأ, deux mots.

Où : n la taille de mot P, n = 5 et m la taille de mot R, m = 5.

Avec n=3 on aura :

P= {امر, رأة }

R= {الم, مرأ}

donc :

Termes en commun = { }

Longueur des termes en commun = 0

$sim_{trigram} = 0$

B. Mesures de similarité à base de termes

➢ **Manhattan** Il calcule la distance qui serait parcourue pour se rendre d'un point de données à l'autre si un chemin semblable à une grille est suivi. La distance entre deux éléments est la somme des différences de leurs composantes correspondantes [66].

Soient A et B, deux points de coordonnées respectives (X_A, Y_A) et(X_B, Y_B), la distance de Manhattan est définie par :

$$d(A,B) = |X_B - X_A| + |Y_B - Y_A|$$

Exemple :

Soit :

P= {تمشي النساء جنبا إلى جنب}

R= {هناك فتيات يمشين متجاورات}

Où : n est la taille de la réponse P, n = 5 et m la taille de la réponse R, m = 4.

On fait l'union des deux vecteurs on aura deux vecteurs de même taille et avec une représentation booléenne comme c'est montré dans le tableau 6 :

	متجاورات	يمشين	فتيات	هناك	جنب	إلى	جنبا	النساء	تمشي
P	0	0	0	0	1	1	1	1	1
R	1	1	1	1	0	0	0	0	0

Tableau 6: Représentation vectorielle des réponses (Manhattan)

On calcule la distance de manhattan :

$$d_{manhattan} = |0-1| + |0-1| + |0-1||0-1| + |1-0| + |1-0| + |1-0| + |1-0| + |1-0| + |1-0|$$

$$d_{manhattan} = 1+1+1+1+1+1+1+1+1$$

$$d_{manhattan} = 9$$

Passage au pourcentage :

$$sim_{\ manhattan} = \left|1 - \frac{d_{manhattan}}{\max(len(v1), len(v2))}\right|$$

$$= \left|1 - \frac{9}{\max(5,4)}\right|$$

$$= |1 - 1.8|$$

$$= 0.8$$

➢ **La similarité cosinus** La similarité cosinus est fréquemment utilisée en tant que mesure de ressemblance entre deux documents $d1$ et $d2$. Il s'agit de calculer le cosinus de l'angle entre les représentations
vectorielles des documents à comparer [23].

$$sim_{cosinus}(d_1, d_2) = \frac{\overrightarrow{d_1}.\overrightarrow{d_2}}{\|\overrightarrow{d_1}\|\|\overrightarrow{d_2}\|}$$

Exemple :

Soit :

P= {هي البيئة التي تصنعها التكنولوجيا}

R= {البيئة التي صنعها الإنسان بعلمه و تقدمه}

Où : n est la taille de la réponse P, n = 5 et m la taille de la réponse R, m =7.

On fait l'union des deux vecteurs on aura deux vecteurs de même taille (voir tableau7) :

	تقدمه	و	بعلمه	الإنسان	صنعها	التكنولوجيا	تصنعها	التي	البيئة	هي
P	0	0	0	0	0	1	1	1	1	1
R	1	1	1	1	1	0	0	1	1	0

Tableau 7: Représentation vectorielle des réponses (Cosine)

On calcule la distance de cosine:

$$sim_{Cosine}(P,R) = \frac{(1*1)+(1*1)}{\sqrt{len(v1)}*\sqrt{len(v2)}}$$

$$= \frac{2}{\sqrt{5}*\sqrt{7}}$$

$$= 0.338$$

➢ **Le coefficient de Dice** mesure la similarité entre chaînes de caractères. Étant donnés deux chaînes x et y, on peut calculer le coefficient comme suit :

$$sim_{dice}(x,y) = \frac{2n_t}{n_x + n_y}$$

Où n_t est le nombre de digrammes (formés de deux caractères consécutifs) communs aux deux chaînes, n_x est le nombre de digrammes dans x et n_y le nombre de digrammes dans y.

Exemple :

Soit :

P= {البيئة التي صنعها الإنسان بعلمه و تقدمه}

R= {البيئة التي يصنعها الإنسان بعلمه و تقدمه}

Où : n la taille de la réponse P, n = 7 et m la taille de la réponse R, m =7.

On calcule les digrammes de chaque réponse:

P= {' ه', 'قد', ' ب', 'ى ', ' ص', 'ن ', 'عه', 'ا ', 'سا', 'دم', 'نع', 'تق', 'عل', 'لب', 'لإ', 'ال', 'ها', ' و', 'ان', 'لت', ' ' ة', 'ئة', ' ا', ' ت', 'إن', 'نس', 'صن', 'يئ', 'مه', 'بع', 'بي', ' و', 'لم', 'تي'}

R= {' ه', 'قد', ' ب', 'ى ', 'ن ', 'عه', 'ا ', 'سا', 'دم', 'نع', 'تق', 'عل', 'لب', 'لإ', 'ال', 'ها', ' و', 'ان', 'لت', ' ي', ' ' ة', 'ئة', ' ا', ' ت', 'إن', 'نس', 'صن', 'يئ', 'مه', 'بع', 'يص', 'بي', ' و', 'لم', 'تى'}

Où : l'ensemble P a 34 éléments, et l'ensemble R a 35 éléments, et leur intersection donne 33 éléments.

{' ه', 'قد', ' ب', 'ى ', 'ن ', 'عه', 'ا ', 'سا', 'دم', 'نع', 'تق', 'عل', 'لب', 'لإ', 'ال', 'ها', ' و', 'ان', 'لت', 'تي', 'لم', 'و ' ' ة', 'ئة', ' ا', ' ت', 'إن', 'نس', 'صن', 'يئ', 'مه', 'بع', 'بي', ' '}

Avec la formule donnée ci-dessus, on obtient :

$$sim_{dice}(x,y) = \frac{2 * 33}{34 + 35} = 0.956$$

➢ **La distance euclidienne** calcule la similarité entre deux documents *d*1 et *d*2 comme la distance entre leurs représentations vectorielles ramenées à un seul point.

$$sim_{euclidienne}(d1,d2) = \|\overrightarrow{d_1} - \overrightarrow{d_2}\| = \sqrt{\sum_{i=1}^{n}(d_{1_i} - d_{2_i})^2}$$

Où *n* est le nombre total de termes représentés, i.e. la taille des vecteurs [4].

Exemple :

Soit :

P= {تزداد نسبة الملوحة في البحار}

R= {تزداد درجة ملوحة هذه البحار}

Où : n est la taille de la réponse P, n = 5 et m la taille de la réponse R, m =5.

On fait l'union des deux vecteurs on aura deux vecteurs de même taille(voir tableau 8) :

	هذه	ملوحة	درجة	البحار	في	الملوحة	نسبة	تزداد
P	0	0	0	1	1	1	1	1
R	1	1	1	1	0	0	0	1

Tableau 8: Représentation vectorielle des réponses(Euclidienne)

$$d_{euclidienne} = \sqrt{(0-1)^2 + (0-1)^2 + (0-1)^2 + (1-1)^2 + (1-0)^2 + (1-0)^2 + (1-0)^2 + (1-1)^2}$$

$$= \sqrt{6} = 2.449$$

Passage vers le pourcentage :

$$sim_{euclidienne}(P,R) = 1 - \frac{d_{euclidienne}}{\max(len(v1), len(v2))}$$

$$= 1 - \frac{2.449}{5} = 0.51$$

➢ **La similarité de Jaccard** L'indice de Jaccard ou coefficient de Jaccard [67] est le rapport entre la cardinalité (la taille) de l'intersection des ensembles considérés et la cardinalité de l'union des ensembles. Il permet d'évaluer la similarité entre les ensembles. Les documents *d*1 et *d*2 sont donc représentés, non pas comme des vecteurs, mais comme des ensembles de termes.

$$sim_{Jaccard}(d1, d2) = \frac{\|d_1 \cap d_2\|}{\|d_1 \cup d_2\|}$$

Exemple :

Soit :

P= {تزداد نسبة الملوحة في البحار}

R= {ترتفع نسبة الملوحة}

Représentons P et R comme des ensembles ensuite faire l'intersection et l'union des ensembles.

Intersection: {'ح', 'و', 'ا', 'ت', 'س', 'ن', ' ', 'ل', 'ف', 'ر', 'ب', 'ة', 'م'}

Union: {'ح', 'و', 'ن', 'ا', 'س', ' ', 'ر', 'ف', 'ل', 'ي', 'ة', 'ت', 'د', 'ع', 'ز', 'ب', 'م'}

Où : la cardinalitéde l'intersection = 13

La cardinalité de l'union = 17

$$sim_{Jaccard}(P, R) = \frac{\text{la cardinalitéde l'intersection}}{\text{La cardinalitéde l'union}} = \frac{13}{17} = 0.764$$

➢ **Le coefficient Overlap** est une mesure de similarité qui mesure le chevauchement entre deux ensembles. Il est lié à l'index de Jaccard et est défini comme la taille de l'intersection divisée par la plus petite de la taille des deux ensembles[12].

$$sim_{overlap}(d1, d2) = \frac{|d_1 \cap d_2|}{\min(|d_1|, |d_2|)}$$

Exemple :

Soit :

P = {تزداد درجة الملوحة}

R = {تزداد درجة ملوحة هذه البحار}

[12] https://en.wikipedia.org/wiki/Overlap_coefficient

Représentons P et R comme des ensembles ensuite faire l'intersection des ensembles :

x: {' ', 'ح', 'ت', 'ز', 'ر', 'م', 'ة', 'د', 'ج', 'ا', 'ل', 'و'}

y: {' ', 'ح', 'ت', 'ب', 'ز', 'ر', 'ذ', 'م', 'ة', 'د', 'ج', 'ا', 'ه', 'ل', 'و'}

Où : card(x) est la cardinalité de l'ensemble x, card(x)=12

card(y) est la cardinalité de l'ensemble y, card(y)= 15

Intersection: {'ز', 'ح', 'و', 'م', ' ', 'ة', 'ج', 'د', 'ر', 'ت', 'ا', 'ل'}

La cardinalité de l'intersection = 12

$$sim_{overlap}(P,R) = \frac{\text{cardinalité de l'intersection}}{\min\left(card(x), card(y)\right)} = \frac{12}{12} = 1$$

C. Les méthodes de similarité syntaxique proposées

❖ **STS (String text similarity) :**

Notre méthode se base sur l'algorithme de sous-séquence commune (LCS) [68] avec une normalisation et de petites modifications pour notre mesure de similarité de chaîne. Nous utilisons trois versions modifiées de LCS et en prenant une somme pondérée. Melamed [69] a normalisé les LCS en divisant la longueur de la plus longue sous-séquence commune par la longueur de la chaîne la plus longue et l'a appelée le plus long sous-séquence commune (LCSR). Mais LCSR ne prend pas en compte la longueur de la chaîne la plus courte qui a parfois un impact significatif sur le score de similarité. Nous avons utilisé pour la normalisation, l'approche d'Islam [39].

Le schéma suivant dans la figure13 résume les étapes de notre approche :

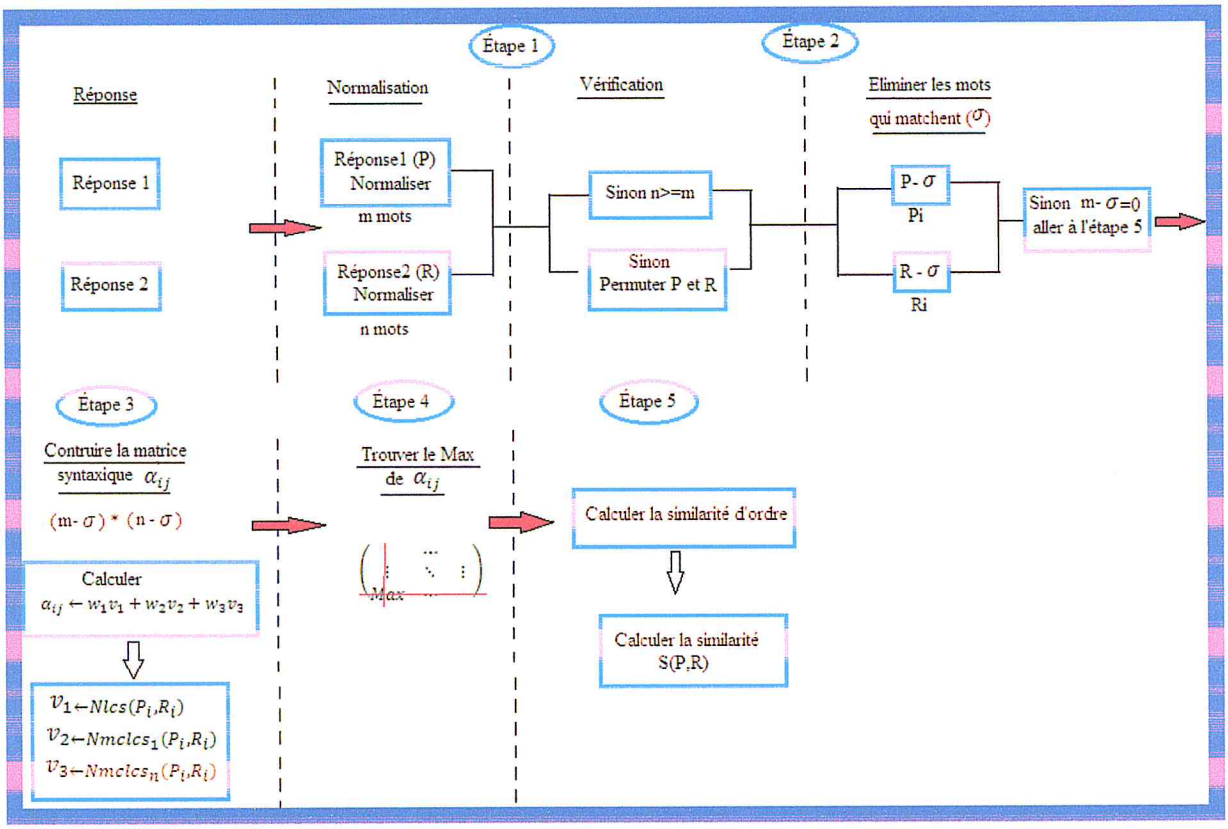

Figure 13: L'approche proposée STS

La première normalisation on utilise la plus longue sous-séquence commune (LCS) afin qu'elle prenne en compte la longueur de la chaîne la plus courte et la plus longue et l'appelle NLCS (normalized longest common subsequence) :

$$U1 = NLCS(ri, sj) = \frac{length(LCS(ri, sj))2}{length(ri) \times length(sj)}$$

La deuxième normalisation nous utilisons la sous-suite commune maximale la plus longue consécutive commençant par le premier caractère et on la normalise ainsi :

$$U2 = NMCLCS1(ri, sj) = \frac{length(MCLCS1(ri, sj))2}{length(ri) \times length(sj)}$$

La troisième normalisation nous utilisons la sous-séquence commune la plus longue consécutive maximale commençant par n'importe quel caractère n et on la normalise ainsi :

$$U3 = NMCLCSn(ri, sj) = \frac{length(MCLCSn(ri, sj))2}{length(ri) \times length(sj)}$$

Nous prenons la somme pondérée de ces valeurs individuelles U1, U2 et U3 pour déterminer le score de similarité des chaînes, où w1, w2, w3 sont des poids et w1 + w2 + w3 = 1. Par conséquent, la similitude des deux chaînes devient :

$$\alpha = w1U1 + w2U2 + w3U3$$

Par exemple :

Si on prend ri = المكنونات et sj = الكائنات, alors :

LCS (ri, sj) = الكنات

MCLCS1(ri, sj) = ال

MCLCSn(ri, sj) = نات

NLCS(ri, sj) = $6^2/(8\times9)$ =0.5

NMCLCS1 = $2^2/(8\times9)$ = 0.05

NMCLCSn(ri, sj) = $3^2/(8\times9)$ = 0.125

La similarité syntaxique devient :

α = w1U1 +w2U2 +w3U3 = 0.33×0.5+0.33×0.05+0.33×0.125 =0.222

- **Similarité d'ordre des mots communs entre les phrases :**

Pour commencer on a considéré un algorithme pour calculer la similarité ordre qui donne l'importance à l'ordre, ceci dit Si les deux phrases ont des mots en commun, on peut mesurer à quel point l'ordre des mots communs est similaire dans les deux textes (si ces mots apparaissent dans le même ordre, ou presque dans le même ordre, ou dans un ordre très différent).

Nous intégrons l'ordre des mots pour tester cette hypothèse et rendre notre méthode plus générique. Nous l'utilisons quand nous considérons l'importance de la similarité syntaxique en

mettant son facteur de poids, wf à moins de 0,5, wf ∈ [0, 0.5]. Nous mettons wf à 0, quand nous voulons ignorer son importance. La valeur de wf devrait être inférieure à 0,5.

Considérons une paire de réponses, respectivement P et R qui ont m et n mots, c'est-à-dire P = p1, p2, ..., pm et R = r1, r2, ..., rn et n ≥ m. Sinon, nous basculons P et R. Nous comptons le nombre de pi (disons δ) pour lequel pi = rj, pour tout p ∈ P et pour tout r ∈ R. Autrement dit, il y a des mots dans P qui correspondent exactement à R, où δ ≤ m.On enlève tous les mots δ de P et les place dans X et R dans Y, dans le même ordre que dans les réponses. Donc, X = {x1, x2, ..., xδ} et Y = {y1, y2, ..., yδ}. Nous remplaçons X en assignant un numéro d'index unique pour chaque mot dans X, de 0 à δ, c'est-à-dire X = {0, 1, ..., δ}. Sur la base de ces numéros d'index uniques pour chaque mot dans X, nous remplaçons également Y où X = Y. Nous proposons une mesure pour mesurer la similarité de l'ordre des mots communs de deux phrases:

$$S0 = 1 - \left| \frac{|X1-Y1|+|X2-Y2|+\cdots+|X\sigma-Y\sigma|}{|X1-X\sigma|+|X2-X\sigma-1|+\cdots+|X\sigma-X1|} \right| \qquad (a)$$

C'est-à-dire que la similarité d'ordre de mots communs est déterminée par la différence normalisée de l'ordre des mots communs. L'équation (b) montre trois cas individuels de (a) :

$$S0=\begin{cases} 1 - \frac{2\sum_{i=1}^{\sigma}|Xi-Yi|}{\sigma^2} & Si\sigma estpair \\ 1 - \frac{2\sum_{i=1}^{\sigma}|Xi-Yi|}{\sigma^2-1} & Si\sigma estimpaire et \sigma > 1 \\ 1 & Si\sigma estimpaire et \sigma = 1 \end{cases} \qquad (b)$$

Par exemple :

P : الدراسة التى تتناول جوانب الطبيعة بما يحدده حياة الكائن و كيفية استخدامه لمكونات البيئة

R : الدراسة التى تتناول الطبيعة و جوانب منها و كيفية إستخدام الكائنات الحية لها

Il existe sept mots dans P qui matchent exactement avec R. On enlève tous les 7 mots de P et R et on les place dans X et Y dans le même ordre dont ils apparaissent dans leurs réponses originales.

X = {'الدراسة', 'التى', 'تتناول', 'جوانب', 'الطبيعة', 'و', 'كيفية'}

Y ={'الدراسة', 'التى', 'تتناول', 'الطبيعة', 'و', 'جوانب', 'كيفية'}

On remplace X par un index unique, pour chaque mot dans X, en commençant par 0 à7

X = {0, 1, 2, 3, 4, 5, 6}

En se basant sur chaque mot avec son index dans X, on fait la même chose pour Y d'où X = Y

Y = {0, 1, 2, 4, 5, 3, 6}

|Xi-Yi|= |0-0|+|1-1|+|2-2|+|3-4|+|4-5|+|5-3|+|6-6)|= 4

$$S0 = 1 - \frac{2\sum_{i=1}^{7}|Xi - Yi|}{7^2 - 1}$$

Notre similarité ordre alors égale à : S0=0.833 et $\sigma estimpairet > 1$

- **Similarité globale des réponses :**

Notre travail est de trouver un score entre 0 et 1 qui indiquera la similarité entre deux réponses P et R. L'idée principale est de trouver, pour chaque mot de la première réponse, l'appariement le plus similaire dans la deuxième réponse.

La méthode comprend 6 étapes :

Étape 1 : Nous utilisons tous les caractères spéciaux, ponctuations et majuscules, On élimine tous ces caractères spéciaux, ponctuations et mots vides. Nous lemmatisons chacun des mots segmentés pour générer les mots. Après le nettoyage, nous supposons que la réponse P = {p1, p2, ..., pm} a m mots et que le texte R = {r1, r2, ..., rn} a n mots et n ≥ m. Sinon, nous basculons P et R.

Étape 2 : Nous comptons le nombre de pi (disons σ) pour lesquels pi = rj, pour tout p ∈ P et Pour tout r ∈ R. Autrement dit, il y a des mots σ dans P qui correspondent exactement à R, où δ ≤ m.On enlève tous les mots δ de P et R.So, P = {p1, p2, ..., pm-σ } et R = {r1, r2, ..., rn-σ }. Si tous les termes correspondent, m-δ = 0, on passe à l'étape 6.

Etape 3 : Nous construisons une matrice de similarité syntaxique (m-σ) × (n- σ)

(M1 = (αij) (m-σ) × (n-σ)) en utilisant le processus suivant: on suppose que tout mot pi

∈ P a τ caractères, c'est-à-dire pi = {c1, c2 ⋯, cτ} et tout mot rj ∈ R a η caractères, c'est-à-dire, rj = {c1, c2 ⋯ , cη} où $\tau \leq \eta$. En d'autres termes, η est la longueur de la plus longue chaine et τ est la longueur de la plus petite. Nous calculons ce qui suit:

U1←NLCS(pi, rj)

U2←NMCLCS1(pi, rj)

U3←NMCLCSn(pi, rj)

αij ←w1U1 +w2U2 +w3U3

αij est une somme pondérée de U1, U2 et U3 où w1, w2, w3 sont des poids et w1 + w2 + w3 = 1. Nous fixons des poids égaux pour nos expériences. Nous mettons αij dans la ligne i et la colonne j de la matrice M1 pour tout i = 1 ⋯ m-σ et j = 1 ⋯ n-σ.

$$M1=\begin{pmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1j} & \dots & \alpha_1(n-\sigma) \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2j} & \dots & \alpha_2(n-\sigma) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha_{i1} & \alpha_{i2} & \dots & \alpha_{ij} & \dots & \alpha_i(n-\sigma) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \alpha\,(m-\sigma)1 & \alpha\,(m-\sigma)2 & \dots & \alpha\,(m-\sigma)j & \dots & \alpha(m-\sigma)(n-\sigma) \end{pmatrix}$$

Etape 4 : Nous construisons une matrice de similarité sémantique (m - σ) × (n - σ) (M2 = (βij) (m-σ) × (n-σ)) en utilisant le processus suivant: Nous mettons βij = semantic Matching (pi, rj) dans la ligne i et la colonne j de la matrice M2 pour tout i = 1 ⋯ m-σ et j = 1 ⋯ n-σ.

Dans notre cas on ne va pas considérer la similarité sémantique, et lui donner une pondération de 0.

$$M2=\begin{pmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1j} & \dots & \beta_1(n-\sigma) \\ \beta_{21} & \beta_{22} & \dots & \beta_{2j} & \dots & \beta_2(n-\sigma) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \beta_{i1} & \beta_{i2} & \dots & \beta_{ij} & \dots & \beta_i(n-\sigma) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \beta\,(m-\sigma)1 & \beta\,(m-\sigma)2 & \dots & \beta\,(m-\sigma)j & \dots & \beta(m-\sigma)(n-\sigma) \end{pmatrix}$$

Etape 5 : Nous construisons une autre matrice conjointe (m - δ) × (n - δ) (M = (γij) (m-δ) × (n-δ)) en utilisant : M ← ψM1 + φM2 (c)

C'est à dire: $\gamma ij = \psi\alpha ij + \varphi\beta ij$ où ψ est le facteur de pondération de la matrice correspondant à la similarité syntaxique qu'on va mettre à 1 et φ est le facteur de pondération de la matrice de similarité sémantique qu'on va mettre à 0, et $\psi + \varphi = 1$. La définition de l'un de ces facteurs à 0 signifie que nous n'incluons pas cette matrice.

$$M=\begin{pmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1j} & \cdots & \gamma_1(n-\sigma) \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2j} & \cdots & \gamma_2(n-\sigma) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma_{i1} & \gamma_{i2} & \cdots & \gamma_{ij} & \cdots & \gamma_i(n-\sigma) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma(m-\sigma)1 & \gamma(m-\sigma)2 & \cdots & \gamma(m-\sigma)j & \cdots & \gamma(m-\sigma)(n-\sigma) \end{pmatrix}$$

Après avoir construit la matrice conjointe, M, nous trouvons l'élément de matrice de valeur maximale, γij. Nous ajoutons cet élément de matrice à une liste ρ et $\rho \leftarrow \rho \cup \gamma ij$ si $\gamma ij \geq 0$. Nous enlevons tous les éléments de la matrice de la ligne i et colonne j de M. Nous répétons la découverte de l'élément de matrice de valeur maximale γij en l'ajoutant à ρ à chaque fois et en supprimant tous les éléments de la matrice de la ligne et de la colonne correspondantes.

Etape 6 : On somme tous les éléments de ρ et on ajoute $\sigma * (1 - wf + wf\ So)$ pour obtenir un score total, où So est un score de similarité d'ordre commun et wf est un poids d'ordre commun qui détermine la similarité des ordres des mots communs. Nous multiplions ce total par la moyenne harmonique réciproque de m et n pour obtenir un score de similarité équilibré entre 0 et 1.

$$S(P,R) = \frac{(\sigma(1-wf+wf*S0)+\sum_{i=1}^{|p|} Pi)*(m+n)}{2mn} \qquad (d)$$

(d) pourrait prendre quatre formes spécifiques: Premièrement, si nous ignorons l'importance de la similarité d'ordre en fixant wf = 0 dans (d), nous obtenons:

$$S(P,R) = \frac{(\sigma+\sum_{i=1}^{|p|} Pi)*(m+n)}{2mn} \qquad (e)$$

Deuxièmement, si nous obtenons une valeur de similarité d'ordre de mots communs (So) à 1, S (P, R) sera indépendante de wf, c'est-à-dire identique à (e). Troisièmement, si nous ignorons l'importance de la similarité syntaxique, nous mettons ψ dans (c) à 0. Quatrièmement, si nous ignorons l'importance de la similarité sémantique, nous mettons φ

dans (c) à 0.

Parcourir un exemple :

Soit :

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

R = « .الدراسة التى تهتم بالبيئة من حيث مكوناتها و كيفية استخدام الكائنات الحية لهذه المكونات»

Cet exemple provient du jeu de données de Gomaa, le sixième couple.

Étape 1 : Après avoir éliminé tous les caractères spéciaux et ponctuations, On supprime tous les mots vides, puis on passe par une normalisation, et On lemmatise en dernier avec une lemmatisation légère, nous obtenons :

P= « الدراسه تتناول جميع جوانب الطبيعه يحدد حياه الكائن كيفيه استخدام مكونات البيئه »

R= « الدراسه تهتم البيئه حيث مكونات كيفيه استخدام الكائنات الحيه هذه المكونات »

Où : n la taille de la phrase P, n = 12 et

m la taille de la phrase R, m = 11.

Étape 2 : On a 5 mots « 'البيئه', 'مكونات', 'استخدام', 'كيفيه', 'الدراسه' » dont P correspond exactement à R, donc nous mettons σ à 5. Nous enlevons (' ,'مكونات', 'استخدام', 'كيفيه', 'الدراسه' البيئه") à la fois de P et de R. Donc, on obtient :

P = {"الكائن', 'حياه', 'يحدد', 'الطبيعه', 'جوانب', 'جميع', 'تتناول' }

R = {'المكونات', 'هذه', 'الحيه', 'الكائنات', 'حيث', 'تهتم'}

Comme m -σ ≠ 0, c'est-à-dire 11-5≠ 0 nous passons à l'étape suivante :

Étape 3 : Nous construisons la matrice de similarité syntaxique M1 de 6× 6 ou chaque élément est calculé comme suit :

Prenons en exemple : le premier élément ayant comme indice (première ligne, première colonne)

$$\alpha 1.1 = \frac{1}{3} * NLCS(\text{تتناول,تهتم}) + \frac{1}{3} * NMCLCS1(\text{تتناول,تهتم}) + \frac{1}{3} * NMCLCSn(\text{تتناول,تهتم})$$

$$\alpha 1.1 = (\frac{1}{3} * 0.041)+(\frac{1}{3} * 0.041)+(\frac{1}{3} * 0.041) = 0.041$$

الكائن	حياه	يحدد	الطبيعه	جوانب	جميع	تتناول	
0	0.041	0	0.023	0	0.041	0.041	تهتم
0	0.333	0.055	0.031	0	0.055	0	حيث
0.75	0.208	0	0.071	0.041	0	0.055	الكائنات
0.133	0.216	0.033	0.228	0.026	0.033	0.022	الحيه
0	0.055	0	0.031	0	0	0	هذه
0.166	0.02	0	0.071	0.041	0.02	0.055	المكونات

Etape 4 : Nous construisons de la même manière une matrice de similarité sémantique M2 de 6 × 6.

Dans notre cas on ne va pas considérer la similarité sémantique, et on lui donne une pondération de 0 dans la matrice combinée.

$$M2=\begin{pmatrix} \beta_{11} & \beta_{12} & \dots & \beta_{1j} & \dots & \beta_1(n-\sigma) \\ \beta_{21} & \beta_{22} & \dots & \beta_{2j} & \dots & \beta_2(n-\sigma) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \beta_{i1} & \beta_{i2} & \dots & \beta_{ij} & \dots & \beta_i(n-\sigma) \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \beta\,(m-\sigma)1 & \beta\,(m-\sigma)2 & \dots & \beta\,(m-\sigma)j & \dots & \beta(m-\sigma)(n-\sigma) \end{pmatrix}$$

Etape 5 : Nous construisons une autre matrice conjointe combinée de dimension (m - δ) × (n - δ) en utilisant la formule : $M \leftarrow \psi M1 + \varphi M2$

Dans notre cas φ est nul car on ne va pas considérer la similarité sémantique, et donc la matrice combinée est la matrice syntaxique elle-même :

الكائن	حياه	يحدد	الطبيعه	جوانب	جميع	تتناول	
0	0.041	0	0.023	0	0.041	0.041	تهتم
0	0.333	0.055	0.031	0	0.055	0	حيث
0.75	0.208	0	0.071	0.041	0	0.055	الكائنات
0.133	0.216	0.033	0.228	0.026	0.033	0.022	الحيه
0	0.055	0	0.031	0	0	0	هذه
0.166	0.02	0	0.071	0.041	0.02	0.055	المكونات

Comme m-$\sigma - longeur(listedesmaximum)$ ≠0 et les éléments de la matrice sont

différents de zéro, alors on cherche la valeur maximale de la matrice qui est : 0.750, puis on supprime la ligne et la colonne de cette dernière, et on ajoute 0.750 à notre liste des maximum qui est vide en premier lieu, notre matrice devient :

حياه	يحدد	الطبيعه	جوانب	جميع	تتناول	
0.041	0	0.023	0	0.041	0.041	تهتم
0.333	0.055	0.031	0	0.055	0	حيث
0.216	0.033	0.228	0.026	0.033	0.022	الحيه
0.055	0	0.031	0	0	0	هذه
0.02	0	0.071	0.041	0.02	0.055	المكونات

Liste des maximums : [0.75]

m-$\sigma - longeur(listedesmaximum)$ ≠0, 10-5-1≠0, 10-5-1≠0 et les éléments de la matrice sont différents de zéro alors on cherche à nouveau la valeur maximale qui est : 0.333, on supprime la ligne et colonne correspondantes à cette valeur, et on obtient :

يحدد	الطبيعه	جوانب	جميع	تتناول	
0	0.023	0	0.041	0.041	تهتم
0.033	0.228	0.026	0.033	0.022	الحيه
0	0.031	0	0	0	هذه
0	0.071	0.041	0.02	0.055	المكونات

Liste des maximums : [0.750, 0.333]

On vérifie à nouveau nos conditions d'arrêtes :

m-$\sigma - longeur(listedesmaximum)$ ≠0,

9-5-2≠0 et les éléments de la matrice sont différents de zéro, alors on cherche notre valeur maximale qui est : 0.228, on supprime la ligne et colonne correspondantes à cette valeur, on aura :

يحدد	جوانب	جميع	تتناول	
0	0	0.041	0.041	تهتم
0	0	0	0	هذه
0	0.041	0.02	0.055	المكونات

Liste des maximums : [0.750,0.333, 0.228]

A ce niveau, on va passer à l'étape suivante car la première condition d'arrêt est vérifiée :

8-5-3=0, c'est-à-dire : m-$\sigma - longeur(listedesmaximum) = 0$.

Etape 6 :

$S(P,R) = \frac{(\sigma+\sum_{i=1}^{|listedesmaximum|}\rho_i)\times(m+n)}{2mn}$ Ou Pi est un élément de la liste des maximums.

$S(P,R) = = ((5 + 1.311) \times (11 + 12))/(2 \times 132) = 0.549$

❖ **TFSS (Term frequency in string similarity)s**

Un aperçu sur le principe de l'approche TFSS est représenté dans la figure 14:

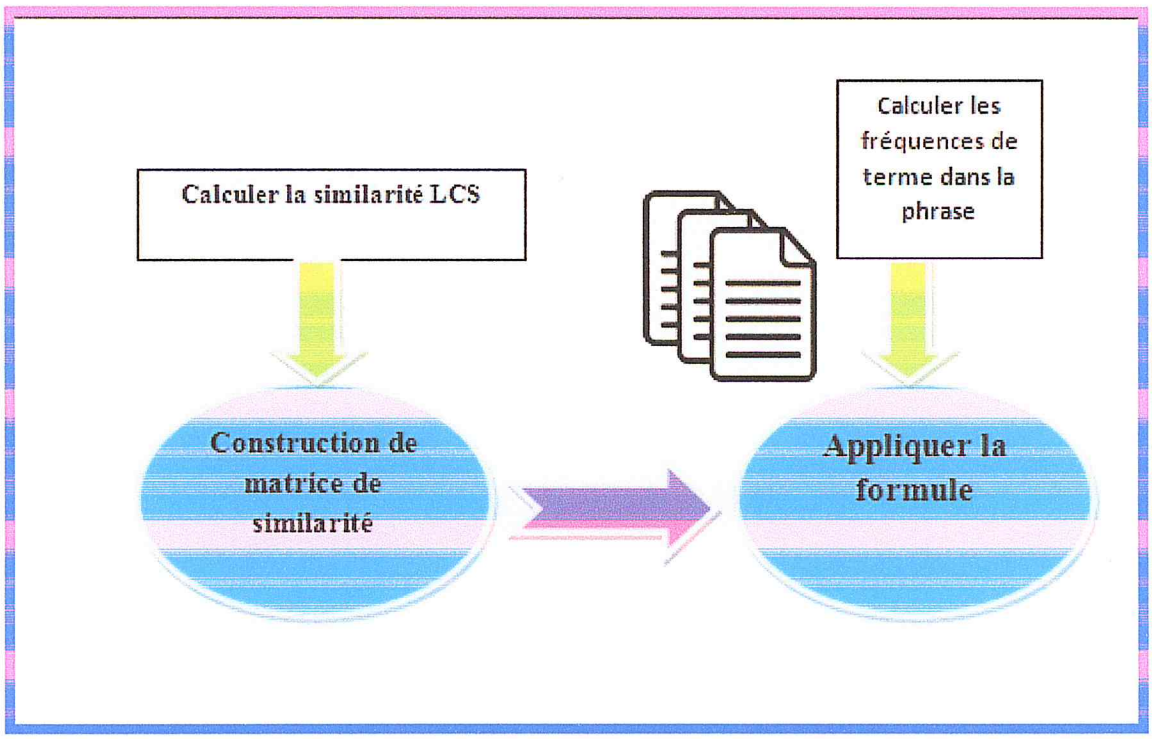

Figure 14: l'approche proposée TFSS

Notre méthode repose sur le modèle BOW; Dans ce modèle, une phrase est représentée comme une collection de mots non ordonnée, sans tenir compte de la grammaire et même de l'ordre des mots. Pour calculer un score de similarité entre les réponses de l'étudiant et celles du modèle, les scores de similarité pour chaque paire de mots doivent être collectés dans une

structure, pour permettre de calculer un score global pour les paires. Parce que chaque mot d'une réponse d'étudiant est comparé à chaque mot de la réponse du modèle, la solution consiste à créer une matrice de similarité de taille N * M, où N est le nombre de mots dans la réponse modèle et M le nombre de mots dans la réponse de l'élève. Dans la matrice, chaque ligne représente un mot dans la réponse modèle, tandis que chaque colonne représente un mot dans la réponse de l'étudiant.

La construction de matrice est faite par le calcul de la similarité entre les mots avec la méthode NLCS proposée dans la méthode précédente, qui est la normalisation de l'algorithme LCS, cette dernière prenne en considération la chaine la plus petite et la plus longue comme c'est mentionné auparavant.

Après avoir construit la matrice de similarité, la similarité entre la réponse de l'étudiant (SA) et la réponse du modèle (MA) est donc déterminée selon l'équation de notation bidirectionnelle suivante, qui a été proposée dans Mihalcea et al. [70].

$$sim(MA,SA) = \frac{1}{2}\left(\frac{\sum_{w\in\{MA\}}(sim(w,SA)*f(w))}{\sum_{w\in\{MA\}}f(w)} + \frac{\sum_{w\in\{SA\}}(sim(w,MA)*f(w))}{\sum_{w\in\{SA\}}f(w)}\right) \dots (1)$$

Où $f(w)$est la fréquence relative d'un mot dans une réponse modèle ou une réponse d'étudiant. $similarité(w,SA)$ est calculée soit par la similarité Max (MaxSim) soit par similarité moyenne (AvgSim). MaxSim est la valeur de similarité la plus élevée entre un mot donné w et tous les mots de la réponse de l'élève. AvgSim est calculé en divisant la somme des valeurs de similarité d'un mot donné par le nombre de mots dans la réponse de l'élève. La même chose vaut $sim(w,MA)$.

Le score de similarité obtenu similarité (MA, SA) a une valeur comprise entre 0 et 1; un score de 1 indique des réponses identiques et 0 n'indique aucune similarité syntaxique.

Parcourir un exemple :

MA = « هو العلم الذى يهتم بالبيئة و الكائنات الحية و الغير حية»

SA = = « الدراسة التي تتناول مكونات البيئة و الكائنات الحية»

Cet exemple provient du jeu de données de Gomaa.

1. Où : n est la taille de la réponse modèle MA, n = 11 et m la taille de la réponse

de l'élève, m = 8

Etape 1 : Nous construisons la matrice de similarité syntaxique de 11× 8 ou chaque élément est calculé comme suit :

Prenons en exemple : le premier élément ayant comme indice (première ligne, deuxième colonne)

$\alpha_{1.2} = NLCS(\text{العلم, الدراسة})$

$\alpha_{1.2} = 0.11428571428571428$

	هو	العلم	الذى	يهتم	بالبيئة	و	الكائنات	الحية	و	الغير	حية
الدراسة	0	0.114	0.142	0	0.183	0	0.160	0.257	0	0.257	0.047
التى	0	0.2	0.562	0.062	0.142	0	0.281	0.2	0	0.2	0
تتناول	0.083	0.133	0.041	0.041	0.095	0	0.083	0.033	0	0.033	0
مكونات	0.083	0.033	0.041	0.0416	0.023	0	0.333	0.033	0	0.033	0
البيئة	0	0.133	0.166	0.041	0.857	0	0.187	0.533	0	0.3	0.222
و	0.5	0	0	0	0	1	0	0	1	0	0
الكائنات	0	0.1	0.125	0.031	0.16	0	1	0.1	0	0.1	0
الحية	0	0.16	0.2	0.05	0.457	0	0.1	1	0	0.36	0.6

Etape2 : Appliquer l'équation (1) au Matrice de similarité syntaxique en utilisant MaxSim, nous obtenons le score de similarité suivant entre la réponse modèle et la réponse de l'étudiant:

$$Sim(MA, SA) = \frac{1}{2}\left(\frac{((1*0.257)+(1*0.562)+(1*0.133)+(1*0.333)+(1*0.857)+(1*1)+(1*1)+(1*1))}{8} + \frac{(1*0.5)+(1*0.2)+(1*0.562)+(1*0.062)*(1*0.857)+(2*1)+(1*1)+(1*1)+(2*1)+(1*0.36)+(1*0.6)}{13}\right)$$

$$= \frac{1}{2}(0.643 + 0.703) = 0.673$$

v. L'évaluation de système :

Pour évaluer le système, deux notions vont êtres introduites, la notion de passage au score, pour avoir les notes finales, et de passer de la similarité donné par les approches cités au dessus qui est dans l'intervalle [0,1], à la note que le programme va accorder, et puis la notion de data set, pour pouvoir tester nos approches sur un ensemble de couple de réponses, et enfin évaluer les approches selon plusieurs méthodes comme l'erreur quadratique et le

facteur de Pearson.

1. Passage au score

Un passage vers le score est exigé, pour avoir la note réelle, la similarité est entre 0 et 1, nous avons effectué ce passage en appliquant le Kmeans et la multiplication (voir la figure 15) :

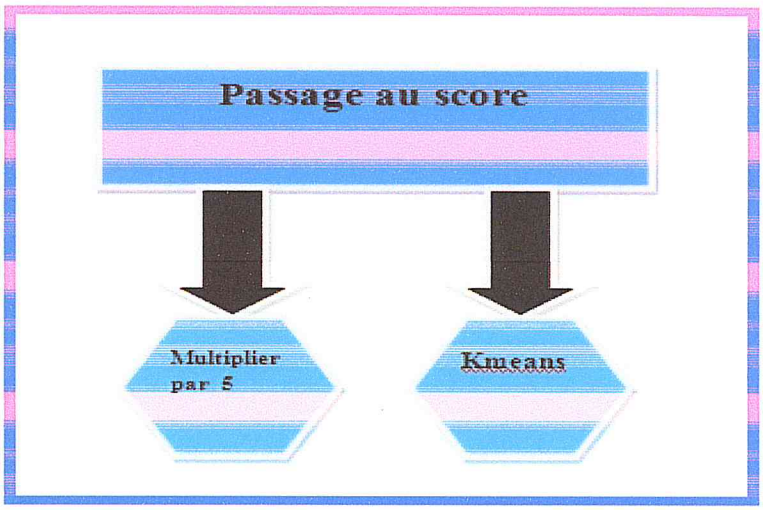

Figure 15: Passage au score

A. Le passage vers le score avec Kmeans

Pour passer les similarités obtenues vers la note finale nous avons utilisé un algorithme non supervisé de clustering non hiérarchique K-means. Il permet de regrouper en clusters distincts les observations du data set. Ainsi les données similaires se retrouveront dans un même cluster. Par ailleurs, une observation ne peut se retrouver que dans un cluster à la fois (exclusivité d'appartenance). Une même observation, ne pourra donc, appartenir à deux clusters différents. Pour traiter les datasets présentés auparavant, où les notes varient de 1 à 5, nous avons choisi 11 classes, k=11 pour permettre un pas de 0.5 dans cet intervalle.

B. Le passage vers le score X5

Ce passage est le plus simple à effectuer en utilisant juste une opération mathématique, notre similarité est obtenue entre 0 et 1, le passage est fait par multiplier cette similarité par 5 pour obtenir une note entre 0 et 5.

2. L'acquisition des data set :

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 [55], [1] 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'autre travaux ayant utilisé ce même dataSet.

A. 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 (à numéroter) 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.

La figue 16 montre un exemple de data set Gomaa en format XML :

Figure 16: Un aperçu XML pour le data set de Gomaa

Comme on peut montrer l'exemple sous forme d'un tableau pour une vue plus claire du data set sur la table 9 :

N°Question	Question	Réponse modèle	Réponses des apprenants	Notes manuelles
1	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
2	اشرحبيئة الإنسان	الإطار الذى يحيا فيه الإنسان مع غيره من الكائنات الحية و يحصل	هى الإطار الذى يحيا الإنسان فيه مع غيره من الكائنات الحية و يحصل منها على مقومات حياته	5

		منها على مقومات حياته	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
3	بيئة الإنسان	الإطار الذى يحيا فيه الإنسان مع غيره من الكائنات الحية و يحصل منها على مقومات حياته	الإطار الذى يحيا فيه الإنسان مع غيره من الكائنات الحية و يحصل منها على مقومات حياته	5
			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 ليحصل على مقومات الحياة	5

Tableau 9 : Un aperçu du dataset GOMAA

B. 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) [57] 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 201 :** data set d'évaluation des travaux en compétition dans le track 1.
- **Le MSRvid 368 SemEval 2017 :** data set 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) [71] 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.Des détails sur les jeux de données sont représentés dans le tableau 10

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

Tableau 10 : Détails sur les jeux de données

Le tableau 11 est un exemple de 3 couples du dataset STS 250 AR :

Datasets STS 250 AR		
Première phrase	**Deuxième phrase**	**Notes**
رجل جالس بمفرده يقرأ على طاولة مستديرة، خارج أحد المتاجر	شخص ما يحمل لوح التزلج ليلا على الرصيف	0.800000
يتسابق بعض الرجال ضمن مسابقة التزلج	تتسابق النساء في سباق الدايتونا 500	1.000000
هناك فتيات يمشين متجاورات	تمشي النساء جنبا إلى جنب	2.600000

Tableau 11 : Appercu du dataset STS 250 AR

3. Corrélation de Pearson et l'erreur quadratique

L'évaluation d'un système implémenté ou d'une approche proposée est indispensable pour estimer le succès d'une recherche. 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 [72] 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) [73]) 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.

Le tableau 12 représente les détails sur la corrélation de pearson :

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 12: Valeurs de 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 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.

vi. L'outil d'évaluation automatique implémenté

Notre travail nous a menés à développer trois outils, l'outil principal est celui d'évaluation automatique de réponses courtes, le deuxième est un outil qui englobe les deux taches des outils NLP, la lemmatisation et la normalisation, les deux processus nécessaires pour une meilleure évaluation automatique en langue arabe, et le troisième c'est un outil d'analyse qui permet d'évaluer les approches en calculant la corrélation de Pearson et l'erreur quadratique.

- **Outil d'évaluation automatique :**

Cet outil permet de calculer la similarité entre les réponses en utilisant une combinaison entre les approches syntaxiques, en attribuant un poids aux approches sélectionnées, à condition que la somme des coefficients ne dépassent pas 1 pour finalement avoir la similarité entre la réponse modèle et la réponse de l'étudiant.

L'outil permet aussi de passer au score, c'est-à-dire de passer de la valeur de similarité à une notation automatique qui peut être donné sur une note choisie par l'utilisateur, de 1 à 5 en utilisant l'algorithme Kmeans.

L'outil aussi exige de choisir la tâche NLP de lemmatisation avec laquelle on veut mesurer notre similarité, une légère lemmatisation, lemmatisation lourde, comme on peut ignorer la lemmatisation.

La figue 17 représente l'outil, en choisissant nos deux approches développées, nous avons attribué un poids de 0.5 pour chacune, en utilisant une lemmatisation lourde, et une notation sur 5.

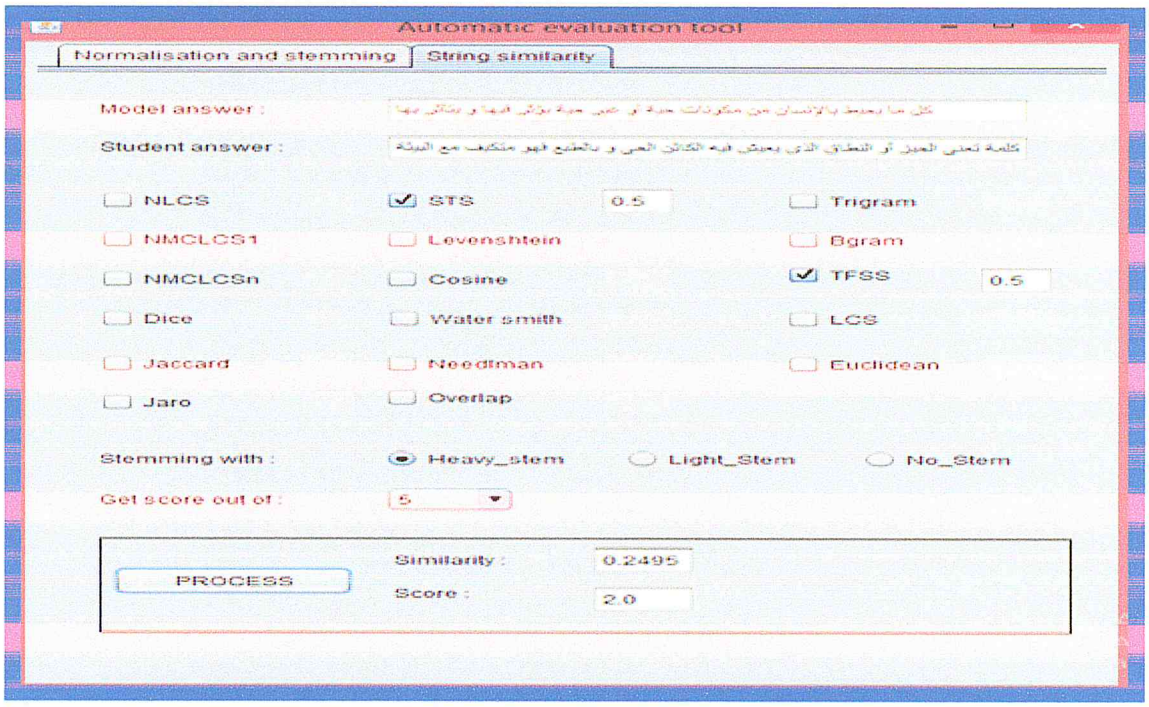

Figure 17: L'outil d'évaluation automatique

- **L'outil de normalisation et lemmatisation :**

Cet outil permet à l'utilisateur d'exécuter les deux tâches de NLP, lemmatisation et normalisation, en sélectionnant la tâche de lemmatisation il doit choisir entre un stem lourd (Khoja) et stem léger (Stanford coreNLP), et voir le résultat dans la zone Output.

Si l'utilisateur sélectionne la tâche de normalisation, il doit ainsi choisir quelle normalisation à exécuter, normalisation d'une des lettres cités, suppression des numéros, ou de lettre latin, et enfin il peut faire une normalisation complète qui englobe toute la liste des choix, l'utilisateur verra l'impact de la normalisation dans la zone Output de l'outil.

Les deux figures 18 et 19 en dessous représentent les deux outils des processus :

- **Normalisation :**

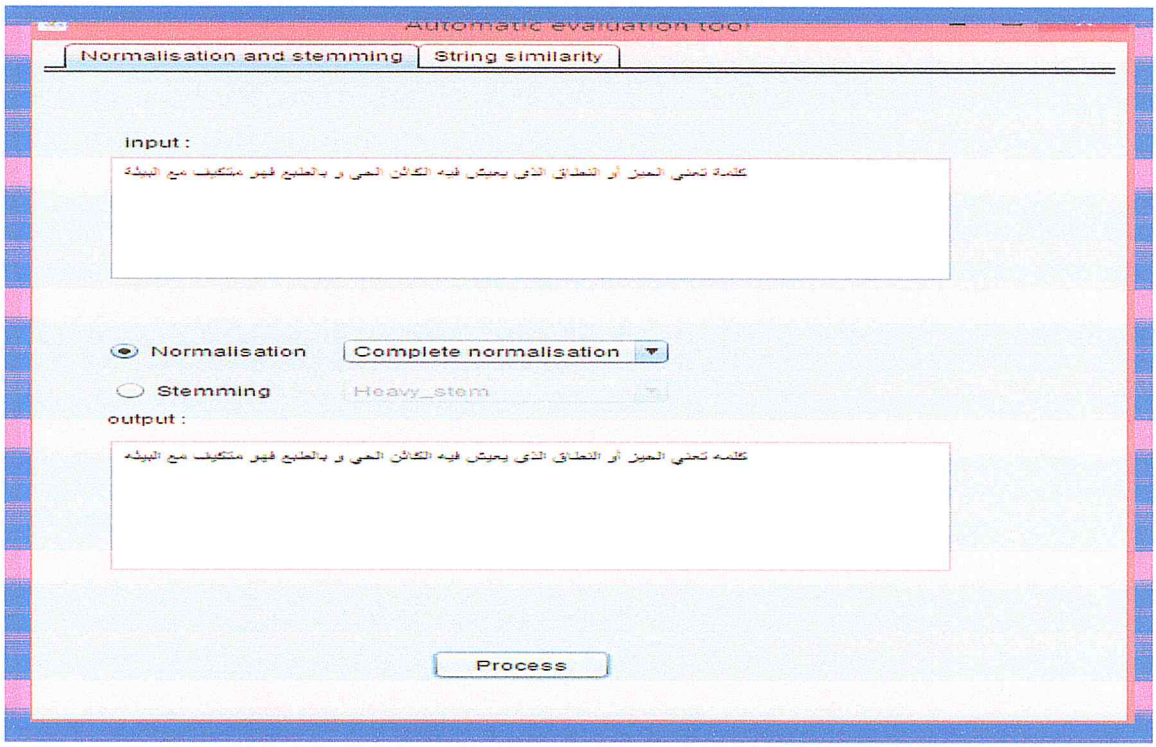

Figure 18: Outil de normalisation

- **Lemmatisation :**

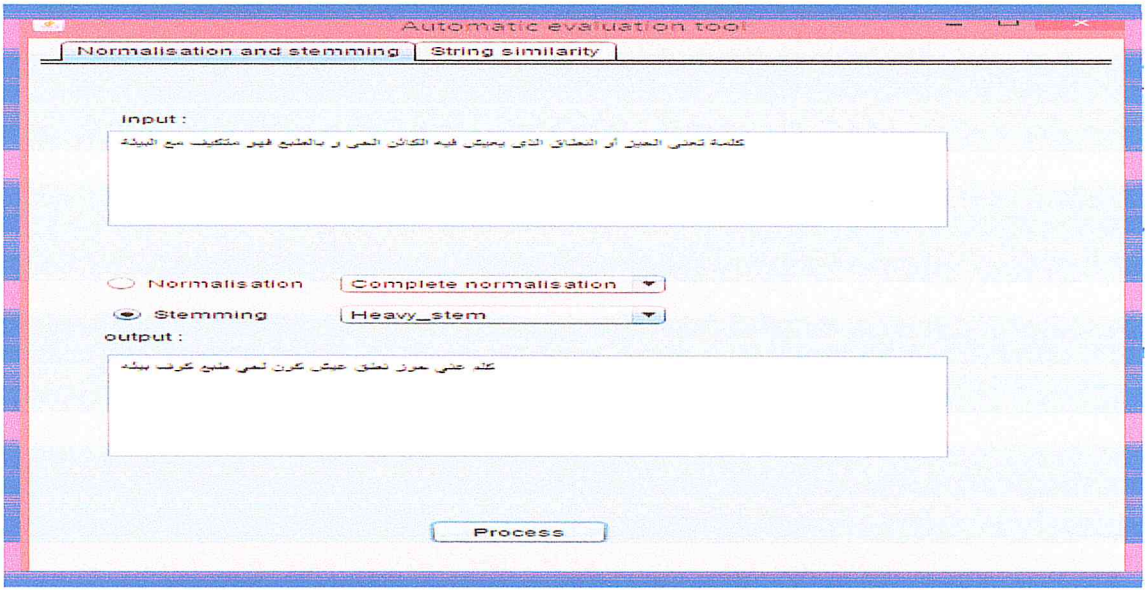

Figure 19: Outil de lemmatisation

- **L'outil d'analyse d'approche :**

Cet outil permet d'évaluer les approches syntaxiques développées et proposées en calculant la corrélation de Pearson et l'erreur quadratique, il lui faut comme paramètres un data set triées en 3 fichiers texte (.txt), fichier des réponses modèles, fichier des réponses des étudiants et le fichier des notes manuelles (voir la figure 20).

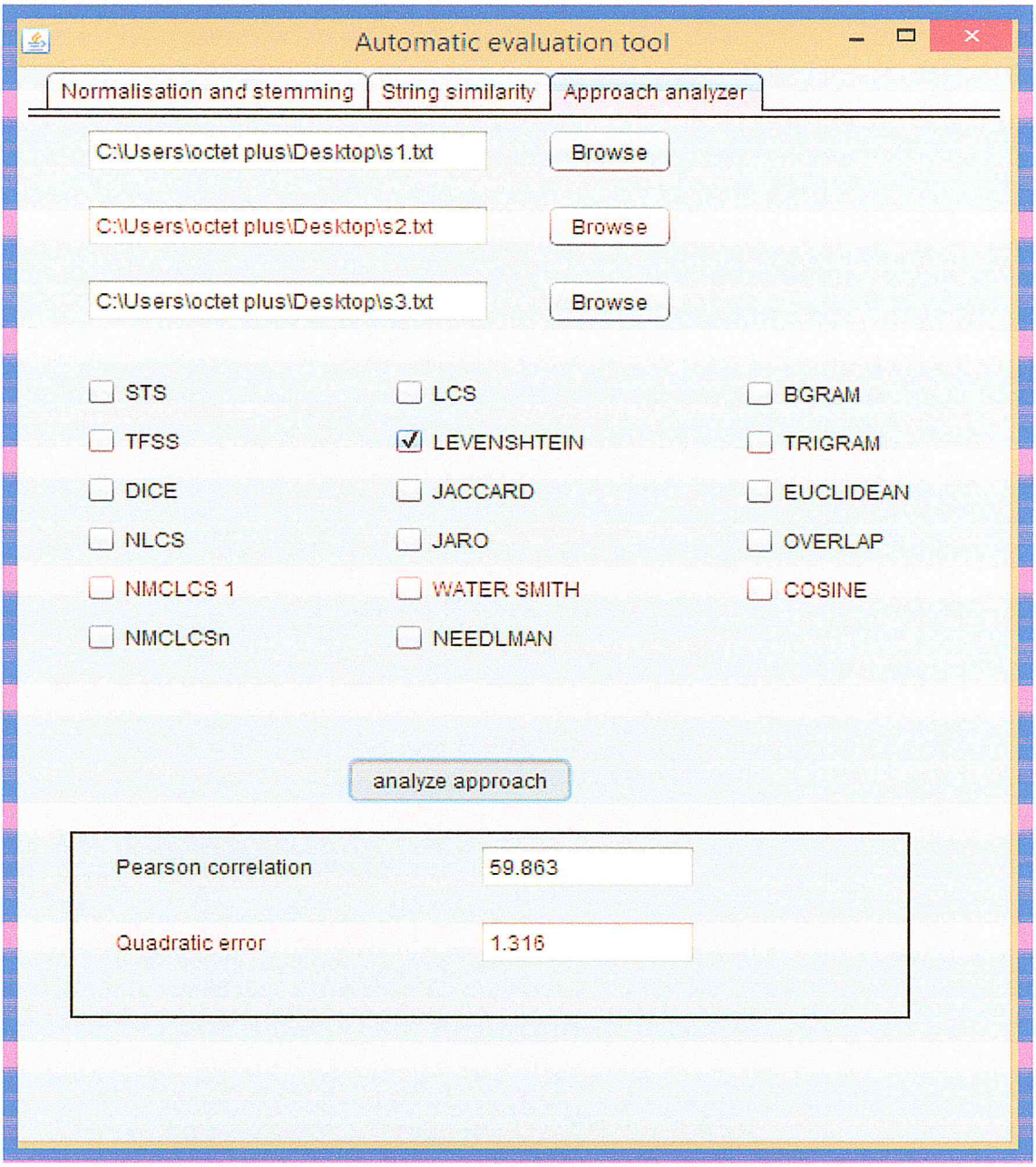

Figure 20: Outil d'analyse d'approche

Conclusion :

Nous avons présentés dans ce chapitre nos approches syntaxiques, ainsi que les différentes étapes de développement de notre système d'évaluation automatique, on a terminé notre chapitre par présenter les méthodes avec lesquelles on a pu évaluer notre système, en passant au score et puis en estimant la valeur de corrélation de Pearson et l'erreur quadratique, dans le prochain chapitre nous allons présenter les résultats, ainsi que les interprétations possibles à nos approches de similarité développées.

IV. Résultats expérimentaux et évaluation

Nous présentons dans ce chapitre les résultats obtenus, avec leur interprétations, nous avons testés nos approches en domaine d'enseignement et d'apprentissage avec la langue arabe d'origine comme le cas dans le data set de Gomaa, et pour un domaine différent de l'enseignement, et traduit en arabe comme le data set de la compétition de SEMeval, nous avons considéré les 3 cas, un stem lourd, un stem léger, et enfin sans appliquer aucun stem

i. Expérimentation et évaluation

Nous allons d'abord commencer par présenter nos résultats des approches syntaxiques, ensuite les résultats de nos approches proposées, et puis procéder à des combinaisons à fin de trouver la meilleure de ces dernières et pouvoir faire une hybridation avec les méthodes sémantiques implémentés par les autres binômes Yasmine et Asma, ainsi que Adel et Hamza ces deux travaux qui se déroulent en parallèle avec le nôtre. Le classifieur Kmeans avec K=11 a donné en général meilleur résultat c'est pour cette raison que nous ne présentons que le calcul de score en utilisant Kmeans.

1. Les approches syntaxiques :

Nous allons commencer par tester toutes les similarités syntaxiques implémentées ainsi que nos approches proposées chacune indépendamment de l'autre, pour pouvoir extraire la meilleure approche, celle qui a donné une meilleure corrélation de Pearson ou une meilleure erreur quadratique.

A. LCS :

Dans la table 13, nous retrouvons les résultats sur les 3 data sets de l'application de l'algorithme LCS. Cette approche a donné une corrélation de Pearson importante dans les trois data set par rapport aux autres approches syntaxiques connues dans la littérature. Le meilleur résultat est obtenu pour le data set de Gomaa, et en utilisant un stem lourd. On explique ça par la capacité de LCS de détecter les mots similaires en prenant la chaine commune la plus longue non consécutive. Nous avons obtenu une valeur meilleure que celle qu'à obtenu Gomaa avec la même approche, car gomaa a modifié le vrai LCS qui considère la chaine la plus longue non consécutive en utilisant une chaine commune consécutive, ce qui a détérioré son résultat à 42 % alors que nous avons obtenu un 67,51 %.

	Approche LCS					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	64.265	1.611	67.511	1.533	63.96	1.453
Ar 250	62.527	1.216	61.364	1.280	61.247	1.248
Ar 368	49.295	1.450	59.182	1.359	52.002	1.381

Tableau 13: Résultats de l'application de LCS sur les Datase

B. Cosine :

Dans la table 14 nous retrouvons les résultats de l'application du cosine sur les 3 datasets. La corrélation de Pearson pour la similarité de Cosine a atteint 77,28 % à Gomaa, un résultat meilleur que le LCS, par rapport au trois data set, et c'est aussi en utilisant un stem lourd, le stem lourd aide à unifier les mots de même racine et vu que la similarité cosine passe par une représentation vectorielle, ceci rapproche les mots de même famille et donc il auront même représentation dans le vecteur, et ainsi le résultat est optimisé.

CP : Corrélation de pearson
EQ : Erreur quadratique

	Approche Cosine					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	70.444	1.399	77.284	1.096	74.907	1.3
Ar 250	61.976	1.425	69.514	1.209	63.460	1.482
Ar 368	57.047	1.35	76.009	1.015	57.225	1.393

Tableau 14: Approche Cosine

C. Bgram :

Dans la table 15 nous retrouvons les résultats de l'application du Bgram sur les 3 data sets. Les similarités de Ngram, peuvent être basées sur les caractères comme elles peuvent être basés sur termes, celles qui sont basées sur caractères ont donné un résultat meilleur que celles qui sont basées sur termes, on explique ceci est qu'il est rare de trouver trois ou quatre mots consécutifs ayant la même forme à la fois dans la réponse de l'élève et dans la réponse du modèle, et donc la similarité Ngram à base terme ne pourrait être meilleure que celle basée sur les caractères.

	Approche Bgram					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	35.930	2.138	42.290	1.933	42.625	2.017
Ar 250	45.166	2.052	42.21	2.01	41.493	2.119
Ar 368	48.211	1.827	53.352	1.751	44.717	1.906

Tableau 15: Approche Bgram

D. Trigram :

Dans la table 16 nous retrouvons les résultats de l'application du Trigram sur les 3 data sets. Les similarités de trigram, peuvent être basées sur les caractères comme elles peuvent être basés sur termes, celles qui sont basées sur caractères ont donné un résultat meilleur que celles qui sont basées sur termes, on explique ceci est qu'il est rare de trouver trois ou quatre mots consécutifs ayant la même forme à la fois dans la réponse de l'élève et dans la réponse du modèle, et donc la similarité trigram à base terme ne pourrait être meilleure que celle basée sur les caractères.

	Approche Trigram					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	48.815	2.3	52.453	2.193	49.811	2.335
Ar 250	32.110	2.357	38.698	2.327	29.676	2.457
Ar 368	35.007	2.109	40.357	2.061	30.903	2.170

Tableau 16: Approche Trigram

E. Dice :

Dans la table 17 nous retrouvons les résultats de l'application du Dice sur les 3 data sets. Au data set Gomaa, Dice a été notre meilleure approche syntaxique avec une corrélation de 82.62 % en utilisant un stem Léger, et même pour le data set Ar368 avec une corrélation de pearson 78.32 % , il a ainsi donné un bon résultat au data 250 mais il n'était pas le meilleur, notre approche proposée de STS l'a dépassé, on explique ceci par le fonctionnement de Dice qui se base sur les mots communs, le cas des réponses des étudiants avec réponses de l'enseignant ou le vocabulaire est presque le même d'où vient cette valeur importante de Pearson.

	Approche Dice					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	81.930	0.998	80.676	1.127	82.627	1.005
Ar 250	68.511	1.128	70.654	1.163	70.083	1.103
Ar 368	73.454	1.161	78.329	1.008	70.153	1.229

Tableau 17: Approche Dice

F. Jaro :

Dans la table 18 nous retrouvons les résultats de l'application du Jaro sur les 3 data sets, cet algorithme est conçu pour la détection des doublons, l'unification des mots par un stem lourd, va considérer les doublons et donc une meilleure détection de similarité qui a permet d'avoir un meilleur résultat en stem lourd par rapport à un stem léger, et donc jaro a un bon impact sur l'arabe.

	Approche Jaro					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	66.036	1.575	70.326	1.413	66.311	1.531
Ar 250	59.414	1.227	61.432	1.254	62.118	1.194
Ar 368	46.659	1.593	54.124	1.426	49.204	1.441

Tableau 18: Approche Jaro

G. Jaccard :

Dans la table 19 nous retrouvons les résultats de l'application du Jaccard sur les 3 data sets, la similarité de Jaccard permet de mesurer la similarité entre les ensembles, en s'appuyant sur les éléments en commun de ces derniers, cette similarité a donné une meilleure corrélation de Pearson dans le data set de Gomaa en utilisant un stem lourd, ce qui nous prenne à la même interprétation des algorithmes précédents, le vocabulaire commun entre les réponses modèles et réponses d'étudiants.

	Approche Jaccard					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	70.673	1.209	74.653	1.114	70.625	1.210
Ar 250	55.603	1.338	63.779	1.25	57.612	1.269
Ar 368	66.900	1.231	68.193	1.195	66.503	1.201

Tableau 19: Approche Jaccard

H. Euclidienne :

Dans la table 20 nous retrouvons les résultats de l'application du Euclidienne sur les 3 data sets, cette similarité comme la similarité cosine passe par une représentation vectorielle pour pouvoir appliquer sa loi qui est applicable sur des chiffres et non pas des mots, mais sa loi de soustraction, élimine la similarité des mots communs ce qui explique le facteur de Pearson faible de 53 % à Gomaa mais plus élevé là ou les mots en communs sont rare comme le data set 368.

	Approche Euclidienne					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	42.654	1.582	53.102	1.443	41.862	1.566
Ar 250	32.320	1.579	54.863	1.371	35.735	1.48
Ar 368	57.406	1.497	72.390	1.338	60.243	1.362

Tableau 20: Approche Euclidienne

I. Damerau-levenshtein :

Dans la table 21 nous retrouvons les résultats de l'application du Damerau-levenshtein sur les 3 data sets, La motivation originale pour établir cette similarité était de mesurer la distance entre un mot correct et un mot comportant une faute d'orthographe humaine afin d'améliorer des applications telles que les vérificateurs d'orthographe, Damerau levenshtein, sur les 3 data sets n'a pas donnée une corrélation importante par rapport aux autres approches, dont nous avons besoin dans l'évaluation automatique et que les fautes d'orthographe seront moins considérables pour une meilleure notation dans un data set comme Gomaa.

	Approche Damerau-levenshtein					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	64.482	1.663	67.615	1.528	63.825	1.581
Ar 250	61.856	1.364	59.117	1.308	59.862	1.315
Ar 368	49.015	1.416	61.913	1.241	51.950	1.390

Tableau 21: Approche Damerau-levenshtein

CP : Corrélation de pearson
EQ : Erreur quadratique

J. Water-smith :

Dans la table 22 nous retrouvons les résultats de l'application du Water-smith sur les 3 data sets, les résultats de water-smith n'étaient pas optimaux, cet algorithme est par exemple utilisé pour aligner des séquences de nucléotides ou de protéines, son efficacité est moins importante dans un domaine d'évaluation automatique pour la langue arabe.

	Approche Water-smith					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	36.66	1.962	30.185	1.951	37.591	1.879
Ar 250	29.743	1.818	44.875	1.937	37.751	1.715
Ar 368	19.274	2.073	28.571	2.211	14.304	2.230

Tableau 22: Approche Water-smith

K. Overlap :

Dans la table 23 nous retrouvons les résultats de l'application de l'Overlap sur les 3 data sets, son résultat n'était pas optimal, cet algorithme qui considère la chaine commune entre les deux réponses et divise le résultat sur la plus petite, n'a pas une grande importance dans l'évaluation automatique pour la langue arabe.

	Approche Overlap					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	39.510	1.855	47.30	1.834	45.180	1.839
Ar 250	46.758	1.767	56.703	1.715	47.984	1.774
Ar 368	64.440	1.769	64.816	1.642	62.235	1.866

Tableau 23: Approche Overlap

L. Needlman :

Dans la table 24 nous retrouvons les résultats de l'application du Needlman sur les 3 data sets, l'utilisation de Needlman est la même que Water-smith, qui sont couramment utilisés en bioinformatique pour aligner des séquences de protéines ou de nucléotides, et moins utilisés dans notre domaine d'évaluation automatique, on remarque que leur corrélation de pearson sont petites et leur impact n'est pas bon pour l'évaluation automatique en langue arabe.

	Approche Needlman					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	65.210	1.626	67.031	1.447	64.678	1.456
Ar 250	61.743	1.228	60.579	1.255	61.843	1.256
Ar 368	48.970	1.4	59.678	1.297	53.455	1.4

Tableau 24: Approche Needlman

2. Nos approches syntaxiques proposées

A. Approche STS

WF : est le coefficient accordé à la similarité d'ordre mentionnée dans le chapitre 3 pour notre approche STS.

Dans la table 25 nous retrouvons les résultats de l'application du STS sur les 3 data sets. Notre approche String text similarity, nous a amélioré le résultat au niveau de data set Gomaa, et elle était l'approche qui a donné le meilleur résultat au niveau du data Ar250 avec une corrélation de Pearson 71.76 % en donnant un poids de 0.1 pour la similarité d'ordre, elle a prouvé une amélioration claire par rapport au LCS classique qui nous a donné 61 % pour le même data set, ceci revient à donner une considération pour la chaine la plus petite aussi et à la similarité d'ordre.

➢ **STS avec un poids de 0.1 pour similarité d'ordre :**

	Approche STS wf=0.1					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	72.956	1.420	77.523	1.112	74.806	1.407
Ar 250	65.944	1.265	71.76	1.145	69.614	1.144
Ar 368	66.097	1.187	76.305	1.019	66.575	1.174

Tableau 25: Approche STS wf=0.1

B. Approche TFSS (Term frequency in string similarity)

Dans la table 26 nous retrouvons les résultats de l'application du STS sur les 3 data sets., la corrélation de Pearson de notre deuxième approche proposée est bien estimé mais ce qui est à mentionner c'est l'amélioration de l'erreur quadratique avec 0.98 %, la meilleure parmi toutes les autres approches implémentées.

	Approche basée sur la fréquence de mot					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	71.597	1.289	76.509	1.132	76.037	1.094
Ar 250	67.708	1.177	68.572	1.208	69.910	1.103
Ar 368	70.668	1.163	77.709	0.989	71.462	1.147

Tableau 26: Approche Basée sur la fréquence de mot (TFSS)

Conclusion des résultats des approches syntaxiques :

Les meilleurs résultats :

Le data set Gomaa : Dice a donné la meilleure corrélation de Pearson avec un pourcentage de 82.62 % en utilisant un stem léger.

Le data set Ar250 : Notre approche STS a donné la meilleure corrélation de Pearson avec un pourcentage de 71.76 % en utilisant un stem Lourd

Le data set Ar368 : Dice a donné la meilleure corrélation de Pearson avec un pourcentage de 78.32 % en utilisant un stem lourd et TFSS meilleure erreur quadratique de 0.98 %.

- Les recherches n'ont pas données une importance à la similarité Dice qui a pourtant donnée un résultat assez satisfaisant surtout pour le domaine d'évaluation automatique qui est le contexte de notre travail et pour un data set natif arabe. Cette similarité Dice qui se base sur les mots communs ce qui explique le bon résultat puisque qu'un étudiant a une grande chance de reprendre les mêmes mots de son cours.
- Le LCS classique qui utilise la chaine commune non consécutive a donné une corrélation importante alors que dans les recherches de Gomaa a donné un résultat moyen vu qu'il a utilisé la chaine commune consécutive.
- Un stem léger est apprécié pour le domaine d'évaluation automatique et pour un data set natif arabe pour pouvoir enlever les pronoms possessifs, le pluriel …etc. dans la langue arabe, il est apprécié pour arriver à détecter la similitude entre les mots.
- Pour un data set traduit en langue arabe et un data set différent de domaine d'apprentissage, le stem lourd est mieux pour sa capacité à extraire la racine du mot pour mieux comparer et arriver à un résultat satisfaisant en similarité syntaxique.

3. Les combinaisons entre approches syntaxiques

Nous allons entamer des combinaisons entre nos approches syntaxiques afin d'apporter des améliorations pour les trois data set au niveau de la corrélation Pearson et l'erreur quadratique. Pour se faire on a appliqué des combinaisons avec une similarité moyenne, ainsi que similarité maximale :

Similarité moyenne : Cette méthode de combinaison non supervisée calcule simplement une moyenne de K scores de similarité par paires:

$$S_{cmb} = \frac{1}{k}\sum_{k=1}^{k} S_k \leftrightarrow s_{ij}^{cmb} = \frac{1}{k}\sum_{k=1}^{k} s_{ij}^{k}$$

Similarité maximale : Cette méthode de combinaison non supervisée calcule un maximum de K similarités par paires:

$$S_{ij}^{cmb} = max(S_{ij}^{1}, \dots\dots, S_{ij}^{k})$$

A. Combinaison des algorithmes ayant une corrélation de Pearson supérieur à 65%

Dans la table 27 nous retrouvons les résultats de l'application de la combinaison des algorithmes ayant une corrélation de Pearson supérieur à 65% sur les 3 data set, le meilleur résultat était 78,49 % ce qui n'a pas dépassé la meilleure corrélation déjà obtenue sans combinaison.

		Combinaison >65%					
		Sans Stem		Khoja stem		Light stem	
	Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	Moyenne	77.328	1.324	78.497	1.353	78.198	1.393
	Max	72.452	1.217	76.676	1.13	73.339	1.24
Ar 250	Moyenne	68.677	1.169	70.806	1.116	71.036	1.203
	Max	68.490	1.151	68.483	1.214	69.071	1.203
Ar 368	Moyenne	72.705	1.106	78.112	0.943	77.409	0.961
	Max	68.666	1.244	75.709	1.022	78.319	1.015

Tableau 27: Combinaison des algorithmes ayant une CP supérieur à 65%

CP : Corrélation de pearson
EQ : Erreur quadratique

B. Combinaison des algorithmes ayant une corrélation de Pearson supérieur à 75%

Dans la table 28 nous retrouvons les résultats de l'application de la combinaison des algorithmes ayant une corrélation de Pearson supérieur à 75% sur les 3 datasets, certaines cases sont vides, car au dataset 250 nous n'avons pas un algorithme qui a une corrélation supérieure à 75 % , on a pu obtenir une corrélation importante avec cette combinaison sans utiliser le stem celle de 81.93 % la meilleure dans le dataset gomaa, si on ne considère pas le cas de light stem.

		combinaison >75%					
		Sans Stem (Dice)		Khoja stem		Light stem	
	Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	Moyenne	81.93	0.998	79.246	1.0931	79.459	1.193
	Max	81.93	0.998	77.157	1.113	77.353	1.151
Ar368	Moyenne			78.15	0.966		
	Max			78.019	1.017		

Tableau 28: Combinaison des algorithmes ayant une CP supérieur à 75%

C. Combinaison STS, Dice

Dans la table 29 nous retrouvons les résultats de l'application de la combinaison STS et Dice sur les 3 datasets.la combinaison entre les meilleurs algorithmes, La combinaison a amélioré la corrélation de Pearson à 78.71 % après qu'elle était 78.32 % au niveau de data set AR368.

CP : Corrélation de pearson
EQ : Erreur quadratique

		Combinaison STS, Dice					
		Sans Stem		Khoja stem		Light stem	
	Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	Moyenne	77.889	1.311	80.251	1.087	78.926	1.305
	Max	80.982	1.205	80.552	1,06	82.272	0.999
Ar 250	Moyenne	67.864	1.166	71.618	1,111	69.659	1.151
	Max	67.785	1,137	71.616	1.148	70.352	1.094
Ar 368	Moyenne	70.484	1.113	78.712	0.986	70.558	1.163
	Max	72.862	1.206	77.669	1.001	70.24	1.242

Tableau 29: Combinaison STS, Dice

D. Combinaison STS, Dice, Cosine

Dans la table 30 nous retrouvons les résultats de l'application de la combinaison STS, Dice, Cosine , la combinaison Max a donné un résultat important de 82.32 %, c'est expliqué par la combinaison des algorithmes ayant des valeurs de Pearson importantes.

		Combinaison STS, Dice,Cosine					
		Sans Stem		Khoja stem		Light stem	
	Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ(%)
Gomaa	Moyenne	77.176	1.217	79.357	1.119	79.032	1.192
	Max	80.719	1.196	80.293	1.058	82.325	0.991
Ar 250	Moyenne	66.884	1.2	71,403	1.124	69.520	1.182
	Max	67.654	1.143	70.417	1.183	70.146	1.098
Ar 368	Moyenne	66.237	1.209	77.5	0.975	66.178	1.186
	Max	72.862	1.206	77.773	1.005	70.192	1.226

Tableau 30: Combinaison STS, Dice, Cosine

CP : Corrélation de pearson
EQ : Erreur quadratique

E. Combinaison STS, Dice, Jaccard, Jaro, Cosine

Dans la table 31 nous retrouvons les résultats de l'application de la combinaison STS, Dice, Jaccard, Jaro, Cosine, cette combinaison a donnée un résultat optimal de 80,57 % mais pas pour toucher les meilleurs résultats.

		Combinaison STS, Dice, Jaccard, Jaro , cosine					
		Sans Stem		Khoja stem		Light stem	
	Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ(%)
Gomma	Moyenne	78.788	1.168	80.576	1.129	78.347	1.424
	Max	72.023	1.250	76.251	1.173	72.86	1.263
Ar 250	Moyenne	65.846	1.193	70.951	1.122	70.2	1.112
	Max	60.462	1.212	65.544	1.251	62.597	1.212
Ar 368	Moyenne	68.407	1.123	77.230	1.037	67.368	1.163
	Max	51.844	1.426	57.55	1.404	51.867	1.405

Tableau 31: Combinaison STS, Dice, Jaccard, Jaro , cosine

F. Combinaison TFSS, Dice

Dans la table 32 nous retrouvons les résultats de l'application de la combinaison TFSS, Dice. Notre approche TFSS en la combinant avec Dice a pu donner un meilleur résultat au niveau de l'erreur quadratique 0.95% pour le data set 368.

	Combinaison Similarité TFSS, Dice					
	Sans Stem		Khoja stem		Light stem	
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ(%)
Gomaa	79.138	1.064	80.523	1.053	80.834	1.109
Ar 250	68.526	1.12	70.874	1.134	71.146	1.061
Ar 368	73.407	1.145	78.654	0.956	71.033	1.211

Tableau 32: Combinaison Similarité TFSS, Dice

Conclusion des résultats de combinaison des approches syntaxiques :

Les meilleurs résultats :

On constate que les résultats obtenus pour les combinaisons syntaxiques en utilisant les 3 approches de stem, stem lourd, léger et sans stem n'ont pas apportées une amélioration aux résultats des similarités syntaxiques testées indépendamment sauf pour certains cas et en considérant l'erreur quadratique :

Le data set Gomaa : Aucune amélioration

Le data set AR250 : La combinaison de tous les algorithmes a pu améliorer l'erreur quadratique de 1.10 % à 1.08 %.

Le data set AR368 : La combinaison Dice + STS a améliorée la corrélation de Pearson de 78.32 % à 78.71 %

La combinaison Dice + FTSS a améliorée l'erreur quadratique de 0.97 % à 0.95 %.

4. Hybridation des similarités syntaxiques et sémantiques :

Pour avoir des résultats meilleurs, et faire l'hybridation avec les approches sémantiques, on a tout d'abord combiné nos approches syntaxiques avec les approches sémantiques en combinant les matrices de similarité, c'est-à-dire en utilisant notre approche STS qui passe par une matrice de similarité, on a combiné notre matrice syntaxique avec la matrice sémantique du premier binôme (Asma et Yasmine) puis avec la matrice sémantique du deuxième binôme (Adel et Hamza), en donnant un poids de **0.5** pour chaque matrice, puis on a procédé à une autre combinaison différente de la combinaison des matrices de similarité, nous avons combinés notre meilleure approche syntaxique avec la meilleure approche de chacun des deux binômes en utilisant la combinaison moyenne, et en dernier, nous avons combinés les trois STS implémentés par nous et les deux autres binômes, en passant par la combinaison des trois matrices , et puis les 3 meilleures approches de chaque binôme en utilisant la combinaison moyenne.

A. Combinaison entre STS sémantique et STS syntaxique

Dans la table 33 nous retrouvons les résultats de l'application de la combinaison de l'approche sémantique développée par Asma et Yasmine et notre approche syntaxique STS, nous avons combinés nos matrices de similarité et en donnant un poids de **0.5** pour chaque matrice, et un poids de 0.1 pour la similarité d'ordre nous avons obtenus une corrélation de 76.61 % à Gomaa le data set principal de notre travail.

		Combinaison (Sem-Syntaxique)			
		CNN_Corpora		khaleej	
	WF	wf=0.0	wf=0.1	wf=0.0	wf=0.1
Gomaa	EQ(%)	1.18	1.15	1.19	1.15
	CP(%)	76.60	76.61	76.50	76.41
STS 250	EQ(%)	1.18	1.18	1.19	1.18
	CP(%)	70.81	70.92	71.06	71.23
STS 368	EQ(%)	0.98	0.98	0.99	1.00
	CP(%)	77.86	77.29	77.53	76.74

Tableau 33: Combinaison (Sem-Synt)

B. Combinaison entre STS syntaxique et STS sémantique en utilisant SkipGram :

Dans la table 34 nous retrouvons les résultats de l'application de la combinaison d'approche sémantique qui utilise les word embedding développée par Adel et Hamza et notre approche syntaxique STS, nous avons combinés nos matrices de similarité et en donnant un poids de 0.5 pour chaque matrice, le meilleur résultat était en combinant avec STS sémantique qui utilise le skipgram avec un poids de 0.1 pour la similarité d'ordre nous avons obtenus une corrélation de 78.07 % à Gomaa, nous nous limitons à représenter le résultat de cette dernière.

	Combinaison STS, SkipGram wf=0.1					
	Sans Stem		Khoja stem		Light stem	
	CP(%)	EQ(%)	CP(%)	EQ(%)	CP(%)	EQ(%)
Evaluation	CP (%)	EQ (%)	CP (%)	EQ (%)	CP (%)	EQ (%)
Gomaa	75.556	1.27	76.887	1.145	78.075	1.204
Ar 250	69.440	1.128	72.412	1.148	70.889	1.118
Ar 368	70.817	1.095	76.741	1.056	70.592	1.109

Tableau 34: Combinaison STS, SkipGram wf=0

CP : Corrélation de pearson
EQ : Erreur quadratique

C. Combinaison des trois STS en utilisant la combinaison des matrices :

Dans la table 35 nous retrouvons les résultats de l'application de la combinaison des trois matrices sémantiques et syntaxiques, en donnant un poids de 1/3 pour chacune, nous avons obtenus une corrélation de 76.08 % à Gomaa.

		Combinaison STS (Sem-WE-Synt)							
		CBOW				SKIP-GRAM			
		CNN_Corpora		khaleej		CNN_Corpora		khaleej	
	WF	wf=0.0	wf=0.1	wf=0.0	wf=0.1	wf=0.0	wf=0.1	wf=0.0	wf=0.1
Gomaa	RMSE	1,14	1.178	1.13	1.17	1.25	1.19	1.26	1.17
	CP (%)	76,02	75.68	75.70	75.66	75,07	75.76	74.91	76.08
STS 250	RMSE	1,15	1.16	1.18	1.17	1.14	1.17	1.18	1.18
	CP (%)	71,61	71.71	71.34	71.36	71.42	71,12	71.24	71.07

Tableau 35: Combinaison Trois STS Sémantique, STS /WE, Syntaxique

D. Combinaison des meilleures approches des trois binômes :

Dans la table 36 nous retrouvons les résultats de l'application de la combinaison des trois meilleures approches pour les trois binômes en utilisant la combinaison moyenne, nous avons obtenus une corrélation importante de 80.63 % à Gomaa.

Combinaison BEST (Sem-WE-Synt)	
Gomaa Dataset	
Corrélation de Pearson : **80.63**	Erreur Quadratique : **1.10**

Tableau 36: Combinaison BEST (Sem-WE-Synt)

E. Combinaison Meilleures approches sémantique/syntaxiques

Dans la table 37 nous retrouvons les résultats de l'application de la combinaison de notre meilleure approche syntaxique et la meilleure approche sémantique développée par Asma et Yasmine en utilisant la combinaison moyenne, une amélioration remarquable pour la similarité sémantique et le meilleur résultat obtenu pour le Binôme Asma et Yasmine.

Combinaison Best-sémantique et Best-syntaxique	
Gomaa Dataset	
Corrélation de Pearson : **81.49**	Erreur Quadratique : **0.98**

Tableau 37: Combinaison Best-sémantique et Best-syntaxique

F. Combinaison Meilleures approches sémantique avec WE /syntaxiques

Dans la table 38 nous retrouvons le résultat de l'application de la combinaison de notre meilleure approche syntaxique et la meilleure approche sémantique qui utilise les WE développée par Adel et Hamza, cette combinaison était la meilleure combinaison obtenue pour les trois binômes, en donnant un poids de 0.19 pour notre approche syntaxique et 0.81 pour la meilleure approche sémantique qui utilise les WE, et c'était le meilleur résultat obtenu dans ce travail de l'évaluation automatique pour la langue arabe.

Combinaison Best-Sémantique (0.81), Best-syntaxique (0.19)	
Gomaa Dataset	
Corrélation de Pearson : **84.2313**	Erreur Quadratique : **0.93**

Tableau 38: Combinaison Best-Sémantque, Best-Syntaxique

CP : Corrélation de pearson
EQ : Erreur quadratique

Conclusion pour l'hybridation des approches sémantiques et syntaxiques :

La combinaison des travaux des trois binômes a apportée une amélioration pour le domaine d'évaluation automatique, on a atteint une corrélation très importante de 84.23 % et une erreur quadratique optimale de 0.93 %.

ii. Discussion :

Les résultats obtenus sont très satisfaisants pour l'évaluation automatique dans langue arabe, si on prend en compte que la notion de similarité syntaxique, on a pu atteindre une corrélation de Pearson de 82.62 % pour la data set de gomaa qui illustre notre contexte d'évaluation automatique pour la langue arabe, et 78.32 % pour le data set AR368, et enfin une corrélation de Pearson de 71.76 % pour le data set AR250, ce qui n'est pas négligeable pour un tel data set qui n'est pas natif arabe.

Ce que les stemmers ont apportés à notre travail, on constate qu'au niveau d'un data set natif arabe et construit dans le domaine de l'enseignement ou le vocabulaire des phrases à comparer est à peut près le même , un stem lourd est moins apprécié par rapport au stem léger, il donne une corrélation de Pearson de 80 % alors qu'avec un stem léger il donne un 82 %, on explique cette différence, est que le stemmer lourd transforme les mots à leur racines, ce qui détériore le résultat, car il se peut que des mots différents auront la même racine avec un stem lourd, comme le cas avec les deux mots « الكائن», «يتكون»,la racine des deux mots est la même « كون» et donc la similarité syntaxique va les considérer comme similaires alors que ce n'est pas le cas, par contre un stem léger est apprécié et donne un résultat meilleur que de laisser les phrases sans aucun stem, on explique ça par l'unification des mêmes mots, s'ils sont au pluriel ou singulier, si un pronom possessif est ajouté, vu qu'en arabe les pronoms possessifs sont une partie du mot, donc un stem léger vient pour les enlever, et les mots de même famille seront considérés similaires syntaxiquement.

D'un autre côté, le stem lourd a été très efficace dans le data set AR250, ou les phrases à comparer ne sont pas du même vocabulaire comme les phrases de Gomaa, une explication qui va contredire l'explication au niveau de data set Gomaa, ou le vocabulaire utilisé est très différent, vu que le data set n'est pas natif arabe, le stem lourd a imposé son rôle, et a donné un meilleur résultat comparant à un stem léger ou bien sans exécuter aucun stem.

Les combinaisons entre approches syntaxiques ont pu améliorer le résultat au niveau du data set AR250, pour un stem léger mais pas pour dépasser le meilleur résultat syntaxique qui a été obtenu par l'approche développée STS avec un stem lourd, mais elles ont apportées une amélioration au niveau de data set Ar368 en combinant nos approches proposées avec Dice.

Nous remarquons que la combinaison avec l'approche sémantique permet d'atteindre des résultats nettement meilleurs et atteindre une corrélation très optimale de 84.23 %.

La similarité LCS classique ne prenait en considération que la chaine la plus longue, notre approche STS qui considère la chaine la plus petite aussi a pu apporter une amélioration à la similarité syntaxique, aussi la combinaison de notre approche avec différentes approches syntaxiques déjà existantes et par conséquent à la similarité en général, ce qui est à mentionner aussi, et à critiquer c'est bien l'utilisation de LCS classique, ce dernier qui considère la chaine commune la plus longue non consécutive, et pas une chaine consécutive, car l'utilisation d'une chaine consécutive va détériorer les résultats , ce que nous avons remarqués dans d'autres travaux.

Pour conclure, le travail sur la similarité syntaxique a pu apporter une amélioration au travail de similarité sémantique effectué au même temps par les autres binômes, ce qui prouve le rôle de la similarité syntaxique pour améliorer le résultat des approches sémantiques dans la langue arabe et par conséquent l'évaluation autoatique dans la langue arabe.

V. Conclusion et perspectives

Pour conclure, on va présenter ces deux tableaux récapitulatives 39 et 40 des résultats obtenus du data set de Gomaa et le data set Ar 250, pour récapituler les interprétations et extraire les meilleures conclusions de notre travail :

atégories	Les approches de lemmatisation	Sans stemming		Stemmer Khoudja		Light Stemmer	
		C.pearson	E.quadratique	C.pearson	E.quadratique	C.pearson	E.quadratique
es similarités yntaxiques	Approche STS(wf=0.1)	72.956	1.420	77.523	1.112	74.806	1.407
	NLCS	68.953	1.7	70.909	1.599	68.670	1.751
	NMCLCS1	32.528	3.021	37.178	2.368	34.268	2.57
	NMCLCSn	47.259	2.616	52.711	2.341	51.383	2.44
	Approche TFSS	71.597	1.289	76.509	1.132	76.037	1.094
	LCS	64.265	1.611	67.511	1.533	63.96	1.453
	Cosine	70.444	1.399	77.284	1.096	74.907	1.3
	Bgram	35.930	2.138	42.290	1.933	42.625	2.01
	Trigram	48.815	2.3	52.453	2.193	49.812	2.33
	Dice	81.930	0.998	80.676	1.127	82.627	1.005
	Jaro	66.036	1.575	70.326	1.413	66.311	1.531
	Jaccard	70.673	1.209	74.653	1.114	70.625	1.210
	Euclidienne	42.654	1.582	53.102	1.443	41.862	1.56
	Damerau-levenshtein	64.482	1.663	67.615	1.528	63.825	1.581
	Water smith	36.66	1.962	30.185	1.951	37.591	1.37
	Overlap	39.510	1.855	47.30	1.834	45.180	1.83
	Needlman	65.210	1.626	67.031	1.447	64.678	1.456
ombinaison les approces yntaxiques	TFSS , Dice	79.138	1.063	80.523	1.053	80.834	1.1099
	Combinaison >65% (Moyenne)	77.328	1.324	78.497	1.353	78.198	1.393
	combinaison >75% (Moyenne wf0)	(juste Dice) 81.930	0.998	79.246	1.0931	79.459	1.193
	STS , Dice (Moyenne)	77.889	1.311	80.251	1.087	78.926	1.305

Tableau 39: Résultat global du data set Gomaa

Catégories	Les approches de lemmatisation	Sans stemming		Avec stemmer Khoudja		Light stemmer	
		C.pearson	E.quadratique	C.pearson	E.quadratique	C.pearson	E.quadratique
Les similarités syntaxiques	STS	65.944	1.265	71.76	1.145	69.614	1.144
	LCS	62.527	1.216	61.364	1.280	61.247	1.248
	similarité GOMAA	67.708	1.177	68.572	1.208	69.910	1.103
	NLCS	65.254	1.285	66.161	1.198	65.345	1.256
	NMCLCS1	42.792	2.126	40.741	2.159	40.195	2.018
	NMCLCSn	57.639	1.827	63.471	1.488	61.383	1.624
	Trigram	32.110	2.357	38.698	2.327	29.676	2.437
	Cosine	61.976	1.425	69.514	1.209	63.460	1.482
	Bgram	45.166	2.052	42.21	2.01	41.493	2.119
	Dice	68.511	1.128	70.654	1.163	70.083	1.103
	Jaro	59.414	1.227	61.432	1.254	62.118	1.194
	Jaccard	55.603	1.338	63.779	1.25	57.612	1.269
	Euclidienne	32.320	1.579	54.683	1.371	35.735	1.480
	Damerau-levenshtein	61.856	1.364	59.117	1.308	59.862	1.315
	Water smith	29.743	1.813	44.875	1.937	37.751	1.712
	Overlap	46.758	1.767	56.703	1.713	47.984	1.774
	Needlman	61.743	1.228	60.579	1.255	61.843	1.256
Combinaisons des approches Syntaxique	Dice,STS(moyenne)	67.864	1.166	71.618	1,111	69.659	1.151
	Combinaison 65(moyenne)	67.231	1.18	71.252	1.106	69.071	1.203
	Sts,Dice,Jaccard,Jarro,Cosine(moyenne)	65.846	1.193	70,951	1.122	70.200	1.112

Tableau 40: Résultat global du data set Ar250

Les similarités à prendre en considération pour l'évaluation automatique dans la langue arabe sont les approches avec lesquelles nous avons obtenues une corrélation de Pearson supérieure à 70 % dans le data set de Gomaa, les approches qu'on avait marquées en vert dans les deux tableaux 39 et 40, c'est-à-dire : nos deux approches syntaxiques proposées TFSS et STS, Dice, Jaro, Jaccard, et cosine, et les approches à bannir, c'est les approches qui ont données une valeur inférieure à 70 %, qui sont marquées en rouge.

Le rôle de la similarité syntaxique ne pourrait être ignoré pour améliorer la notion de similarité sémantique entre les phrases dans la langue arabe, et donc pour améliorer l'évaluation automatique pour la langue arabe.

L'utilisation des différents data set pour l'évaluation de notre système, nous a permet de voir l'impact de la similarité syntaxique sur les différents domaines, d'où on a remarqué que la similarité syntaxique peut donner un résultat bien estimé dans un data set natif arabe construit dans le domaine de l'éducation et donc un data set natif arabe est préféré pour évaluer les approches implémentés et voir leur impact sur la langue arabe.

Le contexte de notre travail de l'évaluation automatique et le déroulement du travail en parallèle avec la similarité sémantique nous a permet d'exploiter les différentes approches de similarité syntaxiques et aussi sémantiques, ainsi d'avoir une idée générale sur les systèmes ASAG, et leur fonctionnement dans la langue arabe.

En perspectives, nous aurons aimés tester nos approches syntaxiques sur des data set de différente langue, pour voir si la similarité syntaxique joue un rôle si important en d'autres langues qui sont moins riches et complexes que l'arabe.

Et enfin, avoir la possibilité d'injecter notre système d'évaluation automatique basé sur nos meilleures approches syntaxiques combinées aux meilleures approches sémantiques autant qu'un plugin sur la plateforme Moodle, et laisser la possibilité d'une réalisation concrète de système.

VI. Bibliographie

[1] W. H. Gomaa et A. A. Fahmy, « Arabic Short Answer Scoring with Effective Feedback for Students », *Int. J. Comput. Appl.*, vol. 86, n° 2, p. 35–41, 2014.

[2] T. Mitchell, T. Russell, P. Broomhead, et and N. Aldridge, « Towards robust computerised marking of free-text responses FREE-TEXT RESPONSES », 2002.

[3] D. Perez-Marin, « Adaptive Computer Assisted Assessment of free-text students' answers: An approach to automatically generate students' conceptual models », 2009.

[4] E. Negre, « Comparaison de textes: quelques approches... », 2013.

[5] P. Dessus *et al.*, « Free-Text Assessment in a Virtual Campus To cite this version : HAL Id : hal-01547314 Free-Text Assessment in a Virtual Campus », 2017.

[6] N. A. Chinchor, « Overview of Muc-7/Met-2 », *Seventh Messag. Underst. Conf.*, 1998.

[7] R. Swartz, J. Burstein, C. Leacock, R. Swartz, J. Burstein, et C. Leacock, « Automated evaluation of essays and short answers », *5th Comput. Assess. Conf.*, 2001.

[8] S. Abney, « Part-of-Speech Tagging and Partial Parsing », *Corpus-Based Methods Lang. Speech*, p. 118–136, 1996.

[9] D. Marcu, *The theory and practice of discourse parsing and summarization*. MIT press, 2000.

[10] R. Williams et H. Dreher, « Automatically Grading Essays with Markit© », *Issues Informing Sci. Inf. Technol.*, vol. 1, p. 693–700, 2004.

[11] C. Leacock et M. Chodorow, « C-rater: Automated Scoring of Short-Answer Questions », *Comput. Hum.*, vol. 37, p. 389–405, 2003.

[12] R. Siddiqi, « Improving learning and teaching through automated short answer marking », p. 174, 2010.

[13] H. Wachsmuth, B. Stein, et G. Engels, « Information extraction as a filtering task », in *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, 2013, p. 2049–2058.

[14] H. Wachsmuth, B. Stein, et G. Engels, « Constructing efficient information extraction pipelines », *Cikm*, p. 2237, 2011.

[15] J. Burstein, S. Wolff, et C. H. I. Lu, « Free-Responses ».

[16] B. Dorr, J. Hendler, S. Blanksteen, et B. Migdaloff, « On beyond syntax: Use of lexical conceptual structure for intelligent tutoring », *Intell. Lang. tutors Theory Shap. Technol.*, p. 289–310, 1995.

[17] D. Callear, J. Jerrams-Smith, V. Soh, D. J. Jerrams-smith, et H. P. Ae, « CAA of Short Non-MCQ

Answers », in *In Proceedings of the 5th International CAA conference*, 2001.

[18] D. Sima, B. Schmuck, S. Szöll\Hosi, et Á. Miklós, « Intelligent short text assessment in eMax », in *Towards intelligent engineering and information technology*, Springer, 2009, p. 435–445.

[19] D. Sima, B. Schmuck, et S. Szollosi, « Intelligent short text assessment in eMax », in *AFRICON 2007*, 2007, p. 1–7.

[20] G. Miller, *WordNet: An electronic lexical database*. MIT press, 1998.

[21] D. Fi et H. A. L. Id, « Building a free French wordnet from multilingual resources Benoˆ ıt Sagot , Darja Fi˘ To cite this version »:, 2011.

[22] C. Leacock et M. Chodorow, « Combining local context and WordNet similarity for word sense identification », in *MIT Press*, 1998, p. 265–283.

[23] M. Palmer, « VERB SEMANTICS AND LEXICAL Zhibiao W u », p. 133–138.

[24] P. Resnik, S. M. Laboratories, et T. E. Drive, « Taxonomy », vol. 1, 1977.

[25] D. Lin, « An Information-Theoretic Definition of Similarity log P (common (A ; B)) (A ; B) = log P (description (A ; B)) », 1989.

[26] J. J. Jiang, « Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy », n° Rocling X, 1997.

[27] E. Gabrilovich et S. Markovitch, « Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis », p. 1606–1611, 2006.

[28] P. D. Turney, « Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL », in *Machine Learning: ECML 2001*, 2001, p. 491–502.

[29] D. Lin, « Extracting Collocations from Text Corpora », 1995.

[30] T. Mikolov, G. Corrado, K. Chen, et J. Dean, « Vector Space », p. 1–12.

[31] A. Mnih et G. Hinton, « A Scalable Hierarchical Distributed Language Model », p. 1–8.

[32] T. Mikolov, W. Yih, et G. Zweig, « Linguistic Regularities in Continuous Space Word Representations », n° June, p. 746–751, 2013.

[33] M. Karafi et J. H. Cernock, « ˘ a s a s », n° September, p. 1045–1048, 2010.

[34] T. Mikolov, K. Chen, G. Corrado, et J. Dean, « Distributed Representations of Words and Phrases and their Compositionality », p. 1–9.

[35] P. Vincent, « A Neural Probabilistic Language Model », vol. 3, p. 1137–1155, 2003.

[36] J. Pennington, R. Socher, et C. D. Manning, « GloVe : Global Vectors for Word Representation ».

[37] R. Mihalcea, C. Corley, et C. Strapparava, « Corpus-based and Knowledge-based Measures of Text Semantic Similarity », p. 775–780, 2005.

[38] Y. Li, D. Mclean, Z. Bandar, J. D. O. Shea, et K. Crockett, « Sentence Similarity Based on

Semantic Nets and Corpus Statistics », p. 1–35.

[39] A. Islam et D. Inkpen, « Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity », vol. 2, n° 2, 2008.

[40] K. Shaalan, Y. Samih, M. Attia, P. Pecina, et J. Van Genabith, « Arabic Word Generation and Modelling for Spell Checking », p. 719–725.

[41] S. I. Hajeer, « Comparison on the Effectiveness of Different Statistical Similarity Measures », vol. 53, n° 8, p. 14–19, 2012.

[42] H. Khafajeh et G. G. Kanan, « AUTOMATIC QUERY EXPANSION FOR ARABIC TEXT RETRIEVAL BASED ON AUTOMATIC QUERY EXPANSION FOR ARABIC TEXT RETRIEVAL BASED ON ASSOCIATION AND », n° October, 2015.

[43] A. O. Al-Thubaity, « A 700M$\mathplus$ Arabic corpus: {KACST} Arabic corpus design and construction », *Lang. Resour. Eval.*, vol. 49, n° 3, p. 721–751, oct. 2014.

[44] L. Ouahrani, « String similarity for Arabic short answer grading », *Intern. Rep.*, p. 118.

[45] E. Atwell, « A Review of Semantic Search Methods to Retrieve Information from the Qur ' an Corpus », 2015.

[46] I. Retrieval, *Introduction to Information Retrieval*. 2008.

[47] M. Mustafa, A. S. Eldeen, S. Bani-ahmad, et A. O. Elfaki, « A Comparative Survey on Arabic Stemming : Approaches and Challenges », p. 39–67, 2017.

[48] S. Khoja et R. Garside, « Stemming arabic text », *Lancaster, UK, Comput. Dep. Lancaster Univ.*, 1999.

[49] K. Darwish et C. Park, « Building a Shallow Arabic Morphological Analyzer in One Day », p. 1–12.

[50] T. Buckwalter, « Issues in Arabic Orthography and Morphology Analysis », p. 3–6.

[51] A. Abdelali, « Localization in modern standard Arabic », *J. Am. Soc. Inf. Sci. Technol.*, vol. 55, n° 1, p. 23–28, 2004.

[52] S. Ghwanmeh, G. Kanaan, R. Al-Shalabi, et S. Rabab'ah, « Enhanced Algorithm for Extracting the Root of Arabic Words », in *2009 Sixth International Conference on Computer Graphics, Imaging and Visualization*, 2009, p. 388–391.

[53] M. N. Al-Kabi, S. A. Kazakzeh, B. M. Abu Ata, S. A. Al-Rababah, et I. M. Alsmadi, « A novel root based Arabic stemmer », *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 27, n° 2, p. 94–103, 2015.

[54] K. Darwish et D. W. Oard, « CLIR Experiments at Maryland for TREC-2002 : Evidence combination for Arabic-English retrieval 1 Introduction 2 Methodology », n° 1, 2002.

[55] W. Gomaa et A. Fahmy, « Automatic scoring for answers to Arabic test questions », *Comput. Speech Lang.*, vol. 28, 2013.

[56] P. Kolb, « DISCO: A multilingual database of distributionally similar words », in *In Proceedings of KONVENS*.

[57] D. Cer, M. Diab, E. Agirre, L. Specia, M. View, et B. Country, « SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Cross-lingual Focused Evaluation », 2017.

[58] A. Magooda, M. A. Zahran, M. Rashwan, H. Raafat, et M. B. Fayek, « Vector Based Techniques for Short Answer Grading », p. 238–243, 2012.

[59] E. Moatez et B. Nagoudi, « Semantic Similarity of Arabic Sentences with Word Embeddings », p. 18–24, 2017.

[60] P. A. V Hall et G. R. Dowling, « Approximate String Matching », *ACM Comput. Surv.*, vol. 12, n° 4, p. 381–402, 1980.

[61] B. Lovins, « Development of a Stemming Algorithm * », vol. 11, n° June, p. 22–31, 1968.

[62] M. A. Jaro, « Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida », *J. Am. Stat. Assoc.*, vol. 84, n° 406, p. 414–420, 1989.

[63] M. A. Jaro, « Probabilistic linkage of large public health data files », *Stat. Med.*, vol. 14, n° 5–7, p. 491–498, 1995.

[64] S. B. Needleman et C. D. Wunsch, « A General Method Applicable to Search for Similarities in Amino Acid Sequence of 2 Proteins », *J. Mol. Biol.*, vol. 48, p. 443–453, 1970.

[65] M. S. Waterman, « Identification of Common Molecular Subsequences Identification of Common Molecular Subsequences », p. 195–197, 1981.

[66] E. F. Krause, *Taxicab Geometry : an adventure in non-Euclidean geometry*, Livre impr. 1987.

[67] P. Jaccard, « Étude comparative de la distribution florale dans une portion des Alpes et des Jura », *Bull. del la Société Vaudoise des Sci. Nat.*, vol. 37, p. 547–579, 1901.

[68] L. Allison et T. Dix, « A bit-string longest-common-subsequence algorithm », *Inf. Process. Lett.*, vol. 23, p. 305–310, 1986.

[69] W. Group, « t e r m n i u s d a i g o n a l », 1999.

[70] S. Fernando et M. Stevenson, « A Semantic Similarity Approach to Paraphrase Detection », *Proc. 11th Annu. Res. Colloq. UK Spec. Interes. Gr. Comput. Linguist. (CLUK 2008)*, p. 45–52, 2008.

[71] S. R. Bowman et C. Potts, « A large annotated corpus for learning natural language inference ».

[72] J. Cohen, « Statistical power analysis for the behavioral sciences. 2nd ». Hillsdale, NJ: erlbaum, 1988.

[73] W. Greene, « Annexes : exercices et corrigés ».

Annexes

Liste des mots d'arrêtes (Stop words) :

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