Ministère de l'enseignement supérieur et de la recherche scientifique

UNIVERSITE SAAD DAHLEB DE BLIDA 1

Faculté des Sciences

Département d'informatique

MEMOIRE DE MASTER

En Informatique

Spécialité : Ingénierie du Logiciel

THEME

Généralisation d'approches basées Espace Sémantique pour un système d'évaluation des réponses courtes adapté à la langue arabe.

Réalisé par :

Abdellaoui Salah

Proposé et encadré par :

Mme. OUAHRANI Leila

Composition de jury :

M. Ferfera Sofiane	Président
Mme. Cherfa Imene	Examinateur

Soutenu Le : 30 / 09 / 2019.

الملخص

في هذا العمل البحثي، نحن مهتمون بمجال التدريس، بشكل أكثر دقة تقييم الإجابات القصيرة للطلاب، أو ما يسمى **ASAG**(نظام التقييم التلقائي للإجابات القصيرة). أثبت التقييم التلقائي فعاليته الكبيرة في توفير العديد من الفوائد للمعلمين والطلاب من حيث الجهد والوقت. عملنا هو استمرار لعمل بحثي تم إنجازه بالفعل في عام 2018 ويتناول مقاييس التشابه الدلالي لنظام التقييم التلقائي للإجابات القصيرة المطبقة على اللغة العربية، والتي تعاني من نقص الأدوات والموارد اللغوية. وقد تم اقتراح منهج فلكي قائم على الفضاء الدلالي للكلمات. لذا فإن مهمتنا هي التحقق من صحة وقابلية تقبل هذا المنهج المقترح للغات الأخرى التي تعاني من نقص الأدوات اللغوية مثل اللغة العربية. لتنفيذ هذا التحقق، اخترنا اللغة الإنجليزية ووضعنا العناصر المناسبة لتطوير هذا المنهج وتعميمه لباقي اللغات.

الكلمات المفتاحية: التقييم التلقائي للإجابات القصيرة، جذر الكلمة، ASAGS، معالجة اللغات الطبيعية، NLP، قياسات التشابه، التشابه الدلالي، فضاء الناقلات، الفضاء الدلالي، ناقلات السياق.

Résumé

Dans ce travail de recherche nous nous intéressons au domaine de l'enseignement plus précisément l'évaluation des réponses courtes des étudiants, dites ASAG (Automatic Short Answer Grading System). L'évaluation automatique a prouvé sa grande efficacité à fournir de nombreux avantages pour les enseignants et les étudiants en terme d'effort et de temps. Notre travail rentre dans la continuité d'un travail de recherche déjà réalisé en 2018 qui traite des mesures de similarité sémantique pour un système d'évaluation automatique des réponses courtes appliqué à la langue arabe qui souffre de manque d'outils et de ressources linguistiques. Une approche basée espace sémantique a été proposée. Ainsi notre travail consiste à valider la généricité de cette approche proposée aux autres langues souffrant du manque d'outils l'linguistiques que la langue arabe. Pour réaliser cette validation nous avons choisi la langue anglaise et élaboré les éléments adéquats de développement d'une telle généralisation de l'approche.

Mots clé : Evaluation automatique des réponses courtes, ASAGS, Stem, traitement du langage naturel, TALN, mesures de similarité, similarité sémantique, espace vectoriel, espace sémantique, vecteur de contexte.

Abstract

In this research work, we are interested in the field of teaching more precisely the evaluation of the short answers of the students, known as ASAG (Automatic Short Answer Grading System). The automatic evaluation has proved its great efficiency in providing many benefits for teachers and students in terms of effort and time. Our work is a continuation of a research work already done in 2018 that deals with semantic similarity measures for a system of automatic assessment of short answers applied to the Arabic language, which suffers from a lack of tools and linguistic resources. A semantic space-based approach has been proposed. Therefore, our job is to validate the genericity of this proposed approach to other languages suffering from the lack of linguistic tools than the Arabic language. To carry out this validation we chose the English language and elaborated the appropriate elements of development of such a generalization of the approach.

Keywords: automatic short answer grading, ASAGS, Stem, natural language processing, NLP, similarity measurements, semantic similarity, vector space, semantic space, context vector

Remerciements

Nous tenons tout d’abord à remercier **Allah** le tout puissant, qui nous a donné la force, la capacité et surtout la patience pour accomplir ce travail.

Nous tenons à adresser nos plus chaleureux remerciements à Madame « Ouahrani Leila », notre promotrice de ce PFE, pour son aide illimitée et son soutien pendant toute cette période pleine de défis. Ses conseils avisés et sa patience nous ont aidés à surmonter l'hésitation, l'embarras et de rediriger le projet pour avoir de meilleurs résultats. Son œil critique nous a été très précieux pour structurer le travail et pour améliorer la qualité des différentes sections sans oublier sa constante disponibilité à notre égard et de nous avoir donné l’occasion de travailler sur un tel sujet de recherche qui est très riche d’information et de découverte pour nous.

Notre gratitude s'adresse également à l'université de Bouira qui nous a fourni un accès sur un serveur à distance pour accomplir notre travail.

Nos vifs et chaleureux remerciement et gratitude à nos parents, frères, sœurs ainsi que toute la famille pour leur soutien, encouragement et amour illimité. « Elhamdoullilah de vous avoir à nos côtés, nous vous aimons du fond du cœur ».

Nous tenons à exprimer nos sincères remerciements à nos amies pour leur soutien et encouragement qu’avec, nous avons pu surpasser des périodes de stress.

Nous tenons à remercier toute personne qui a participé de près ou de loin à l’exécution de ce modeste travail.

Liste des tables

Liste des figures

Liste des abréviations

ASAGS	**Système d'évaluation automatique des réponses courtes (Automatic Short Answer Grading System)**
RM	**Réponse Modèle**
RE	**Réponse Etudiant**
NLP/TALN	**Traitement automatique du langage naturel**

Table des matières

Chapitre 1 : Contexte et problématique

Dans ce premier chapitre on va présenter une introduction générale sur notre travail. Elle nous permet de préciser la problématique et de formuler les objectifs et mettre en évidence l'importance de notre travail en précisent ces champs et limites.

1. Introduction générale

L'évaluation du savoir acquis par les élèves est la tâche principale des enseignants. Ce travail est effectué classiquement par l'évaluation et l'attribution des notes sur les réponses des étudiants en classe. Cette évaluation se fait sous la forme de quiz, examen, feuille de travail ou interrogations ...etc.

Cette évaluation est importante car elle fournit aux enseignants une vue sur leurs méthodes d'enseignement et à juger, si ces dernières ont été efficaces ou non. Cependant cette évaluation est une tâche individuelle et un processus ennuyeux, répétitif et qui consomme beaucoup de temps et vue aussi par fois comme une tâche casse-tête et non ré-compensative…

De plus cette dernière n'aide que rarement les enseignants à améliorer leurs manière d'enseignement du a plusieurs raisons comme l'utilisation de petit exercices généralement de difficulté facile a porté de tous les étudiants et aussi l'utilisations répétitives des QCM, car l'évaluation avec un sujet consistant comme celui des examens engendra beaucoup de temps et plus d'efforts de la part des enseignants.

Pour pallier à tous ces problèmes nous nous intéressons aujourd'hui à l'évaluation assisté par ordinateur noté CAA (Computer Assisted Assessment).

Des systèmes d'évaluation automatique sont en pratique dans le domaine de l'enseignement particulièrement l'enseignement en ligne noté MOOC (Massive open online course) depuis plusieurs années, Toutefois ces systèmes ne traitant que les questions à choix multiples (QCM) ou à réponses vrai/faux sont facile pour un ordinateur mais ces derniers sont reportés par les chercheurs qu'ils ont jugé faible et insuffisantes pour évaluer les connaissances acquises comparés aux réponses libres, Premièrement pour le fait de deviner les réponses correct et deuxièmes pour la présence de possibilités de plusieurs réponses juste.

En revanche, les questions à réponses courtes qui nécessitent des réponses données par les examinés en langage naturel ont été jugées plus efficaces pour évaluer les connaissances acquises par les apprenants.

Cependant, la réalisation d'un système d'évaluation automatique pour les réponses courtes noté ASAGS (pour Automatic Short Answer Grading System) qui traite les

réponses dans la langue naturel présente un défi en raison de sa complexité vis-à-vis du côté morphologique[1] et sémantique[2] ainsi que la variation linguistique[3] et la nature subjective[4] de l'évaluation.

En outre, le type de question a un impact important sur la réponse. Chaque question nécessite un type précis de réponse. Nous nous intéressons aux questions à réponses ouverte[5]. La catégorie des réponses ouvertes permet de recueillir des réponses qualitatives et la plupart du temps riches en informations ce qui signifie sa complexité. Les questions à réponses courtes font partie des questions ouvertes qui regroupent aussi les essais. Ce type est caractérisé par la longueur et la profondeur des idées présentées. En raison du nombre limité de mots autorisés, les idées dans un court essai devraient être présentées clairement et succinctement. Un court essai devrait être d'environ cinq cents mots. Il est censé répondre à une question ou un argument. S'il s'agit d'un débat, une déclaration de thèse claire devrait être fournie pour montrer la partie prise dans l'argumentation et les arguments attendus à soulever. Les courts essais présentent souvent des opinions ou des opinions individuelles. Les idées sont présentées superficiellement puisque la durée de l'essai est limitée[1].

2. Problématique

Le *principe général de l'évaluation consiste à comparer la réponse de l'apprenant avec la réponse* ***modèle*** *(de référence) formulée par l'enseignant et d'attribuer un score. 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. 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, … Pour faire face à ces enjeux il a été développé, dans un travail précédent [3]**,** des approches statistiques d'évaluation automatique des réponses courtes utilisant une distribution multidimensionnelle basée sur les espaces sémantiques créés à partir de corpus de textes

[1] La morphologie est l'étude de la formation des mots et de leurs variations. Autrement, c'est le regroupement de différents mots à travers leurs parties, comme les suffixes, préfixes, radicaux

[2] La sémantique lexicale est l'étude du sens des mots -ou plutôt des morphèmes- d'une langue donnée.

[3] Une réponse donnée pourrait être articulée de différentes façons qui ont tous le même sens.

[4] Une question peut avoir de multiples réponses possibles.

[5] Une question ouverte est une question pour laquelle il n'y a pas de réponses préétablies proposées au répondant, celui-ci est donc entièrement libre dans sa réponse.[2]

en langue arabe. Les espaces sémantiques permettent de capturer les propriétés syntaxiques et sémantiques des mots. La problématique à considérer dans ce travail consiste à répondre à la question de savoir : à quel point les approches développées sont-elles **génériques et indépendantes** de la langue pour être appliquées à toute autre langue souffrant des mêmes défis que la langue arabe ? Pour répondre à cette question nous choisissons la validation par rapport à l'anglais vu la disponibilité des ressources linguistiques (corpus, data sets, …).

3. Objectifs du travail

- Généraliser les approches déjà développées pour la langue arabe à l'anglais en considérant les spécificités de l'anglais et développer les outils adéquats,
- Evaluer le degré de généricité des approches développées en validant par des data sets et des métriques de performance.

4. Importance de notre travail

Dans le cadre de ce travail, nous visons à apporter un plus à ce qui a été déjà fait. Autrement dit: réaliser un système d'évaluation automatique des réponses courtes qui introduit le concept de similarité sémantique (le sens des phrases), Ce travail a été déjà réaliser pour la langue arabe. Notre travail est de faire la même chose pour la langue anglaise en étudiant la généricité de ce système et dire si l'approche sur laquelle se base ce système de traitement de réponses courtes, est généralisable ou non pour les autres langues et particulièrement, celle souffrantes du même manque d'outils linguistique comme la langue arabe. Pour cela, nous devons adapter des techniques existantes dans le domaine du traitement automatique du langage naturel (NLP).

D'une part, comme mentionné précédemment, la langue arabe souffre du manque de ressources linguistiques (notamment le WordNet[1] arabe qui manque de richesse par rapport à son homologue anglais ainsi que les dictionnaires arabe en ligne). Par conséquent, nous sommes limités à certaines méthodes, mais aussi nous nous adaptons

[1] *WordNet* est une base de données de l'anglais regroupant des unités lexicales selon leurs relations sémantiques et lexicales.

avec le peu d'outils NLP disponibles sur le net bien qu'ils ne sont pas vraiment performants. Ainsi que l'indisponibilités des ressources linguistiques nous a rendu la tâche plus difficile. Ce qui nous a motivé à réaliser nos propres ressources tels que le corpus de domaine et les outils de traitement NLP.

D'autre part, les ASAGS réduisent considérablement le besoin d'implication humaine. Cela a un impact important sur le domaine de l'éducation, ce qui facilite ainsi une partie de la charge de l'enseignant, améliore ses performances et détermine si ses objectifs ont été atteints.

Parmi les avantages de l'automatisation du marquage, mentionnons les économies de temps et de coûts, ainsi que la réduction des erreurs et des injustices attribuables aux préjugés humains, à l'épuisement ou au manque de cohérence [3].

En effet, nous avons pu démontrer que ce système est généralisable c'est-à-dire qu'on peut l'adapter pour n'importe quelle autre langue qui est dans les mêmes circonstances que celle de l'arabe de préférences, ou même celles qui ne l'est pas (il faut juste avoir les données nécessaires que nous évoquerons dans les prochains chapitres).

5. Champs de notre travail et limites

- Ce travail est affilié au traitement automatique du langage naturel (TALN), qui est un domaine multidisciplinaire impliquant la linguistique, l'apprentissage automatique et donc l'intelligence artificielle. De plus, nous nous intéressons au domaine éducatif qui constitue la base du savoir et de la moralisation,
- Ce système exige la disponibilité d'une réponse modèle (réponse de référence),
- La disponibilité d'un corpus lié au domaine de la recherche ainsi qu'un stemmer de traitement du texte de la langue choisi est nécessaire dans notre démarche,
- Le système développé est dédié pour l'évaluation sur la langue anglaise.
- Ce système peut gérer deux langue à la fois : l'anglais et l'arabe. Et peut gérer d'autre langues si disponibilité d'un espace sémantique ou bien à la fois un corpus et un stemmer dédié à la langue choisie.

Chapitre 2 : Etat de l'art

La phase élémentaire dans l'initiation d'une étude ou d'une recherche commence par l'état de l'art sur le sujet abordé. Cela consiste à reformuler toutes les connaissances acquises ainsi que les travaux déjà faits sur la même thématique. Cette étape présente la base de prise de décision concernant la méthode et la technique adaptées dans cette recherche.

Dans ce chapitre nous allons entamer le fonctionnement global des ASAGS, les approches d'évaluation automatique de réponses courtes particulièrement celles qui traitent l'arabe, les approches de similarité existantes en général et précisément celles de la similarité sémantique, les outils et ressources NLP existants et disponibles en ligne ainsi que les travaux connexes.

Le principe du fonctionnement d'un ASAGS repose sur un ensemble de processus. Ce système doit avoir la question à poser bien formulée, claire et dans un contexte précis d'un examen réel convenable aux conditions ordinaires. De plus, une réponse modèle et sa note données par l'expert du domaine (l'enseignant) sont indispensables comme données de base pour le fonctionnement d'un ASAGS. Ensuite, l'utilisateur/étudiant saisie sa réponse qui doit répondre aux critères de la réponse courte cités précédemment. Par la suite, le système évalue cette réponse en calculant la similarité entre elle et les données de base en utilisant les outils et ressources NLP, puis il convertie cette similarité en un score (note) selon le barème donné et enfin renvoie ce résultat à l'utilisateur.

1. Fonctionnement global des ASAGS

ASAGS (pour Automatic Short Answer Grading System) qui traite les réponses dans la langue naturelle présente un défi en raison de la complexité de la langue vis-à-vis du côté morphologique et sémantique ainsi que la variation linguistique et la nature subjective de l'évaluation. On cite ici la langue arabe qui est un bon exemple de cette complexité. De plus, l'arabe est une langue assez répandue, parlée par plus de 300 millions de personnes à travers le monde. Du point de vue du langage naturel, la langue arabe se caractérise par une ambiguïté élevée et une morphologie riche et complexe sans oublier le manque considérable de ressources linguistiques : corpus arabes, lexiques et dictionnaires, outils de traitement NLP… Tous ces aspects repoussent et découragent le progrès de la recherche dans la branche de l'évaluation automatique des réponses courtes en langue arabe. C'est pour cela, peu de travaux ont été réalisés dans ce contexte.

D'une part, comme mentionné précédemment, la langue arabe souffre du manque de ressources linguistiques (notamment le WordNet arabe qui manque de richesse par rapport à son homologue anglais ainsi que les dictionnaires arabes en ligne).

WordNet : est une base de données de l'anglais regroupant des unités lexicales selon leurs relations sémantiques et lexicales.

2. Processus principaux des ASAGS

Dans la littérature, S. Burrows et al. [4] ont proposé un pipeline de développement de système ASAG représenté par 6 artefacts (rectangles) et 5 processus (ovales) « Voir Figure 1 ».

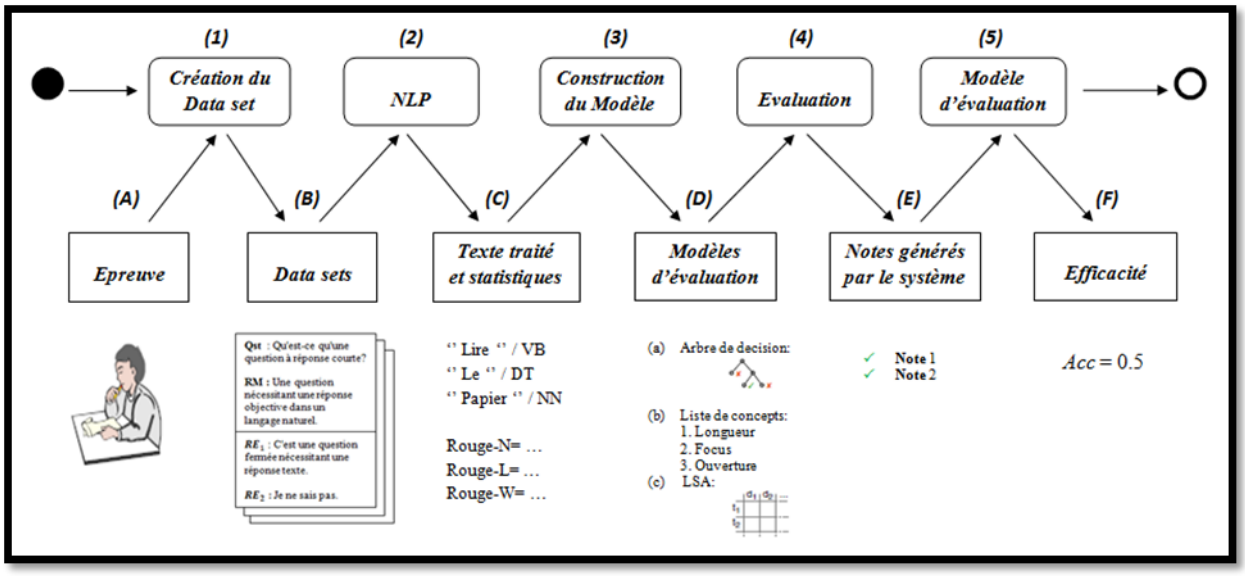


Figure 1 : Pipeline de développement de système ASAG [4].

Les modules du fonctionnement d'un ASAG sont relatifs les uns aux autres :

2.1. Création du dataset

Cette étape nécessite de faire une épreuve (A) (examen, test, interrogation…) pour plusieurs étudiants (apprenants). Les réponses collectées seront organisées avec leur réponse modèle (réponse type) pour chaque question de l'épreuve donnée par l'examinateur (l'enseignant) ainsi que l'évaluation manuelle donnée par ce dernier (notes manuelles). Toutes ces données représentent le dataset (B).

2.2. Traitement NLP du dataset

Le dataset (B) passe maintenant par une phase de traitement automatique du langage naturel NLP vue que les questions de (A) imposent des réponses en langage naturel. Ce traitement comprend de différentes techniques comme le stemming, la normalisation, étiquetage morpho-syntaxique (POS : Part-Of-Speech Tagging), calcul des pondérations (TF, IDF…) … Un ensemble de texte traité ainsi que des statistiques (C) résultent.

2.3. Construction du modèle d'évaluation

Cette étape est l'étape la plus importante dans le fonctionnement d'un ASAG car elle est la base du calcul de similarité entre la réponse étudiant et la réponse modèle. Les données textuelles traitées sont utilisées afin de les transformer en un modèle (ceci peut

exiger des ressources externes). Il existe plusieurs types de représentation du modèle (D) comme le VSM (modèle d'espace vectoriel), les arbres de décisions, …

2.4. L'évaluation automatique du dataset

Après avoir construit le modèle d'évaluation, il est maintenant possible de calculer la valeur de similitude entre le couple de réponses (RM, RE). La valeur de similarité varie dans l'intervalle [0, 1]. Par la suite, cette valeur sera interprétée. Pour se faire, un passage au score est effectué pour avoir la note automatique finale (E).

2.5. L'évaluation du modèle construit

Dans ce stade, une estimation de la qualité du modèle construit est indispensable. Pour se faire, une valeur de précision ou d'exactitude (F) est calculée comme repère qui reflète cette qualité. Plusieurs méthodes existent déjà pour cet objectif (Coefficient de corrélation de Pearson, l'erreur quadratique RMSE, la précision…).

3. Une vue sur l'historique des ASAG

Il existe plusieurs approches qui traitent le sujet d'évaluation automatique des réponses courte dont la recherche s'est développée depuis 1996. Steven Burrows et al. dans leur article [4] ont identifié 35 systèmes dans 4 méthodes différentes (représentés dans la Figure 2) **:**

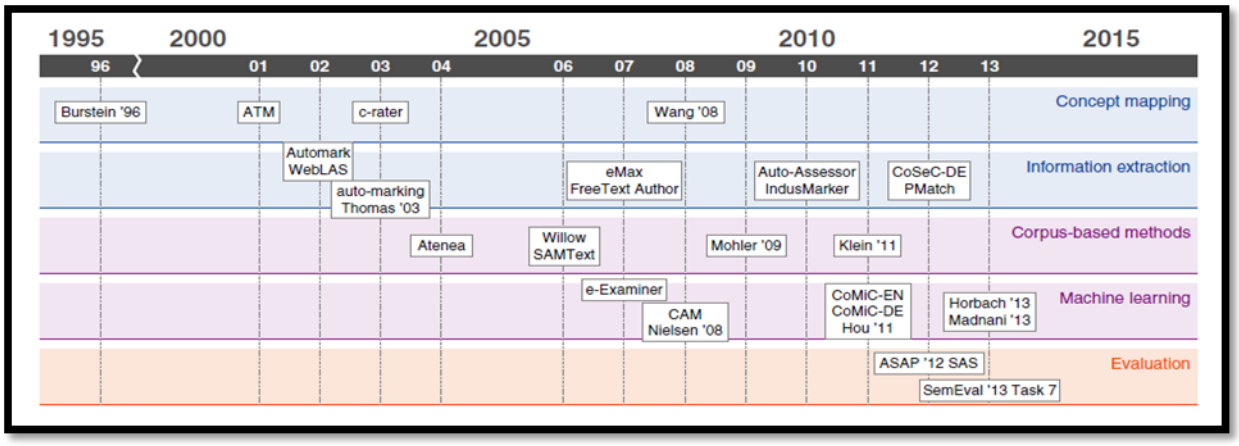


Figure 2 : Les ères et les tendances du classement automatique des réponses courtes [4].

3.1. Les méthodes basées sur le mappage de concepts (Concept Mapping) :

Le principe ici est de considérer les réponses des élèves comme constituées de plusieurs concepts et de détecter la présence ou l'absence du concept clé lors du classement. Cela se fait en parcourant les concepts clé (ceux de la réponse modèle) un par un dans la réponse étudiant et pour chaque concept trouvé, incrémenter le score.

Cette méthode s'adapte bien avec deux types de question. L'un demande une solution à un problème plus une justification. L'autre demande plusieurs explications au même problème.

ATM : Le marqueur de texte automatique ATM (Automatic texte Marker) [5] décompose les réponses des enseignants et des étudiants dans des listes de concepts minimaux comprenant pas plus de quelques mots chacune, et compte le nombre de concepts en commun pour fournir un score d'évaluation. Chaque concept est essentiellement la plus petite unité possible dans une réponse qui peut être attribuée un poids aux fins de classement. Les pondérations sont additionnées pour produire la note globale.[4]

C-rater : L'évaluateur conceptuel (Concept rater ou C-rater) [6] vise à faire correspondre autant de concepts de niveau de phrase que possible entre les réponses de l'enseignant et celles de l'élève. L'appariement est basé sur un ensemble de règles et une représentation canonique des textes utilisant la variation syntaxique, l'anaphore, la variation morphologique, les synonymes et la correction orthographique. Plus précisément, les réponses des enseignants sont entrées dans une phrase distincte pour chaque concept. Cela simplifie l'évaluation puisque seul un concept est considéré à la fois lors du classement. Cette technique évite d'avoir recours à une solution indirecte, telle que diviser la question en plusieurs parties [7] et l'on affirme que cela peut conduire à une plus grande précision [1] . En outre, le format d'entrée en langage naturel est avantageux par rapport à d'autres systèmes qui nécessitent une expertise et l'utilisation d'un langage de balisage [8]. [4]

3.2. Les méthodes basées sur l'extraction d'information (Information Extraction)

Le concept de cette méthode est l'extraction des données à partir des sources non structurées comme les textes libres, ensuite les mettre sous forme structurée (arbres d'analyse...). Les méthodes d'extraction d'informations peuvent être considérées comme

une série d'opérations de correspondance de modèle telles que les expressions régulières ou les arbres d'analyse. Dans le cas des systèmes des réponses courte, chaque réponse est présentée sous forme structuré et par la suite évaluer leurs dépendances.

Auto-évaluateur: ou Auto-Assessor [9] se concentre sur le classement (évaluation) des réponses des étudiants à phrase unique sous forme canonique basé sur la correspondance des coordonnées de sac de mots (BOW) et les synonymes avec WordNet [10]. Coordonner l'appariement dans l'ASAG se réfère simplement à des termes individuels correspondant entre les réponses des enseignants et des étudiants. Dans l'auto-évaluateur, chaque mot qui correspond exactement reçoit un point de crédit, les mots liés à partir du WordNet reçoivent un crédit partiel, tandis que le reste ne reçoit pas de crédit.

3.3. Les méthodes basées-corpus (Corpus Based Methods)

Le principe est d'utiliser des propriétés statistiques des corpus, qui sont des ensembles de textes. Ces méthodes peuvent être utiles lors de l'interprétation des synonymes dans les réponses courtes. Alors, afin de limiter les réponses correctes qui peuvent être identifiées, utiliser seulement le vocabulaire des réponses modèles. Ensuite, pour renforcer ce vocabulaire, prendre en considération ses synonymes et effectuer des traductions en autre langue afin d'éviter les difficultés de la langue source. C'est dans ce contexte que se situe notre travail.

3.4. Les méthodes basées sur l'apprentissage automatique (Machine learning)

Les systèmes d'apprentissage automatique utilisent généralement des données étiquetées qui sont les réponses modèles notées ainsi qu'un certain nombre de mesures extraites du langage naturel (techniques de traitement et similarité) qui sont ensuite combinées afin d'avoir un score en utilisant un modèle (fonction d'estimation) de classification ou de régression. Cela peut être soutenu par une boîte à outils d'apprentissage automatique telle que Weka [11].

Madnani et al. [12] mettent en place un système de notation des questions de compréhension de lecture sur les niveaux de vie (système nommé Madnani [13]. Chaque texte comporte trois paragraphes, et les réponses des élèves requièrent spécifiquement une phrase donnant un résumé global et trois phrases supplémentaires donnant un résumé de chaque paragraphe. L'approche d'apprentissage automatique comprend huit

caractéristiques (BLEU, ROUGE, mesures concernant différentes dimensions de la copie de texte, nombre de phrases et nombre de mots de connecteur de discours couramment utilisés) en tant qu'entrées dans un classificateur de régression logistique.[4]

4. Les modèles BOW et VSM

Dans la thématique des ASAGS, un modèle est défini comme toute représentation qui permet de mesurer le degré de similitude entre le couple (RM, RE) avec une précision raisonnable entre le score automatique et celui manuel. Fréquemment dans le domaine de l'apprentissage automatique, les données textuelles utilisées sont converties en vecteurs. Le BOW (pour Bag Of Words) est fait dans cet objectif. Le modèle de sac de mots est un moyen d'extraire des caractéristiques du texte pour les utiliser dans des algorithmes de l'apprentissage automatique, plus précisément dans le traitement automatique du langage naturel (TALN).

4.1. Le modèle BOW (bag of words)

C'est un modèle couramment utilisé qui permet de compter tous les mots dans un morceau de texte. Fondamentalement, il crée une matrice d'occurrences pour une phrase ou un document, sans tenir compte de la grammaire ni de l'ordre des mots. Ces fréquences de mots ou occurrences sont ensuite utilisées comme fonctionnalités pour faire de l'apprentissage automatique ou bien la classification des documents.

Par exemple, on va prendre 2 documents avec un petit paragraphe chacun.

Le 1er document: *it was the worst of times*

2éme document: i*t was the age of wisdom and the age of foolishness*

Maintenant, comptons les mots :

	it	was	The	times	of	worst	age	wisdom	and	foolishness
Doc 1	1	1	1	1	1	1	0	0	0	0
Doc 2	1	1	2	0	2	0	2	1	1	1

Cette approche peut refléter plusieurs inconvénients, tels que l'absence de contexte et de sémantiques, et le fait que les mots vides tel que (« it » ou « the ») ajoutent du bruit

à l'analyse. De plus certains mots ne sont pas pondérés correctement (le poids du mot « the » dans le 2ém document est inférieurs au mot " wisdom " dans le même document).

Pour résoudre ce problème, une approche consiste à redimensionner la fréquence des mots en fonction de leur fréquence d'apparition dans tous les textes (pas uniquement celui que nous analysons), de manière à pénaliser les scores des mots fréquents comme « the », qui serons décrémenté comparativement aux autres mots vue que ces mots se répétant presque dans tous les documents d'une manière flagrante. Cette approche de score s'appelle « Fréquence du terme et Fréquence de document inverse » (TFIDF) qui permet d'améliorer le sac de mots en fonction de la pondération. Grâce à TFIDF, les termes fréquents dans le texte sont « récompensés » (comme le mot « the » dans notre exemple), mais ils sont également « punis » si ces termes sont fréquents dans d'autres textes que nous incluons également dans l'algorithme. Au contraire, cette méthode met en valeur et « récompense » les termes uniques ou rares en considérant tous les textes. Néanmoins, cette approche n'a toujours pas de contexte ni de sémantique.

4.2. Le modèle VSM (modèle d'espace vectoriel) :

Une fois le BOW est construit, une étape de vectorisation est généralement nécessaire. Ceci se fait en construisant une matrice dont les caractéristiques varient selon le domaine. Un exemple simple et connue est la création d'une matrice M de taille D*X où D est l'ensemble de document et X est la taille du dictionnaire du vocabulaire du corpus (l'ensemble des lexèmes obtenus des documents). Chaque ligne i de la matrice est une représentation d'un document du corpus et chaque colonne j est un mot (terme) du dictionnaire tandis que l'élément M (i,j) représente une valeur binaire indiquant la présence ou l'absence du mot j dans le document i (1 pour présent et 0 pour absent) « Tableau 1 ». D'autres exemples de représentation seront présentés prochainement dans la « section 2.4.2. ».

Tableau 1 : Exemple de représentation VSM.

	Terme 1	Terme 2	Terme 3	…	Terme n
Document 1	1	1	0	…	0
Document 2	1	0	0	…	1
…	…	…	…	…	…
Document n	0	0	0	…	1

5. Les approches de similarité dans les systèmes d'évaluation automatique

La notion de similarité a été très tôt perçue comme un concept clé en intelligence artificielle ainsi qu'elle intervient dans plusieurs de ses domaines : l'apprentissage automatique, la recherche d'information, la détection de fraudes (l'empreinte digitale), la détection du plagiat, la traduction automatique des corpus, le résumé de texte…

Il existe trois catégories principales des approches de similarité (voir Figure 3) :

- Similarité syntaxique (String-based similarity).
- Similarité sémantique (Semantic similarity).
- Similarité hybride (hybrid similarity)

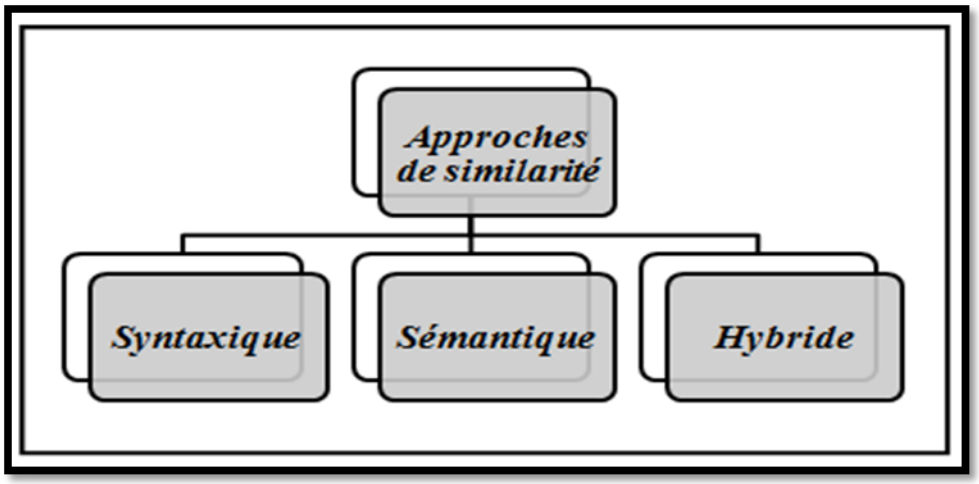


Figure 3 : Les approches de similarité.

5.1. Similarité syntaxique

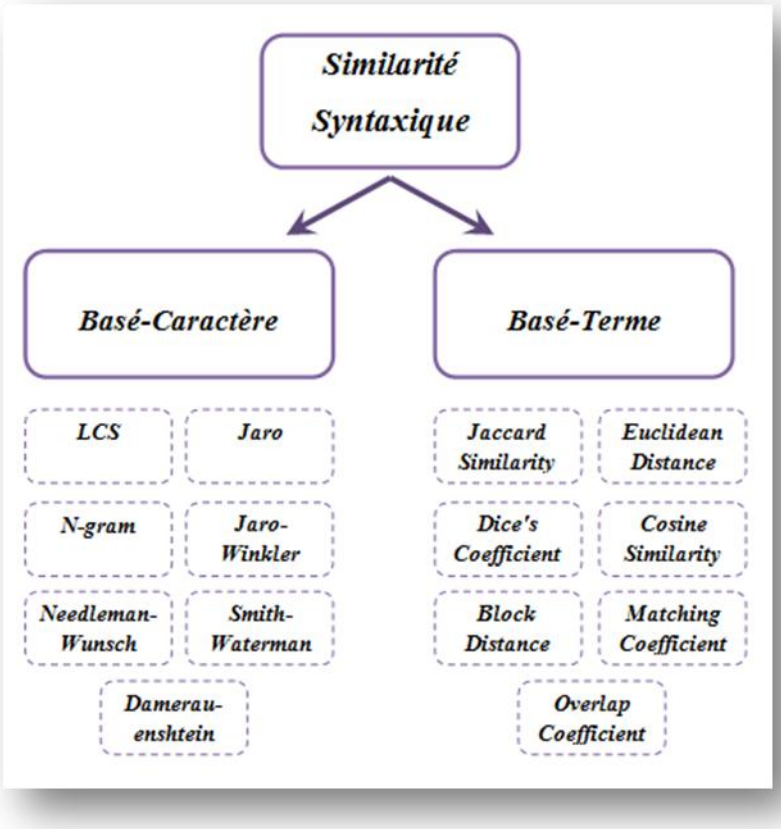


Figure 4 : Mesures de similarité syntaxique.

En mathématiques et en informatique, une mesure permettant de comparer des documents textuels, consiste à comparer des chaines de caractères. C'est une métrique qui mesure la similarité ou la dis-similarité entre deux chaînes de caractères. Par exemple, les chaînes de caractères "voiture" et "voiturier" peuvent être considérées comme similaires, d'autre part, ''voiture'' et ''véhicule'' ne le sont pas. Une telle mesure sur les chaines de caractères fournit une valeur obtenue algorithmiquement [13]. « Figure 4 » montre les mesures de similarité syntaxique.

Parmi ses mesures, sept entre elles sont basées sur des caractères tandis que les autres sont des mesures de distance basée sur les termes. Parmi ces mesures nous présentons :

- **Indice de Jaccard:**

L'indice de Jaccard (ou coefficient de Jaccard) est le rapport entre le cardinal (la taille) de l'intersection des ensembles considérés et le cardinal de l'union des ensembles

[14]. Il permet d'évaluer la similarité entre les ensembles. Soit deux ensembles A et B, l'indice est :

$$J(A,B) = \frac{(A \cap B)}{(A \cup B)}$$

- **Distance euclidienne :**

La distance euclidienne calcule la similarité entre deux documents d1 et d2 comme la distance entre leurs représentations vectorielles ramenées à un seul point [13].

$$Sim_{euclidienne} = \sqrt{\sum_{i=1}^{n}(d1_i - d2_i)}$$

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

- **Cosinus :**

La similarité cosinus est fréquemment utilisée [15] 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. La similarité obtenue $simcosinus(d1, d2) \in [0; 1]$ [13].

$$Sim_{cosinus} = \frac{\overrightarrow{d1}.\overrightarrow{d2}}{\|\overrightarrow{d1}\|.\|\overrightarrow{d2}\|}$$

- **Indice de Dice :**

L'indice de Dice mesure la similarité entre deux documents d1 et d2 en se basant sur le nombre de termes communs à d1 et d2 [13].

$$sim_{dice} = \frac{2\,Nc}{N1 + N2}$$

Où Nc est le nombre de termes communs à d1 et d2, et N1 (resp. N2) est le nombre de termes de d1 (resp. d2).

- **Coefficient de corrélation de Pearson :**

Le coefficient de corrélation de Pearson calcule la similarité entre deux documents d1 et d2 comme le cosinus de l'angle entre leurs représentations vectorielles centrées-réduites. La similarité obtenue $simpearson(d1;\ d2) \in$ [-1; 1] [13].

$$sim_{pearson}(d1, d2) = sim_{cosinus}\ (\ d1 - \overline{d1}\ , d2 - \overline{d2}\ )$$

D'où $\overline{d1}$ (resp.$\overline{d2}$) représente la moyenne de d1 (resp. d2).

5.2. Similarité sémantique :

La similarité sémantique se base sur le sens/signification des mots. Deux concepts sont considérés comme sémantiquement similaires s'il y a une synonymie, hyponymie[1] (Figure 5), antonymie, ou troponymie[2] entre eux [13]. Dans cette approche, des ressources NLP sont indispensables.

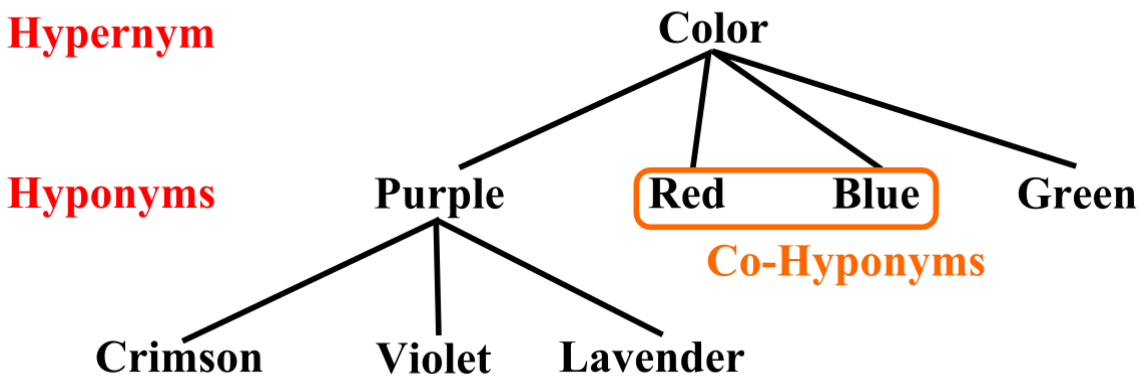


Figure 5 : Exemple d'hyponymie.

Dans la littérature, Nous distinguons deux types de similarité sémantique (voir Figure 6) :

[1] Relation sémantique hiérarchique d'un lexème à un autre selon laquelle l'extension du premier terme est incluse dans l'extension du second. Le premier terme est dit hyponyme de l'autre. Haut-de-forme est un hyponyme de chapeau et chapeau est un hyponyme de coiffure.

[2] Relation sémantique entre deux verbes, l'un décrivant de manière plus précise l'action de l'autre. Le premier verbe est dit troponyme du second.

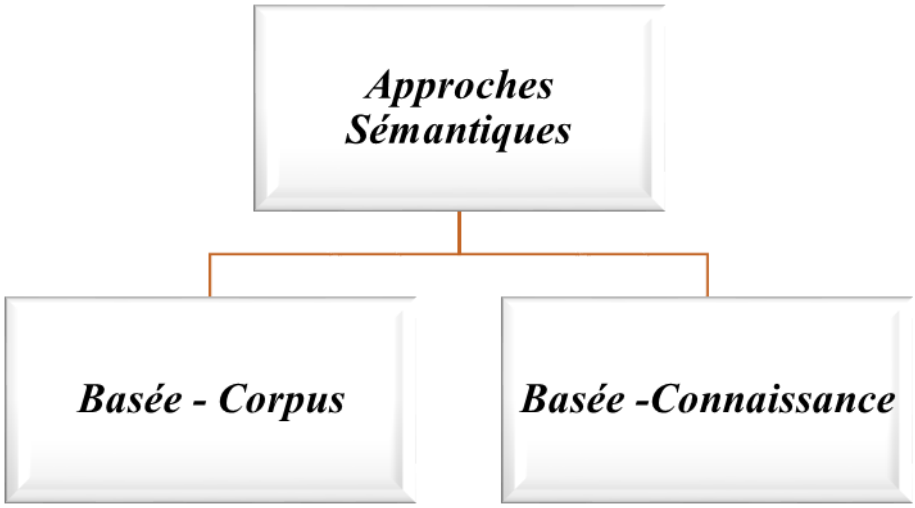


Figure 6 : Mesures de similarité sémantique.

5.2.1. L'approche statistique (corpus-based)

Les mesures basées sur des corpus ne nécessitent pas la compréhension du vocabulaire ou de la grammaire de la langue d'un texte. « Figure 7 » montre les mesures basée-corpus.

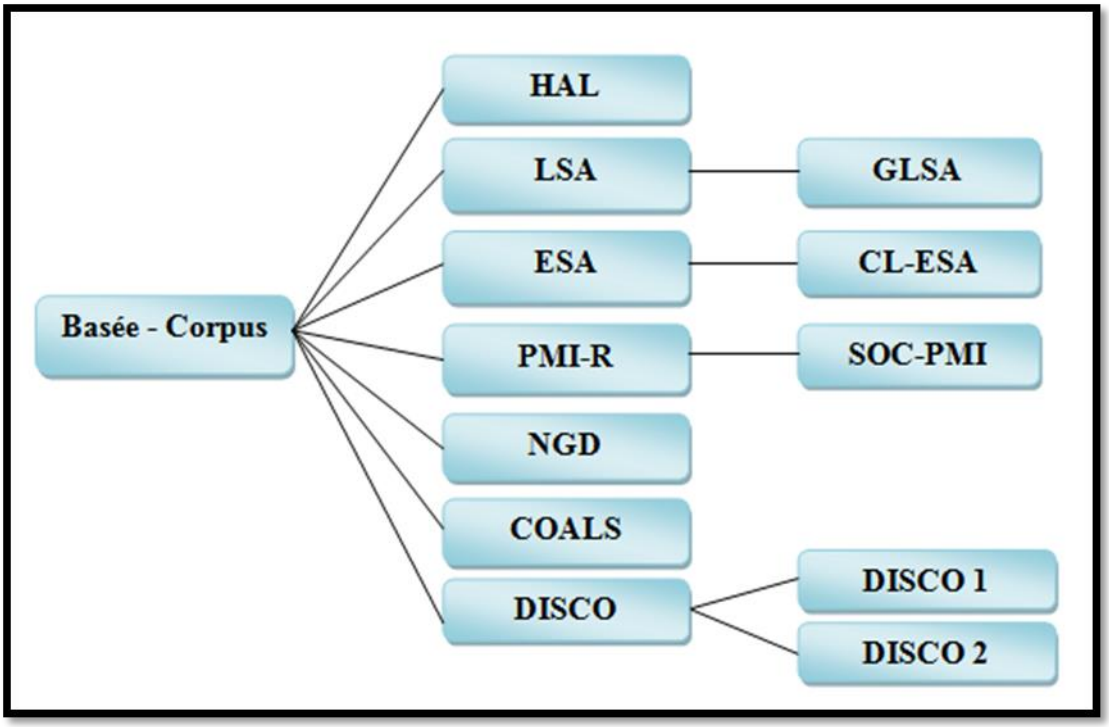


Figure 7 : Les mesures de l'approche Basée-Corpus.

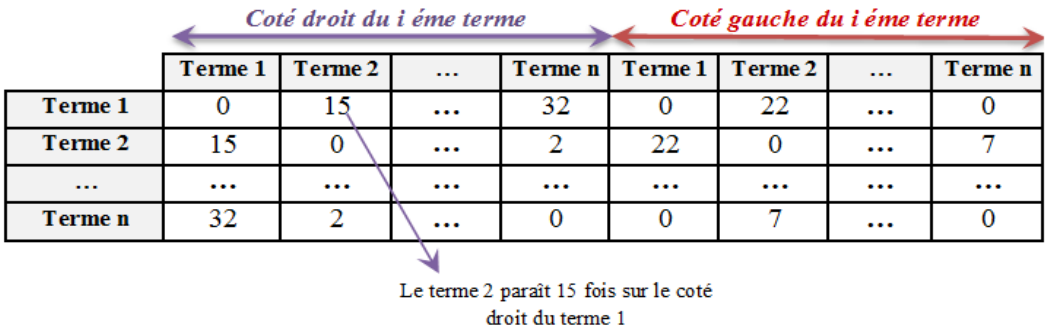
Toutefois, la similarité basée sur le corpus est une mesure de similarité sémantique statistique qui détermine la similarité entre les mots en fonction de l'information obtenue d'un corpus volumineux. Elle consiste a créé un espace sémantique à partir des cooccurrences de mots. Une matrice mot à mot est formée dont chaque élément de la matrice est la force d'association entre le mot représenté par la ligne et le mot représenté par la colonne. Les valeurs matricielles sont accumulées en pondérant la cooccurrence de manière inversement proportionnelle à la distance de focalisation du mot. Les mots voisins les plus proches sont considérés comme reflétant davantage la sémantique du mot cible et sont donc pondérés plus haut c'est-à-dire attribuer une importance [16]. Les mots sont représentés par des vecteurs sémantique d'où la notion d'espace vectoriel VSM (Vector Space Model).

Le modèle VSM vise à convertir l'ensemble des textes du corpus du format textuel en format numérique. Ceci se fait en construisant une matrice des fréquences des termes exhaustifs du corpus (vocabulaire du BOW).

Dans l'approche sémantique, chaque ligne de la matrice représente le vecteur de contexte du terme par rapport à son apparition autour du reste des termes. Cependant, pour calculer la similarité entre deux mots (terme), il faut récupérer leurs vecteurs de contexte et puis appliquer une des mesures syntaxiques citées précédemment sur ces deux vecteurs. Les représentations les plus connues dans cette approche sont comme suit :

- **HAL (Hyperspace Analogue to Language) :**

La méthode construit un espace sémantique à partir des cooccurrences de mots obtenues d'un corpus volumineux. Une matrice mot à mot est alors formée dont chaque élément de la matrice est la force d'association entre le mot représenté par la ligne et le mot représenté par la colonne. Au fur et à mesure que le texte est analysé, un mot de mise au point est placé au début d'une fenêtre de dix mots qui enregistre quels mots voisins sont comptés comme co-occurrents. Les valeurs matricielles sont accumulées en pondérant la cooccurrence de manière inversement proportionnelle à la distance du mot de focalisation. Les mots voisins les plus proches sont considérés comme reflétant davantage la sémantique du mot cible et sont donc pondérés plus haut. HAL enregistre également les informations de classement des mots en traitant différemment la cooccurrence selon que le mot voisin est apparu avant ou après le mot de mise au point [17]. « Figure 8 » montre un exemple de représentation selon HAL.



	Terme 1	Terme 2	…	Terme n	Terme 1	Terme 2	…	Terme n
Terme 1	0	15	…	32	0	22	…	0
Terme 2	15	0	…	2	22	0	…	7
…	…	…	…	…	…	…	…	…
Terme n	32	2	…	0	0	7	…	0

Figure 8 : Exemple de représentation vectorielle selon HAL.

- **LSA (Latent Semantic Analysis) :**

Technique de similitude basée sur le Corpus. LSA suppose que les mots qui ont une signification proche se produiront dans des textes similaires. Une matrice contenant des nombres de mots par paragraphe (les lignes représentent des mots uniques et des colonnes représentent chaque paragraphe) est construite à partir d'un grand texte et une technique mathématique appelée décomposition de valeur singulière (SVD) est utilisée pour réduire le nombre de colonnes tout en préservant la similitude Structure entre les lignes [17]. La différence entre LSA et HAL est que l'espace sémantique construit par HAL se constitue des mots du corpus tandis que celles de LSA contiennent les documents (paragraphes) du corpus. De plus, l'idée d'invention d'HAL est venue à partir de la méthode LSA.

Les mots sont ensuite comparés en prenant le cosinus de l'angle entre les deux vecteurs formés par deux lignes quelconques (calcul de similarité). Un exemple de VSM est présenté dans « Tableau 2 » selon LSA.

Tableau 2 : Exemple de représentation vectorielle selon LSA.

	Document 1	Document 2	…	Document n
Terme 1	15	0	…	50
Terme 2	3	74	…	5
…	…	…	…	…
Terme n	32	2	…	12

- **COALS (Correlated Occurrence Analogue to Lexical Semantic):**

La méthode COALS emploie une stratégie de normalisation qui factorise largement la fréquence lexicale. Le processus commence par la compilation d'une table de

cooccurrence de la même manière que dans HAL, sauf que la distinction gauche / droite est ignoré de sorte qu'il n'y a qu'une seule colonne pour chaque mot (en sommant les cooccurrences gauche et droite). Toutefois, COALS emploi aussi une fenêtre à quatre mots voisin comme les mots du même contexte que le mot en question. Il faut noter que plus le corpus est de petite taille, plus la fenêtre doit avoir une taille importante et vice versa. Un exemple de VSM est présenté dans « Tableau 3 » selon COALS.

Tableau 3 : Exemple de représentation vectorielle selon COALS.

	Terme 1	Terme 2	...	Terme n
Terme 1	0	15	...	32
Terme 2	15	0	...	2
...	...	...	...	...
Terme n	32	2	...	0

5.2.2. L'approche topologique (knowledge-based)

La similarité basée sur la connaissance est l'une des mesures de similarité sémantique qui repose sur l'identification du degré de similitude entre les mots en utilisant des informations dérivées de réseaux sémantiques [17]. Le réseau le plus connue dans le domaine des linguistiques et de mesures de similarité est bien le WordNet. C'est une base de données lexicale développée par des linguistes du laboratoire des sciences cognitives de l'université de Princeton depuis une vingtaine d'années. Son but est de répertorier, classifier et mettre en relation de diverses manières le contenu sémantique et lexical de la langue anglaise [18]. Quelques mesures de la similarité basé-connaissance sont présentées dans la « Figure 9 ».

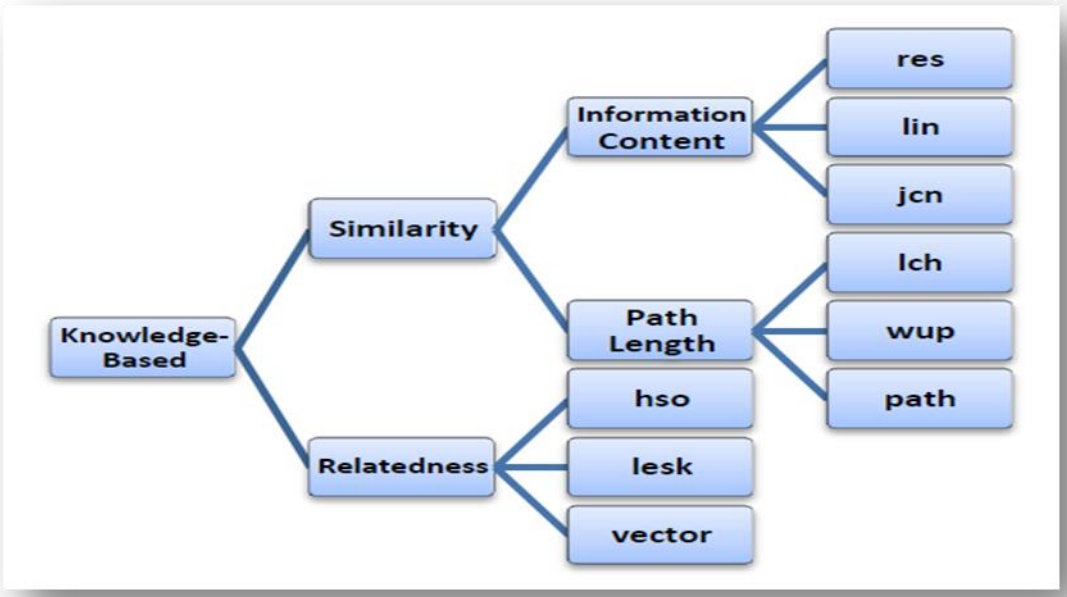


Figure 9 : Les mesures de l'approche Basé-Connaissance [16].

5.3. Similarité hybride

Les résultats des recherches faites par les chercheurs concernant la filière du Machine-Learning et plus précisément celle de la linguistique ont prouvé que l'application d'une combinaison de mesures de similarité donne de supérieures valeurs du facteur de corrélation et donc de meilleurs résultats par rapport à l'application d'une seule mesure de similarité.

Cette combinaison concerne l'arrangement de deux ou plusieurs mesures de similarité de la même approche ou bien un arrangement de mesures d'approches différentes.

6. Revue sur quelques travaux connexes dans le domaine des ASAGS

De nombreux chercheurs s'intéressent au thème de l'évaluation automatique des réponses courtes et ainsi qu'à la similarité. Chaque approche à ses caractéristiques : langue, domaine, ressources nécessaires.

Les travaux que nous menons dans le cadre d'une généralisation d'approche pour la langue arabe nous permettent 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. [19][20]. Les auteurs ont utilisé des mesures de similarité syntaxiques et des mesures basées sur le corpus pour développer leur système de notation à réponse courte. Ils ont testé les mesures sur le dataset (GOMAA dataset) qu'ils ont construit eux-mêmes. Leurs résultats ont montré que les meilleures valeurs de corrélation obtenues en utilisant des mesures syntaxiques ont été obtenues en utilisant respectivement les approches de distance de n-gramme et de distance de Manhattan. Dans la deuxième étape, ils ont mesuré la similarité en utilisant des mesures de similarité basées sur le corpus [21]: 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.

7. Les travaux liés au même Projet entamés durant l'année 2017/2018

Notre travail rentre dans le cadre de la continuité des travaux entamés l'année passée et menés dans l'objectif d'élaboration de méthodologies et d'outils pour l'évaluation automatique des réponses courtes destinée à la langue arabe.

En effet,

[23] présente une synthèse expérimentale des différentes approches de mesures de similarité syntaxique appliquées à des ensembles de données (Data Sets) exprimées dans la langue arabe.

[2] présente une synthèse expérimentale de plusieurs approches de mesures de similarité sémantique appliquée à des ensembles de données (Data Sets) exprimées dans la langue arabe.

[24] donne une synthèse expérimentale de l'utilisation des Word Embedding appliquée à des ensembles de données (DataSets) exprimés dans la langue arabe.

La synthèse a permis d'un côté, de retenir les meilleures approches en vue d'une hybridation de mesures syntaxiques, sémantiques et utilisant les Word Embedding dans le système d'évaluation automatique et d'un autre côté, l'élaboration de plusieurs approches que nous reprenons dans ce travail sans redéveloppement des codes associés.

8. Les travaux liés au même projet dans la langue anglaise

Dans ce contexte on trouve aussi les travaux connexes à la langue anglaise de Mohler [33] celui est a créé le dataset de Mohler sur lequel nous avons appliqué nos approches développées. Dans ces travaux Mohler compare des approches basées corpus de type LSA (latent semantic analysis) à la fois sur un corpus générique et un corpus spécifique de domaine en utilisant Wikipédia comme corpus. Il combine ces approches pour améliorer les résultats de similarité.

Tableau 4 : Résultats des mesures de similarités basés corpus mené par Mohler sur un corpus générique et un corpus spécifique.

Mesure - Corpus	Taille	Corrélation
Trainement sur des corpus génériques		
LSA BNC	566.7 MB	0.4071
LSA Wikipedia	1.8 GB	0.4286
LSA Wikipedia (small)	0.3 MB	0.3518
ESA Wikipedia	1.8 GB	0.4681
Trainement sur un corpus spécifique de domaine		
LSA Wikipedia CS	77.1 MB	0.4628
LSA slides	0.3 MB	0.4146
ESA Wikipedia CS	77.1 MB	0.4385

9. Conclusion

La correction automatique des réponses courtes comme nous venons de voir est un domaine dont les racines sont ancrées et emmêlées avec pleins d'autres domaines tels que le traitement automatique de la langue, le calcul de similarité et l'évaluation automatique. Cet aspect multidisciplinaire offre une multitude d'axes de travail pour traiter le sujet. Dans le prochain chapitre nous exposerons la méthodologie que nous avons suivie dans notre travail et les techniques auxquelles nous avons eu recours.

Chapitre 3 : Système d'évaluation automatique des réponses courtes

Après avoir présenté une vue globale sur l'existant dans le domaine des ASAGS, nous entamons notre approche étape par étape en détaillant chacune d'elles avec des exemples illustratifs pour mieux visualiser la démarche. Ceci est fait en se basant sur les connaissances acquises de l'étude précédente (chapitre 2).

1. Méthodologie

Dans ce qui suit, nous allons résumer les grandes lignes (visualisées au niveau de la Figure 10) par les points suivants :

- ✓ **Etape 1** : Création de l'espace sémantique

1.1. Acquisition du corpus :

Dans cette phase, il s'agit de collecter plusieurs corpus arabes, et effectuer une analyse afin de choisir les plus performants en termes de qualité, de volume et de domaine d'étude.

1.2. Prétraitement du corpus :

Pour traiter le corpus, nous appliquons des techniques du traitement du langage naturel dont nous analysons plusieurs stemmer dédiés pour le traitement de la langue arabe et par la suite choisir le plus convenable.

1.3. Traitement du corpus :

Dans cette étape, nous générons notre espace sémantique selon la démarche détaillée ultérieurement.

1.4. Post traitement du corpus :

Ici nous générons des valeurs représentant la spécificité[1] ou l'importance des mots du corpus.

- ✓ **Etape 2 :** Modèle de calcul de similarité

  Nous présentons les deux modèles adaptés pour le calcul de similarité sémantique qui sont : le modèle somme-vecteurs, le modèle calcul-matriciel et l'hybridation des deux modèles.

[1] La spécificité d'un terme est la quantité d'informations spécifiques au domaine contenues dans le terme. Certains termes contiennent une quantité relativement importante d'informations de domaine et d'autres une quantité relativement faible d'informations de domaine [40].

✓ **Etape 3 :** Passage au score

Après avoir calculé la similarité, nous passons au score/note automatique en utilisant le classifieur non supervisé k-means.

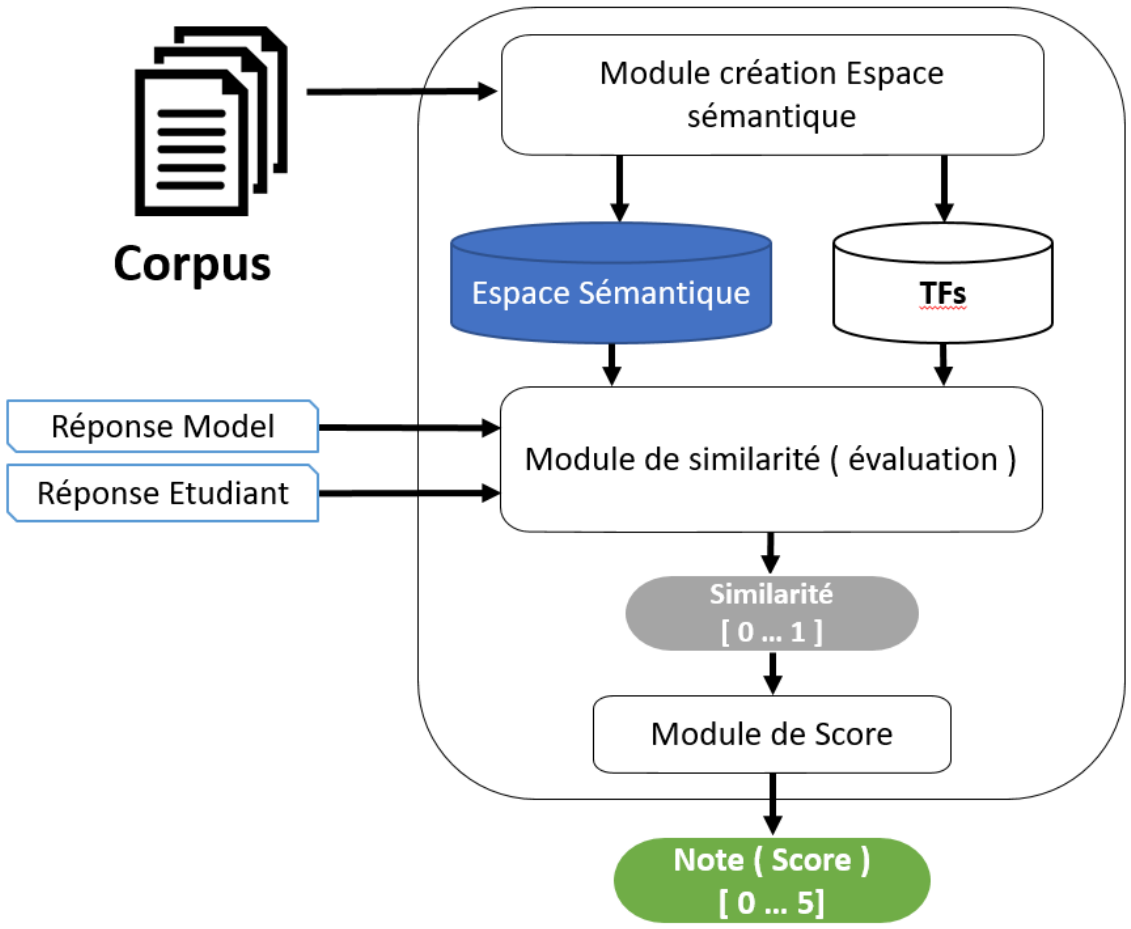


Figure 10 : Schéma des étapes principales du système d'évaluation

2. Construction de l'espace sémantique

Dans cette partie nous présenterons les différentes phases de création de notre espace sémantique « voir Figure 11 » qui sera par la suite utilisé pour extraire les vecteurs de contexte et enfin calculer la similarité entre le couple de réponses (RM, RE), Tout d'abord, l'acquisition et l'analyse du corpus afin de choisir celui qui convient. Ensuite, la phase du prétraitement qui consiste à préparer le corpus pour tout usage. La phase du traitement du corpus comprend l'application de l'une des approches basée corpus cité précédemment dans le but de la construction de

l'espace sémantique. En fin, le post traitement du corpus qui prend en considération l'importance des mots dans le corpus.

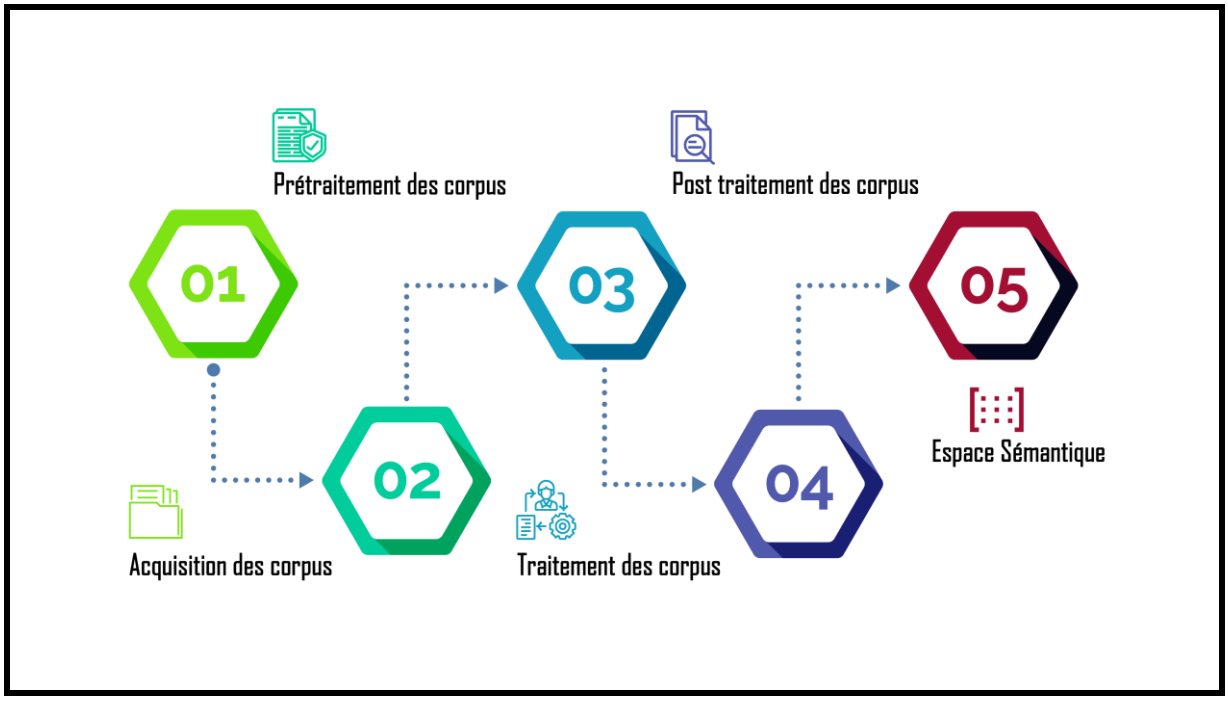


Figure 11 : Phases de création de l'espace sémantique.

2.1. Acquisition des corpus

Les corpus que nous nous sommes procurés sont tous issus du net. Malgré la disponibilité des outils linguistiques pour la langue anglaise, il nous a été plus difficile de trouver des corpus de qualité moyenne et acceptable car les corpus de meilleurs qualité sont accessibles de manière confidentielle ou éducative pour les universités des pays développés ou payante. Toutefois, nous avons eu l'occasion d'analyser et de traiter trois corpus disponibles gratuitement en ligne consacrés à la recherche dans la description ci-dessous :

- ***Gutenberg (Avril 2014)***[1] ***:***

Il s'agit d'une collection de 3 036 livres en anglais écrits par 142 auteurs. Cette collection est un petit sous-ensemble du corpus du projet « Gutenberg electronic text archive », qui contient 25,000 livres électroniques gratuit héberger a http///www.gutenberg.org/. Tous les livres ont été nettoyés manuellement pour

[1] URL : https://web.eecs.umich.edu/~lahiri/gutenberg_dataset.html

supprimer les métadonnées, les informations de licence et les notes des transcripteurs, autant que possible, Ce corpus est créer par Lahiri et Shibamouli en Avril 2014, Et a été mis à jour par Matthew D.Scholefield, Qui a corriger certains problèmes relier avec la version originale, Et a mis à disposition la nouvelle version le 17 Aout 2018.

- ***BBC corpus Anglais (2004 - 2005)***[1] ***:***

Se compose de 2225 documents du site d'information de BBC correspondant à des articles dans cinq domaines d'actualité qui sont classées par dossiers Business, Entertainment, Politics, Sport, Tech.

- ***Leipzig Collection de corpus***[2] ***:***

La collection de corpus de Leipzig présente des corpus dans différentes langues en utilisant le même format et des sources comparables. Toutes les données sont disponibles sous forme de fichiers texte brut et peuvent être importées dans une base de données MySQL à l'aide du script d'importation fourni. Ils sont destinés à la fois à une utilisation scientifique par des linguistes de corpus et à des applications telles que les programmes d'extraction de connaissances.

Les corpus sont de format identique, de taille et de contenu similaires. Ils contiennent des phrases choisies au hasard dans la langue du corpus et sont disponibles dans des tailles allant de 10 000 phrases à 1 million de phrases. Les sources sont des textes de journaux ou des textes recueillis au hasard sur le Web. Les textes sont scindés en phrases. Les documents ne contenant pas de phrases et les langues étrangères ont été supprimés.

Pour notre cas et dû au manque de matériel nous avons choisi les corpus anglais de taille minimale qui est de 10 000 phrases, un totale de 15 document extrait depuis Wikipédia et des sites d'actualité de la langue anglaise.

Dans le tableau suivant « Tableau 5 » on va donner les statistiques relatifs à ces différents corpus comme taille sur le disque et le nombres des mots...etc.

[1] URL : http://mlg.ucd.ie/datasets/bbc.html
[2] URL : http://wortschatz.uni-leipzig.de/en/download/

Tableau 5 : Statistiques relatives aux différents corpus acquis.

	Gutenberg	BBC	Leipzig
Taille sur le Disque	**1.13 GO**	**4.80 MB**	**16.6 MB**
Nombre de documents	**3 036**	**2225**	**15**
Totale de mots du corpus	**210904266**	**854490**	**2941421**
Nombre de mots unique	**32893819 par Fichier**	**27 757**	**19511**

2.2. Prétraitement du corpus

Pour cette tâche, nous travaillons sur tous les corpus décrits précédemment. Elle se compose de deux autres sous-tâches:

2.2.1. Techniques TAL adoptées:

Ceci est fait par les étapes suivantes :

- Supprimer tous les lettres sauf l'alphabet (A, B…Z, a, b.…z), Cette étapes de résume beaucoup de traitement qui suivants comme :
  - Suppression des virgules, apostrophes et symboles de tous types.
  - Suppression des numéros.
  - Suppression des diacritiques[1] (accents aigu, grave…), Sachant que c'est ce qui se fait dans la littérature sauf pour la langue anglaise nous avons pas eu besoin de ce traitement.
  - Suppression des lettres arabes ainsi que les lettre des autres lagunages qui n'utilisant pas l'alphabet comme la langue chinoise…etc.
- Réduire toutes les lettres en miniscule.
- Réduire les successions des espaces en un espace simple.
- Application d'une étape de stemming.

[1] **Diacritique : Outre les 26 lettres de l'alphabet, l'écriture standard du français met en jeu cinq signes diacritiques (accents aigu, grave et circonflexe, tréma et cédille) et les deux ligatures <æ> et <œ>. S'ajoutent ainsi 16 lettres : *é, è, ê, ë, à, â, î, ï, ô, ù, û, ü, ÿ, æ, œ* et ç.**

- Tokeniser chaque document et alors chaque corpus en considérant un espace simple comme séparateur.

2.2.2. L'approche par Stemming adoptée :

En effet, il est très difficile de mettre en œuvre les mécanismes d'évaluation automatique pour n'importe quelle langue du a la forme de base de n'importe quelle phrase ou texte, surtout pour un corpus qui représente une quantité importante de textes… Car il n'existe aucun système jusqu'à présent qui est en mesure de traiter un tels cas en prennent en compte la grande diversité des mots ainsi que la multiplicité des formes que peut prendre ce mot, c'est pour cela qu'une étape de stemming est généralement nécessaire pour réduire les différentes formes d'un mot et aussi pour notre cas réduire la taille des corpus et par la suite réduire la taille des espaces sémantique.

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 [25].

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

- 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.
- 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.

Pour la mise en œuvre de l'approche du stemming dans notre travail, nous avons recherché parmi plusieurs stemmers existants dans la langue anglaise (disponibles en ligne ou téléchargeables). Nous avons testé les différents stemmers sur beaucoup de couples de réponses. Ci-dessous une liste des stemmers trouvés et testés :

- SnowBall stemmer[1].
- Porter stemmer[2].
- Lancaster stemmer[3].

SnowBall stemmer supporte plusieurs langues à la fois parmi ces langues : l'arab, Danish, Dutch, English (standard, Porter), Finnish, French, German, Greek, Hungarian, Irish, Italian, Lithuanian, Nepali, Norwegian, Portuguese, Romanian, Russian, Spanish, Swedish, Tamil, Turkish.

Le site web [26] rend disponibles les trois stemmers (présentés ci-dessus) Avec une documentation d'utilisation et code source fournit.

Nous avons testé les trois stemmer sur les réponses modèles du dataset de Mohler.

« Tableau 6 » représente les trois stemmers testés sur la une réponse modèle du dataset de Mohler comme un échantillon.

Les quelques stemmers qui existent ne présentent pas une évaluation de la précision des résultats obtenus. L'avis d'un expert en langue anglaise 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 :

- SnowBall Stemmer pour un léger stemming.
- Porter stemmer pour un léger stemming.
- Lancaster pour un lourd Stemming

[1] https://pypi.org/project/snowballstemmer/
[2] https://www.nltk.org/_modules/nltk/stem/porter.html
[3] https://www.nltk.org/_modules/nltk/stem/lancaster.html

Tableau 6 : Echantillon des tests sur les trois stemmers.

	Mot	SnowBall	Porter	Lancaster
it provides a limited proof of concept to verify with the client before .actually programming the whole application	it	it	it	it
	provides	provid	provide	provid
	a	a	a	a
	limited	limit	limit	limit
	proof	proof	proof	proof
	of	of	of	of
	concept	concept	concept	conceiv
	to	to	to	to
	verify	verifi	verifi	ver
	with	with	with	with
	the	the	the	the
	client	client	client	cli
	before	before	befor	bef
	actually	actual	actual	act
	programming	program	program	program
	the	the	the	the
	whole	whole	whole	whol
	application	applic	applic	apply

L'approche par stemming a été adoptée aussi bien pour les réponses que pour les corpus. Un stemmer doit avoir une valeur de précision ou erreur qui signifie l'estimation de son exactitude. La plupart des stemmers n'ont pas de documentation disponible en ligne ce qui n'est pas pratique et rend la tâche plus difficile (choisir le stemmer approprié).

2.3. Traitement du corpus

Dans notre démarche, l'approche de similarité statistique « Corpus-Based » est adoptée à cause du manque de ressources dont souffre beaucoup de langue surtout la langue arabe, par mis ces ressources on trouve les dictionnaires et les

lexicons dont l'approche topologique « knowledge-based » impose. Nous allons adopter le modèle BOW qui ne prend pas en considération l'ordre des mots. Notre approche se base sur le concept disant que les mots qui sont sémantiquement liés se trouvent dans le même contexte. Pour cela cette approche repose sur la notion du voisinage (mots voisins). Cependant la présence d'un corpus qui sera manipulé à l'aide du modèle BOW est indispensable. En outre, une fenêtre avec une taille prédéfinie est importante pour concrétiser la notion du voisinage.

Nous avons utilisé le corpus « BBC », le corpus « Gutenberg » et le corpus « Leipzig » ainsi qu'une fenêtre de taille 4. La « Figure 13 » montre le processus effectué afin de générer nos trois espaces sémantiques à l'aide des trois corpus. Pour mieux voir les choses, nous détaillons ce processus ci-dessous :

a) *Les corpus sont acquis et prétraités selon l'étape du « Prétraitement du corpus ».*

b) Construction de la matrice des cooccurrences :

Ou matrice de fréquences/poids. Cette étape signifie la transformation des données textuelles au sein du corpus en données numériques sous forme d'une matrice. Chaque case représente la somme des poids de l'apparence du $i^{éme}$ terme (mot) avec le $j^{éme}$ terme. Cette somme est calculée par rapport à l'emplacement des termes du corpus autour du $i^{éme}$ terme (effectuer le même calcul pour chaque occurrence du $i^{éme}$ terme). Ici nous entamons la notion de voisinage sous forme de ce que l'on appelle une **taille de fenêtre**. Concrètement, si deux mots d'un texte sont en **voisinage** de taille inférieure à celle de la fenêtre, ils sont comptés comme **co-occurrents**. Dans notre approche nous fixons la taille de la fenêtre à 4 c'est-à-dire considérer que les quatre voisins adjacents au $i^{éme}$ terme (ou qui se trouve autours ce terme) des deux côtés (gauche et droite) et ignorer le reste des mots. Pour illustrer ce fonctionnement, nous prenons le texte suivant comme une partie du corpus :

« the web consist of two parts frontend and backend »

Nous considérons le terme « frontend » comme le $i^{éme}$ terme. Le partitionnement des poids est présenté dans la « Figure 12 ». Les voisins du terme

«frontend » auront un poids de 4, les voisins des voisins du terme auront un poids de 3 et ainsi de suite.

The	Web	consist	of	two	parts	frontend	and	backend
0	0	1	2	3	4	0	4	3

Figure 12 : Exemple du fonctionnement de la fenêtre de taille 4.

c) Construction de la matrice des corrélations :

Après avoir construit la matrice des cooccurrences, nous appliquons une stratégie de normalisation. Nous allons alors construire une nouvelle matrice en appliquant la formule ci-dessous prise de l'algorithme COALS pour chaque case/élément de la matrice des cooccurrences.

$$\begin{cases} w'_{a,b} = \dfrac{T w_{a,b} - \sum_j w_{a,j} . \sum_i w_{i,b}}{\left(\sum_j w_{a,j} . (T - \sum_j w_{a,j}) . \sum_i w_{i,b} . (T - \sum_i w_{i,b})\right)^{1/2}} \\ T = \sum_i \sum_j w_{i,j} \end{cases}$$

D'où :

- **a** et **b** sont les deux termes de la matrice de cooccurrences (ligne et colonne).
- $w_{a,b}$ est l'élément de la matrice de cooccurrences du terme a et b.
- i est l'indice des lignes tandis que **j** est celui des colonnes.
- $\sum_j w_{a,j}$ est la somme des colonnes de la ligne du terme a.
- $\sum_i w_{i,b}$ est la somme des lignes de la colonne du terme b.
- $T = \sum_i \sum_j w_{i,j}$ est la somme de tous les éléments de la matrice de cooccurrences.

Pour un large corpus, les valeurs de corrélations sont petites, alors, il est rare que la valeur de corrélation dépasse le **0.01.** De plus, la majorité des corrélations sont négatives (81.8%) [16].

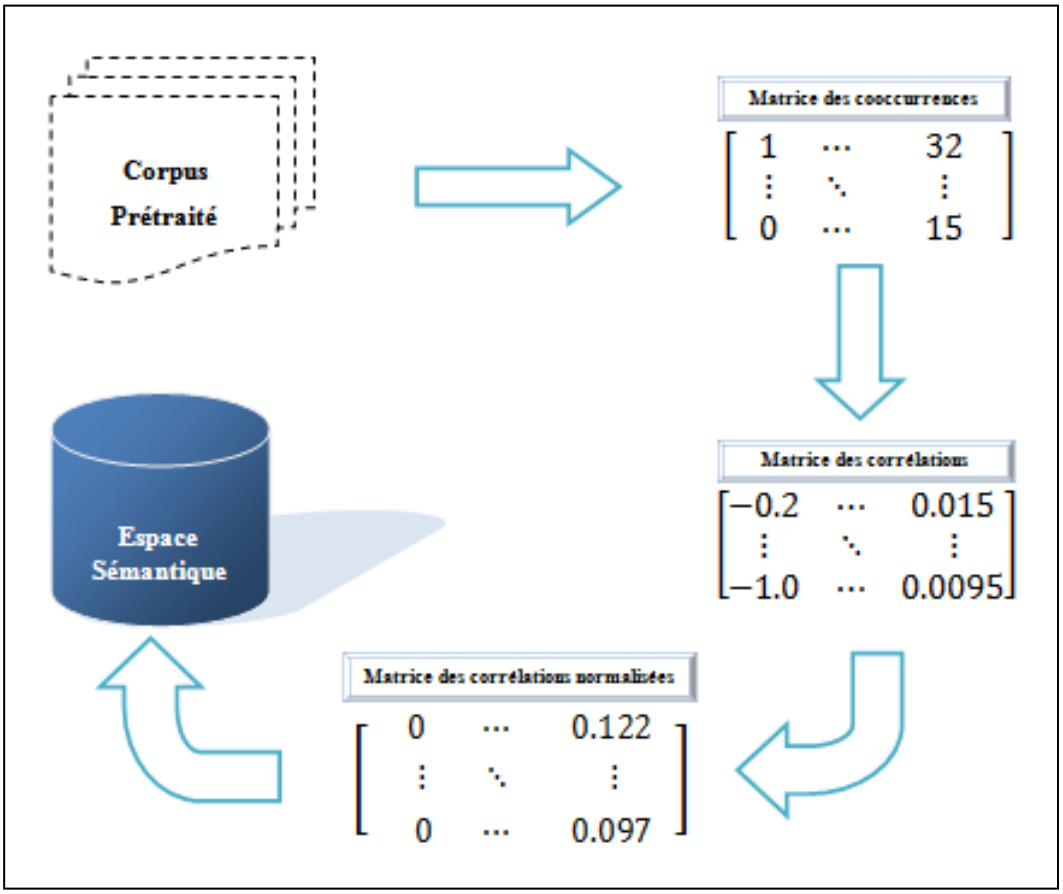


Figure 13 : Exemple illustrant la phase du traitement du corpus.

d) Construction de la matrice de corrélation normalisée :

Les corrélations négatives transportent très peu d'information, alors, nous effectuons encore une autre normalisation. Les valeurs négatives sont normalisées à 0 tandis que les valeurs positives prennent leurs racine carré afin d'amplifier l'importance des nombreuses petites valeurs par rapport aux grandes valeurs [16].

Cette étape représente la dernière étape de création de l'espace sémantique ou proprement dit : les vecteurs du contexte de chaque mot du corpus en entrée. Il est à noter que l'espace sémantique est généré qu'une seule fois. Il est donc stocké pour une utilisation ultérieure.

➢ **Exemple illustratif :**

Nous considérons le texte dans la « Figure 14 » comme un corpus. Dans cet exemple, nous montrons la procédure sans stem et avec stem.

Delegation is another language feature that can be used as an alternative to inheritance.

Figure 14. Exemple de corpus.

✓ **Cas sans stem**

1) Construction de la matrice des cooccurrences :

Tableau 7 : Matrice des cooccurrences dans le cas sans stem.

	Delegation	is	another	language	feature	that	can	be	used	as	an	alternative	to	inheritance
Delegation	0	4	3	2	1	0	0	0	0	0	0	0	0	0
is	4	0	4	3	2	1	0	0	0	0	0	0	0	0
another	3	4	0	4	3	2	1	0	0	0	0	0	0	0
language	2	3	4	0	4	3	2	1	0	0	0	0	0	0
feature	1	2	3	4	0	4	3	2	1	0	0	0	0	0
that	0	1	2	3	4	0	4	3	2	1	0	0	0	0
can	0	0	1	2	3	4	0	4	3	2	1	0	0	0
be	0	0	0	1	2	3	4	0	4	3	2	1	0	0
used	0	0	0	0	1	2	3	4	0	4	3	2	1	0
as	0	0	0	0	0	1	2	3	4	0	4	3	2	1
an	0	0	0	0	0	0	1	2	3	4	0	4	3	2
alternative	0	0	0	0	0	0	0	1	2	3	4	0	4	3
to	0	0	0	0	0	0	0	0	1	2	3	4	0	4
inheritance	0	0	0	0	0	0	0	0	0	1	2	3	4	0

2) Matrice des corrélations :

Tableau 8 : Matrice des corrélations dans le cas sans stem.

	Delegation	Is	another	language	feature	that	can	be	used	as	an	alternative	to	inheritance
Delegation	-0.043	0.304	0.186	0.093	0.013	-0.063	-0.063	-0.063	-0.063	-0.063	-0.061	-0.058	-0.052	-0.043
is	0.304	-0.062	0.208	0.125	0.054	-0.011	-0.075	-0.075	-0.075	-0.075	-0.073	-0.069	-0.062	-0.052
another	0.186	0.208	-0.076	0.16	0.093	0.034	-0.024	-0.083	-0.083	-0.083	-0.081	-0.076	-0.069	-0.058
language	0.093	0.125	0.16	-0.086	0.135	0.079	0.023	-0.033	-0.088	-0.088	-0.086	-0.081	-0.073	-0.061
feature	0.013	0.054	0.093	0.135	-0.091	0.127	0.073	0.018	-0.036	-0.091	-0.088	-0.083	-0.075	-0.063
that	-0.063	-0.011	0.034	0.079	0.127	-0.091	0.127	0.073	0.018	-0.036	-0.088	-0.083	-0.075	-0.063
can	-0.063	-0.075	-0.024	0.023	0.073	0.127	-0.091	0.127	0.073	0.018	-0.033	-0.083	-0.075	-0.063
be	-0.063	-0.075	-0.083	-0.033	0.018	0.073	0.127	-0.091	0.127	0.073	0.023	-0.024	-0.075	-0.063
used	-0.063	-0.075	-0.083	-0.088	-0.036	0.018	0.073	0.127	-0.091	0.127	0.079	0.034	-0.011	-0.063
as	-0.063	-0.075	-0.083	-0.088	-0.091	-0.036	0.018	0.073	0.127	-0.091	0.135	0.093	0.054	0.013
an	-0.061	-0.073	-0.081	-0.086	-0.088	-0.088	-0.033	0.023	0.079	0.135	-0.086	0.16	0.125	0.093
alternative	-0.058	-0.069	-0.076	-0.081	-0.083	-0.083	-0.083	-0.024	0.034	0.093	0.16	-0.076	0.208	0.186
to	-0.052	-0.062	-0.069	-0.073	-0.075	-0.075	-0.075	-0.075	-0.011	0.054	0.125	0.208	-0.062	0.304
inheritance	-0.043	-0.052	-0.058	-0.061	-0.063	-0.063	-0.063	-0.063	-0.063	0.013	0.093	0.186	0.304	-0.043

3) Construction de la matrice des corrélations normalisées :

Tableau 9 : Matrice des corrélations normalisées dans le cas sans stem

	Delegation	is	another	language	feature	that	can	be	used	as	an	alternative	to	inheritance
Delegation	0	0.551	0.431	0.305	0.114	0	0	0	0	0	0	0	0	0
is	0.551	0	0.456	0.354	0.232	0	0	0	0	0	0	0	0	0
another	0.431	0.456	0	0.4	0.305	0.184	0	0	0	0	0	0	0	0
language	0.305	0.354	0.4	0	0.367	0.281	0.152	0	0	0	0	0	0	0
feature	0.114	0.232	0.305	0.367	0	0.356	0.27	0.134	0	0	0	0	0	0
that	0	0	0.184	0.281	0.356	0	0.356	0.27	0.134	0	0	0	0	0
can	0	0	0	0.152	0.27	0.356	0	0.356	0.27	0.134	0	0	0	0
be	0	0	0	0	0.134	0.27	0.356	0	0.356	0.27	0.152	0	0	0
used	0	0	0	0	0	0.134	0.27	0.356	0	0.356	0.281	0.184	0	0
as	0	0	0	0	0	0	0.134	0.27	0.356	0	0.367	0.305	0.232	0.114
an	0	0	0	0	0	0	0	0.152	0.281	0.367	0	0.4	0.354	0.305
alternative	0	0	0	0	0	0	0	0	0.184	0.305	0.4	0	0.456	0.431
to	0	0	0	0	0	0	0	0	0	0.232	0.354	0.456	0	0.551
inheritance	0	0	0	0	0	0	0	0	0	0.114	0.305	0.431	0.551	0

✓ **Cas avec stem :**

deleg anoth languag featur use altern inherit

Figure 15 : Exemple de corpus avec stem.

1) Construction de la matrice des cooccurrences :

Tableau 10 : Matrice des cooccurrences dans le cas avec stem.

	deleg	anoth	languag	featur	use	altern	inherit
Deleg	0	0	2	4	0	3	1
Anoth	0	0	3	1	4	2	4
languag	2	3	0	3	2	4	4
Featur	4	1	3	0	0	4	2
Use	0	4	2	0	0	1	3
Altern	3	2	4	4	1	0	3
Inherit	1	4	4	2	3	3	0

2) Matrice des corrélations :

Tableau 11 : Matrice des corrélations dans le cas avec stem.

	deleg	anoth	languag	featur	use	altern	inherit
Deleg	-0.111	-0.134	0.017	0.25	-0.111	0.115	-0.062
Anoth	-0.134	-0.163	0.036	-0.08	0.25	-0.029	0.124
languag	0.017	0.036	-0.22	0.036	0.017	0.065	0.065
Featur	0.25	-0.08	0.036	-0.163	-0.134	0.124	-0.029
Use	-0.111	0.25	0.017	-0.134	-0.111	-0.062	0.115
Altern	0.115	-0.029	0.065	0.124	-0.062	-0.205	0.008
Inherit	-0.062	0.124	0.065	-0.029	0.115	0.008	-0.205

3) Construction de la matrice des corrélations normalisées :

Tableau 12 : Matrice des corrélations normalisées dans le cas avec stem.

	deleg	anoth	languag	featur	use	altern	inherit
Deleg	0	0	0.13	0.5	0	0.339	0
Anoth	0	0	0.19	0	0.5	0	0.352
languag	0.13	0.19	0	0.19	0.13	0.255	0.255
Featur	0.5	0	0.19	0	0	0.352	0
Use	0	0.5	0.13	0	0	0	0.339
Altern	0.339	0	0.255	0.352	0	0	0.089
Inherit	0	0.352	0.255	0	0.339	0.089	0

2.4. Post traitement du corpus

En effet, la similarité sémantique est fondée sur l'idée disant que les termes similaires se trouvent fréquemment dans le même contexte. Néanmoins, dans un contexte, les termes n'ont pas tous la même importance. Par conséquent, il est plus judicieux de pondérer l'impact des termes avant l'application d'une mesure de similarité. La fréquence d'apparition d'un terme dans un document est un bon indicateur de l'importance de ce terme. Une fonction de pondération attribue à chaque terme de chaque document une valeur. Cette valeur (ou poids) est calculée en tenant compte de deux grands critères [27]:

La force (capacité) locale du terme dans le document (c'est-à-dire mesurer l'importance du terme dans le document dans lequel il apparait).

La force globale (c'est-à-dire mesurer l'importance du terme dans tout le corpus).

- ***TF (Term Frequency) :*** Ou la fréquence du terme dans un document.

Cette pondération repose sur le calcul de la fréquence du terme dans le document (le nombre de fois que le terme apparait dans le document). Plus un terme est fréquent dans un document plus il est important dans la description de ce document.

$$TF = \frac{\text{Nombre d'occurence du mot dans le document}}{\text{Nombre totale de mots dans le document}}$$

- ***TF – IDF* [28] *:***

Le **TF-IDF** (de l'anglais *term frequency-inverse document frequency*) est une méthode de pondération souvent utilisée en recherche d'information et en particulier dans la fouille de textes. Cette mesure statistique permet d'évaluer l'importance d'un terme contenu dans un document, relativement à une collection ou un corpus. Le poids augmente proportionnellement au nombre d'occurrences du mot dans le document. Il varie également en fonction de la fréquence du mot dans le corpus. Des variantes de la formule originale sont souvent utilisées dans des moteurs de recherche pour apprécier la pertinence d'un document en fonction des critères de recherche de l'utilisateur.

- ***IDF (inverse document frequency): ou frequence inverse de document***

La fréquence inverse de document (*inverse document frequency*) est une mesure de l'importance du terme dans l'ensemble du corpus. Dans le schéma TF-IDF, elle vise à donner un poids plus important aux termes les moins fréquents, considérés comme plus discriminants. Elle consiste à calculer le logarithme (en base 10 ou en base 2[1]) de l'inverse de la proportion de documents du corpus qui contiennent le terme :

$$IDF = \frac{|\,D\,|}{|\{\, dj : ti \in dj \,\}|}$$

| D | : nombre total de documents dans le corpus

|{ dj : t*i* ∈ dj }| : nombre de documents où le terme apparaît

- ***TF – IDF :***

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

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \cdot \mathrm{idf}_i$$

- ***TF $_{log}$:***

Le nombre de fois que le mot apparait dans le corpus [35]

$$TF_{log} = -\log(\frac{TF_{count}}{n})$$

TF_{count} : le nombre d'occurrence du mot dans le corpus et n est le nombre total de mot dans le corpus

- ***TF Min – Max :***

$$TF_{min-max} = \frac{TF_{log}}{max(TF_{log})}$$

Après avoir construit notre espace sémantique, nous sauvegardons ce dernier dans un fichier « .Txt ». Chaque ligne de ce fichier représente un vecteur d'un mot. Les éléments d'un vecteur du mot sont séparés par des espaces simples. Chaque élément est sous format chaine de caractères « String ».

3. Modèles du calcul de similarité sémantique entre deux réponses courtes

Après avoir construit l'espace sémantique, nous entamons la notion du calcul de similarité sémantique. Dans notre travail, nous avons considéré deux niveaux du calcul de similarité [2] :

- Similarité entre phrases : nous avons réalisé ce concept par un modèle que nous nommons « somme vecteurs ».
- Similarité mot-à-mot : nous avons concrétisé ce concept par un modèle de calcul matriciel.
- Une hybridation des deux modèles précédents.

3.1. Le modèle somme-vecteurs (SV)

Nous détaillons les étapes du fonctionnement de la SV représentées au niveau de la « Figure 16 » :

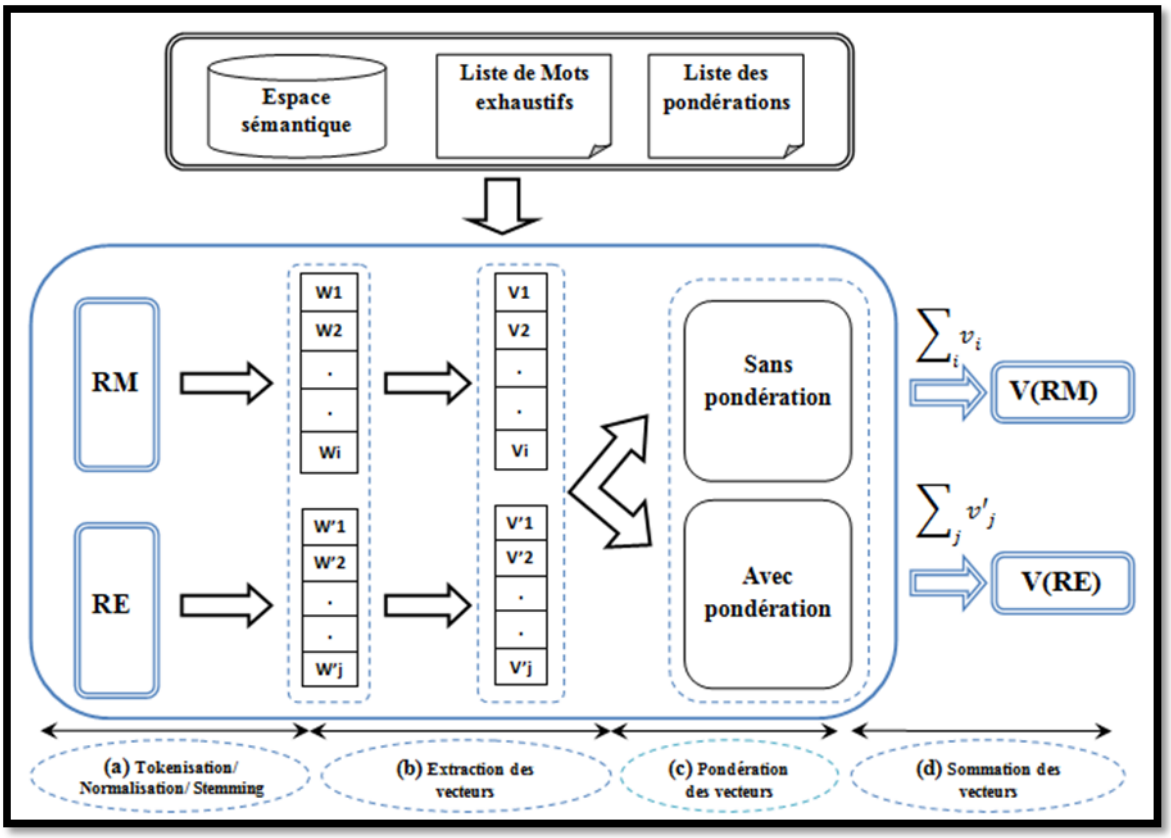


Figure 16 : Vue globale sur le fonctionnement du modèle SV.

Une étape de prétraitement (normalisation, stemming…) est effectuée sur les deux réponses (RM et RE). Par la suite, chaque réponse est transformée en vecteur en appliquant une tokenisation. Deux vecteurs de mots résultent de cette étape.

Maintenant, les mots de chaque réponse sont remplacés par leurs vecteurs de contexte. Cela est fait en récupérant le vecteur de contexte du mot à partir de l'espace sémantique construit et stocké précédemment. Un vecteur de vecteurs résulte à ce stade pour chaque réponse.

Ensuite, nous allons récupérer des valeurs de pondération de chaque mot. Ici nous distinguons deux cas : avec une pondération et sans aucune pondération. Dans le cas de pondération, chaque vecteur de contexte du mot i est multiplié par la valeur de pondération qui représente l'importance du mot i.

Chaque vecteur de vecteurs de contexte passe par une étape de sommation. Cela est obtenue en sommant les vecteurs de contexte de chaque réponse mot par mot ou proprement dit case par case selon la formule suivante :

$$V(R) = \sum_{j=1}^{i} v_j$$

D'où:

R est la réponse dont nous calculons son vecteur (réponse modèle/étudiant).

i est le nombre de termes/mots constituant la réponse R après l'étape du prétraitement.

v_j est le vecteur de contexte du $j^{éme}$ terme de la réponse R.

Nous arrivons au stade du calcul de la valeur de similarité entre les deux réponses. Nous appliquons la mesure **Cosine** sur les vecteurs V(RM) et V(RE). Une valeur entre 0 et 1 est obtenue et qui représente la valeur de similarité entre les deux réponses.

Exemple illustratif :

Nous allons appliquer ce modèle sur un couple de réponses du dataset de Mohler :

a) Les tâches de cette étape sont représentées dans le tableau « Tableau 13 ».

Tableau 13 : Etape a).

	Réponse Modèle	**Réponse Etudiant**
Etat initial	At the main function.	they beging to excute at main
Normalisation	At main function	beging execute at main
Stemming	at main function	begin execute at main
Tokenisation	**VRM=** [at] [main] [function]	**VRE=** [begin] [execute] [at] [main]

b) Récupération des vecteurs de contexte des mots depuis l'espace sémantique :

VRM = v(at) + v(main) + v(function)

VRE = v(begin) + v(execute) + v(at) + v(main)

Un exemple d'un vecteur de contexte :

v(main) = [0.01, 0.12, 0.035, 0, 0,…]

c) Nous supposons que les valeurs des pondérations sont représentées dans le tableau « Tableau 14 ».

Tableau 14 : Valeurs des pondérations

Terme	at	main	function	begin	execute
Poids	0.33	0.78	0.86	0.56	0.70

VRM = 0.33 * v(at) + 0.78 * v(main) + 0.86 * v(function)

VRE = 0.56 * v(begin) + 0.70 * v(execute) + 0.33 * v(at) +0.78 * v(main)

d) Cette tâche est présentée dans le tableau « Tableau 15 ».

Tableau 15 : Etape d).

Réponse Modèle	Réponse Etudiant
V(at) = [0.11, 0.24, 0.45, 0, 0, …] V(main) = [0.11, 0.24, 0.475, …] V(function) = [0.11, 0.56, 0.74, …]	V(begin) = [0.11, 0.4, 0.45, …] V(execute) = [0.16, 0.28, 0.87, …] V(function) = [0.11, 0.56, 0.74, …] V(at) = [0.11, 0.24, 0.45, 0, 0, …] V(main) = [0.11, 0.24, 0.475, …]
VRM = [0.80, 0.825, 1.41, ….]	VRE = [0.96, 1.31, 1.41, 0.80, 0.825….]

Maintenant nous calculons la similarité entre les deux réponses :

Le résultat sans pondération :

$$Sim(RM, RE) = Sim_{Cosin}(Vm, Ve) = 0.70$$

Le résultat avec pondération :

$$Sim(RM, RE) = Sim_{Cosin}(Vm, Ve) = 0.75$$

3.2. Le modèle calcul-matriciel (CM)

Contrairement au modèle précédent, nous construisons une matrice dont les termes d'une réponse représentent les lignes de la matrice tandis que les termes de l'autre réponse représentent les colonnes. Par la suite, chaque élément de la matrice signifie la valeur de similarité entre le vecteur de contexte du $i^{éme}$ terme (réponse 1) et celui du $j^{éme}$ terme (réponse 2).Nous nous sommes inspirés du travail de Islam et Inkpen [29] Ce modèle prend en considération l'ordre entre mot. Ci-dessous nous présentons les étapes détaillées :

a) Une étape de prétraitement (normalisation, stemming, …) est effectuée sur les deux réponses RM, RE. Par la suite, chaque réponse est transformée en vecteur en appliquant une tokenisation. Deux vecteurs de mots de taille m et n résultent de cette étape (la même étape que celle du modèle SV).

b) Suppression des termes en commun « ou encore : les mots qui matchent notant *δ* le nombre de ces mots » de chaque réponse en gardant trace de l'ordre entre les termes de chaque réponse. Nous aurons les deux réponses REM, RMM de taille (m- *δ) et (n- δ).*

c) Si m- *δ = 0 ou n- δ*, = 0 ou les deux sont nuls, nous passons à l'étape h), sinon nous passons à l'étape d).

d) Construction de la matrice sémantique MSem de taille (m- *δ*)*(n- *δ*) dont *la condition* consiste que **(m ≤ n)** c'est-à-dire les lignes correspondent aux mots de la réponse (RMM ou REM) qui a une longueur inferieur ou égale par rapport à l'autre (les colonnes correspondent aux mots de la longue réponse). Chaque élément de MSem présente la corrélation entre les vecteurs de contexte de chaque couple de mots (a, b) tel que a ∈.lignes et b ∈ colonnes, comme le montre la formule suivante (prise de [16]) :

$$MSem(a,b) = \frac{\sum(a_i - \bar{a})(b_i - \bar{b})}{\left(\sum(a_i - \bar{a})^2 \ \sum(b_i - \bar{b})^2\right)^{1/2}}$$

D'où $\bar{a}\ (resp.\bar{b})$ est la moyenne du vecteur du contexte du mot $a\ (resp.\ b)$.

Dans le cas d'une combinaison avec d'autres approches (syntaxique ou word-Embedding), effectuer les tâches e), f). Sinon passer directement à l'étape g) et considérer MSem comme matrice combinée M.

e) Construction de la matrice syntaxique MSyn, ou de la matrice des word-Embedding MWE, ou les deux matrices de taille (m- *δ*) * (n- *δ*). Il est à noter que la condition de l'étape d) doit être vérifiée.

f) Construction d'une matrice combinée M de taille (m- *δ*) * (n- *δ*) des deux matrices (« MSem,MSyn » ou « MSem,MWE ») ou des trois matrices (MSem,MSyn,MWE) précédentes en effectuant la moyenne comme suit :

$$\begin{cases} M \leftarrow 0.5 * M_1 + 0.5 * M_2 \\ M \leftarrow (M_1 + M_2 + M_3)/3 \end{cases}$$

g) Nous sauvegardons la valeur maximale M (i, j) de la matrice M dans une liste ρ (*ρ* ← *ρ*∪ *M (i, j)*), puis nous supprimons tous les éléments correspondants à sa colonne j et sa ligne i de M. Refaire cette étape jusqu'à la satisfaction de l'une des conditions suivantes :

- *La somme des éléments de la matrice M est nulle.*
- *La matrice M est vide.*

h) Calculer le score de similarité noté So. Considérons le couple de réponse, RM et RE ont respectivement m et n jetons (tokens), c'est-à-dire RM = p1, p2, . . ., pm et RE = r1, r2, . . ., rn et n ≥ m. Sinon, nous changeons RM et RE. Nous comptons le nombre de pi (étant δ) pour lequel pi = rj, pour tout p ∈ RM et pour tout r∈ RE. Autrement dit, il y a des jetons δ dans RM exactement correspond à RE, où δ ≤ m. Nous enlevons tous les jetons δ de RM et les place dans X et RE dans Y, dans le même ordre que dans les réponses. Donc, X = {x1, x2, . . ., xδ} et Y = {y1, y2, . . ., yô}. Nous remplaçons X en assignant un numéro d'index unique pour chaque jeton dans X, de 1 à δ, c'est-à-dire X = {1, 2, . . ., δ}. Sur la base de ces numéros d'index uniques pour chaque jeton dans X, nous remplaçons également Y où X = Y. La formule suivante est utilisée pour mesurer la similitude de l'ordre des mots communs du couple de réponses:

$$So = 1 - \frac{|x1 - y1| + |x2 - y2| + \cdots + |x\delta - y\delta|}{|x1 - x\delta| + |x2 - x\delta - 1| + \cdots + |x\delta - x1|}$$

i) Nous sommons tous les éléments de ρ et nous y ajoutons δ (1 - wf + wf So) pour obtenir un score total, où So est le score de similarité d'ordre de mots communs et wf est le poids d'ordre des mots communs où $wf \in [0, 0.5]$. Nous multiplions ce score total par la moyenne harmonique réciproque de m et n pour obtenir un score de similarité équilibré entre 0 et 1, inclusivement.

$$S(P,R) = \frac{(\delta(1-wf+wfS_0)+\sum_{i=1}^{|\rho|}\rho i)\times(m+n)}{2nm} \quad (11)$$

- L'équation (11) est simplifiée en équation (12) si :
  - L'importance de l'information syntaxique est ignorée en mettant wf à 0.
  - La valeur de similarité d'ordre de mots communs So=1.

$$S(P,R) = \frac{(\delta+\sum_{i=1}^{|\rho|}\rho i)\times(m+n)}{2mn} \quad (12)$$

Enfin, pour mesurer la similarité d'une paire de mots, ce modèle est ensuite utilisé. Nous déterminons la similarité sémantique et syntaxique/WordEmbedding et les combinons en un score de similarité. Ces scores sont ensuite utilisés pour calculer la similarité globale de deux segments de texte. Le score de similarité pour chaque paire de mots est pondéré en tenant compte de la spécificité de ses mots. Afin d'améliorer les résultats de notre modèle, nous avons exprimé la spécificité des mots en utilisant la pondération (présenté dans la section 3.2.4) de fréquence terme normalisé, qui donne un poids plus élevé aux paires avec des mots plus spécifiques.

Exemple illustratif

Ci-dessous (dans Tableau 16) le couple du dataset de Mohler (Le deuxième couple de réponses) sur lequel nous appliquons l'étape a) et b):

Tableau 16. Etape a) et b) du modèle CM.

Etape		Réponse Modèle	Réponse étudiant
aa)	**Initiale**	At the main function.	they beging to excute at main
	Prétraitement	at main function	begin execute at main
bb)	**Suppression des δ**	function	Begin execute
	La liste δ	at, main	

Voici la matrice générée avec le modèle CM dans le cas d'une matrice sémantique :

M1 =

	begin	execute	at	main
at	0.11	0.06	1	0
main	0.56	0.23	0.01	1
function	0.80	0.70	0.34	0.85

Par la suite, nous appliquons une itération pour récupérer le maximum de la matrice et supprimer la ligne et la colonne de cet élément :

M2 =

	begin	execute
function	0.80	0.70

➢ **Le calcul de similarité :**

La liste des max : [0.80]

Application de la formule (12) détaillée précédemment.

Similarité = 0.31

3.3. Hybridation

Dans cette phase, nous visons à concrétiser la notion d'hybridation entre les mesures afin d'avoir de meilleurs résultats. Pour ce faire, nous allons adopter une combinaison interne ainsi qu'une combinaison externe.

Pour la combinaison interne, nous combinons les résultats des deux modèles de calcul précédents en deux manières :

La moyenne :

Dans ce type, nous prenons la moyenne des deux valeurs de similarité (calculé par SV et CM) entre les deux réponses. Les résultats seront présentés dans le chapitre suivant.

Le maximum :

Contrairement à la moyenne, dans ce cas, nous prenons la valeur maximum entre les deux valeurs de similarité. De même, les résultats dans le prochain chapitre.

Pour la combinaison externe, nous allons maintenant combiner nos modèles de base avec les méthodes de deux autres binômes travaillant sur la même thématique :

Hennich Adel Nassim et Hannoufi Mohamed Hamza [30].

Abdallah Amina et Guerroudja khadidja dont le thème est « Mesures de similarité syntaxique pour un système d'évaluation automatique des réponses courtes : Application à la langue arabe ».

4. Passage au score

Après avoir mesuré la valeur de similarité entre les deux réponses, nous allons effectuer un passage au score de la valeur de similarité vers une note ou un score selon le barème donné. Pour ce faire, nous adoptons la technique suivante :

- Classifieur k-means [31](Ou k-moyenne)

C'est une technique de classification non supervisée. Elle consiste à avoir un ensemble de données comme entrée (input) et elle donne en sortie ces données classifiées dans K classes. C'est un algorithme très populaire et facile dans sa mise en œuvre. L'inconvénient de cet algorithme réside dans l'exigence de la détermination du nombre de classe comme entrée. Néanmoins, dans notre approche, ceci présente un avantage car nous fixons dès le départ le nombre des classes selon le barème donné.

Dans le prochain chapitre, les couples de réponses dataset utilisés dans l'évaluation de notre approche sont noté sure 5 points. Nous fixons K à 11 pour avoir 11 classes

de note (0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5) afin de prendre en compte la subjectivité du processus d'évaluation.

5. Conclusion

Au cours de ce chapitre, nous venons de présenter les principes de l'approche d'évaluation adaptée à l'anglais. Ce dernier est un ensemble de plusieurs modules intégrés pour effectuer une évaluation automatique de réponses courtes en langue anglaise. Ce processus nécessite un prétraitement du corpus dont il sera utilisé pour créer notre VSM « espace sémantique » ainsi que pour la génération des pondérations dans le but de tester son impact. Le processus suivant s'en charge du calcul de similarité qui sera convertis en score par la suite. Pour se faire, nous nous sommes inspirés d'autres travaux pour élaborer les modèles proposés (SV, CM, Hybridation).

Après avoir conçue notre système, des métriques d'évaluation ainsi que des dataset seront utilisés pour déterminer le degré de son efficacité et sa fiabilité. Les résultats des tests ainsi que ceux des hybridations seront présentés dans le chapitre suivant.

Chapitre 4 : Résultats expérimentaux et évaluation

Dans ce chapitre, nous montrons les expériences et les résultats des tests réalisés afin d'estimer la qualité de notre système et sa performance comparée aux autres systèmes qui existent, ainsi nous discutons des résultats obtenus et de la généricité de l'approche.

Le « Tableau 17 » présente la liste des abréviations employée au cours de ce chapitre.

Tableau 17 : Listes des Abréviations.

Abréviation	Expression
DS	Dataset
CM	Modèle Calcul Matriciel
SV	Modèle Somme-Vecteurs
WE	Word-Embeddings
RMSE	Root Mean Squared Error
CP	Coefficient de Pearson
WF	Similarité d'ordre

1. Démarche expérimentale

1.1. Outils développés

Pour l'approche que nous avons adaptée, nous avons repris le code déjà développé pour la langue arabe. La première difficulté est celle de pouvoir comprendre le code car il n'était pas suffisamment commenté. La deuxième difficulté réside sur l'aspect du code qui est non modulaire ce qui rend plus difficile son adaptation. En dehors des algorithmes basiques liée à l'approche, nous avons fait les acquisitions nécessaires à l'anglais et bien sûr développer les outils correspondants.

Notre application englobe un système d'évaluation automatique des réponses courtes ainsi que d'autres outils du traitement du langage naturel que nous avons dû développer ou réadapter. Ce système se compose de différents modules intégrés entre eux présentés ci-dessous :

- **Outils de Normalisation et stemming** :

Dans cette partie, nous donnons la possibilité à l'utilisateur de percevoir le changement effectué sur un texte donné en entrée. L'utilisateur peut choisir l'opération du stemming en spécifiant le stemmer (Snowball, Tashaphyne ou Porter), et les opérations de normalisation en spécifiant les paramètres à effectuer parmi une liste de paramètres (suppression des numéros, suppression des diacritiques…). En fin du traitement, l'utilisateur a la possibilité de sauvegarder le résultat. « voir Figure 17».

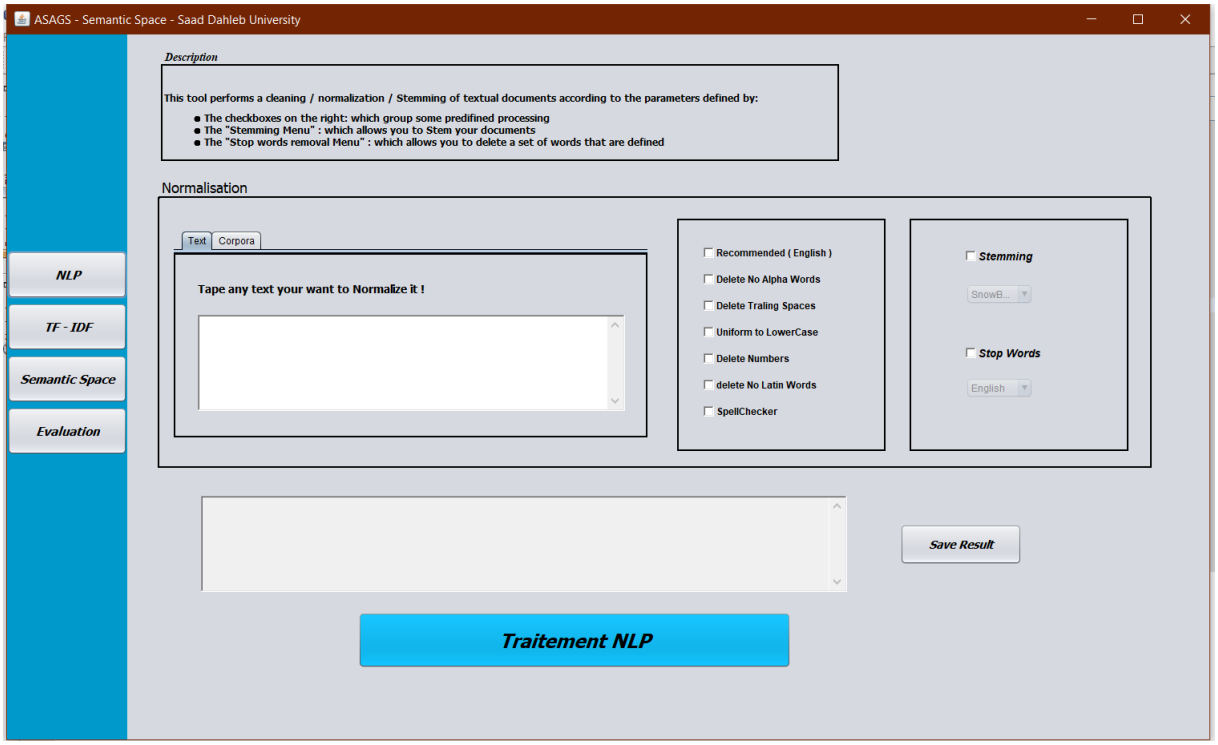


Figure 17 : Outil de Normalisation et de Stemming.

- **Weigthning : Calcul des poids de mots**

Cette partie permet de faire différents traitements sur un corpus choisi par l'utilisateur :

1) Les pondérations (TF, IDF, TF-IDF,),

2) Fréquence des mots et leur nombre total,

3) Liste des mots exhaustifs.

L'utilisateur doit donner en entrés le corpus sous forme d'un dossier contenant que des fichiers texte, il doit davantage indiquer le chemin du dossier de sortie, et choisir un ou bien plusieurs traitement « voir Figure 18 ».

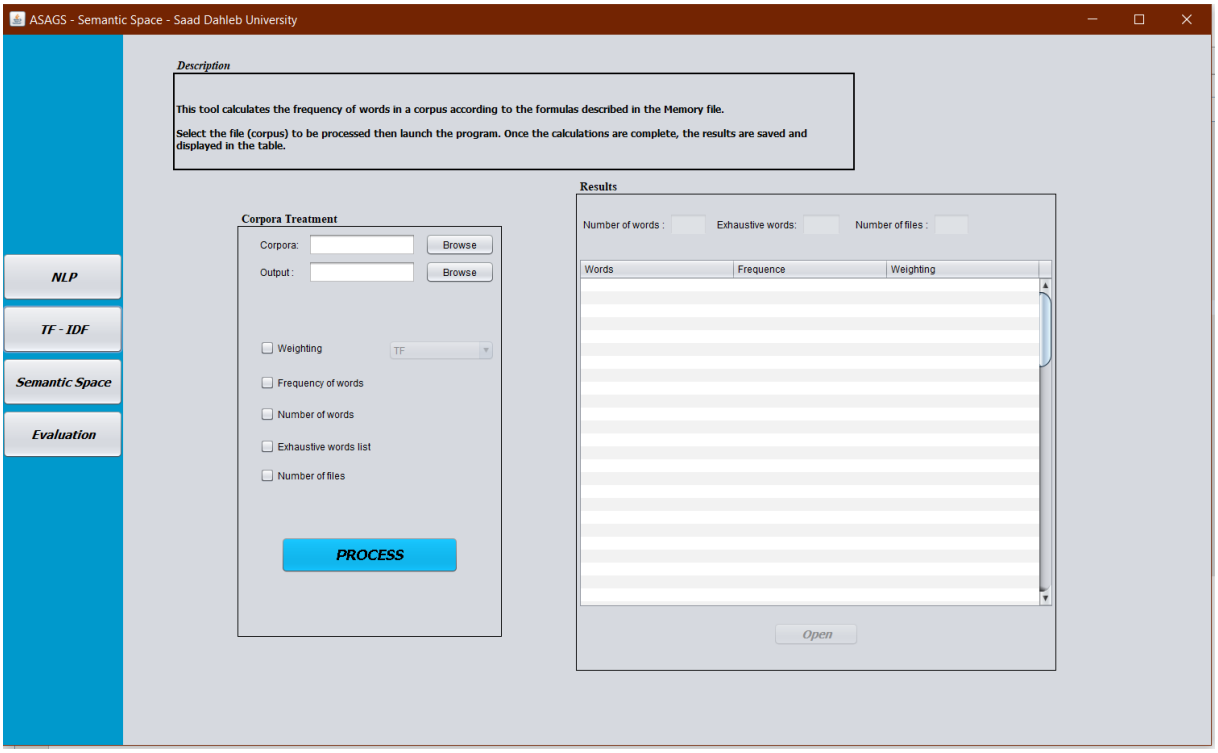


Figure 18 : Outil NLP.

- **Outil de création de l'espace sémantique**

Cet outil permet de calculer l'espace sémantique d'un corpus donné par l'utilisateur en entrée et L'utilisateur doit donner en entrée le chemin du dossier de sortie où les résultats vont être sauvegardés « voir Figure 19».

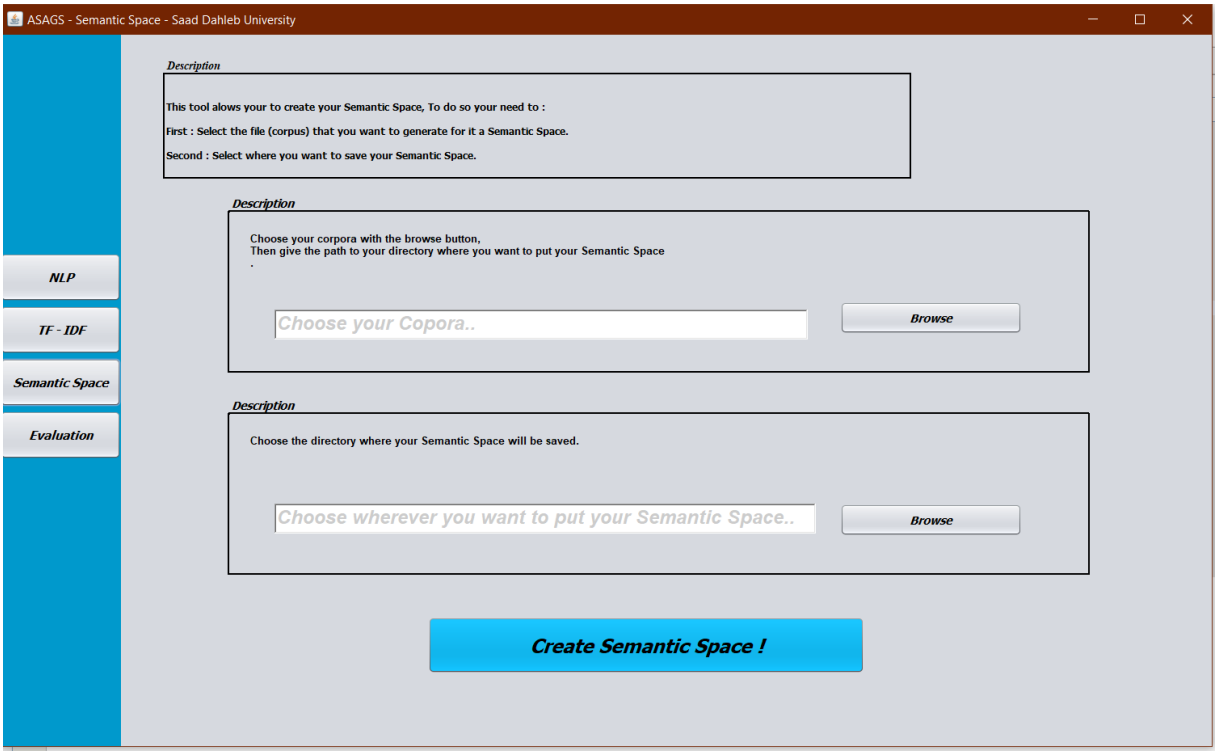


Figure 19 : Outil de création de l'espace sémantique

- **Outils d'évaluation automatique des réponses courte**

Ce module nous permet de calculer la similarité entre deux réponses courtes ainsi qu'un DS avec nos modèles et leurs combinaisons détaillées dans le chapitre précédent. Il permet aux utilisateurs de choisir de faire le calcul avec des ressources internes (Espace sémantique et pondération déjà calculés) ou bien de choisir des ressources externes « voir Figure 20 ».

Du coté DS, l'utilisateur a la possibilité d'évaluer un DS de son choix c'est-à dire généré les notes automatiques, il doit avoir en entrée trois fichiers textes, l'un présente les réponses modèles et l'autre les réponses des étudiants pour le calcul des similarités entre les différentes réponses. Et enfin les notes (moyenne des 2 annotateurs) manuelle du DS pour calculer le coefficient de Pearson et l'erreur quadratique pour l'évaluation des approches utilisées… Enfin il faut choisir un dossier de sortie ou les résultats seront enregistrés.

Tandis que pour le coté des deux réponses, l'utilisateur doit saisir deux réponses (RM et RE) et choisir le reste des paramètres. Le résultat s'affiche directement dans l'interface.

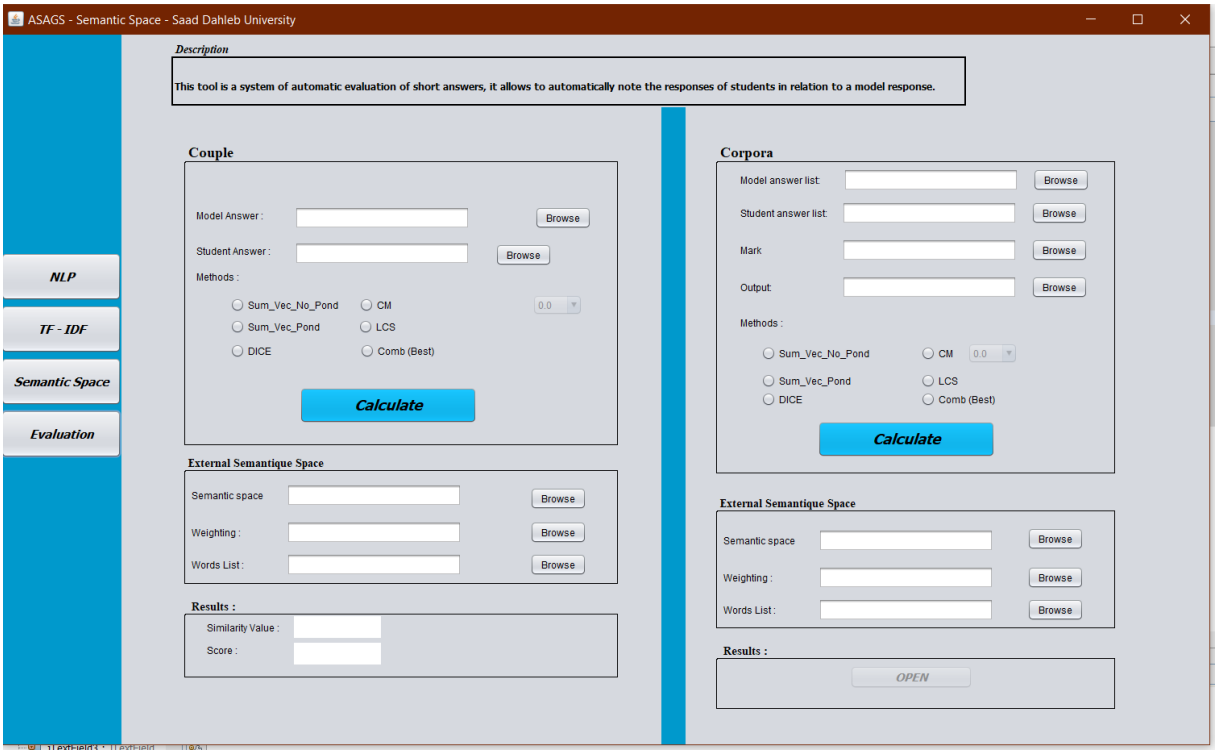


Figure 20 : Outils d'évaluation automatique des réponses courtes.

1.2. Ressources matérielles et logicielles utilisées lors du développement

Nous avons rencontré un problème matériel dans la phase principale de notre approche « Traitement du corpus ». Ce dernier n'a pas abouti sa fin à cause de l'insuffisance de la mémoire de nos PC portables (8 Go de RAM). Pour cela, nous avons eu la chance de travailler sur un serveur à distance « voir Figure 21 » fournit par l'université de Bouira et c'est la configuration minimale requise pour notre travail.

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

Figure 21 : Référence et caractéristiques du serveur.

Tout fois lors du traitement des corpus nous avons été confrontés à d'autres problèmes d'installations des package et l'utilisation du serveur par plusieurs

personnes à la fois. Nous avons alors du augmenté la capacité de notre RAM jusqu'à 16 Go de Ram. Ce qui nous a permis de poursuivre le travail pour avoir des résultats assez acceptables. Car l'idéal est d'avoir plus de Ram, et plus de puissance de calcul. Voire un serveur multi-core fonctionnant simultanément avec mémoires caches dédiées ou partagées.

Nous avons travaillé avec le langage de programmation *Python 3.7* sur notre machine personnelle et *Python 2.4 et 3.6* sur le serveur à distance ainsi que le langage *Java*.

Pour les interfaces, planifiées en java, nous avons utilisé l'environnement de développement *Netbeans 8.0.2* et *PyCharm Community 2018 3.4* pour *Python*.

1.3. Jeux de données (Datasets) et métriques d'évaluation

1.3.1. Datasets[32]

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 anglaise. 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 anglaise il existe un seul DS [33] largement cité dans l'évaluation des ASAGS en langue anglaise qui est disponible. Dans toute la suite nous allons considérer ce DS et l'identifier par « Mohler DS[17] » [33].

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

1.3.2. Mohler DataSet [33]

Le Data set comporte 3 sujets chacun d'eux porte sur un cours d'initiation à l'informatique à l'Université de North Texas. Les données sont au format texte brut. Chaque sujet comprend la question, la réponse de l'enseignant et un ensemble de réponses de l'élève, avec les notes moyennes de deux annotateurs incluses. Il a été demandé aux deux annotateurs de noter l'exactitude sur une échelle entière

[17] URL : http://web.eecs.umich.edu/~mihalcea/downloads/ShortAnswerGrading_v1.0.tar.gz

comprise entre 0 et 5. La corrélation entre les annotateurs sur l'ensemble des données était de 0,6443 En utilisant le coefficient de Pearson. Un identifiant d'étudiant unique est également fourni.

L'ensemble des données (Mohler DS) comprend 3 sujets d'examen chacun d'eux contient 7 questions, pour le 1ér sujets on a 29 réponses étudiants par question et bien sûr le même nombre pour les réponses modèles avec un totale de 203 couple (RE, RM), pour le 2ém sujet nous avons 30 réponses par question avec un total de 210 couple de réponses, pour le 3ém sujet nous avons 31 réponses par question avec un totale de 217 couple de réponses, donc le totale de réponses du DS est de 630 réponses.

Tableau 18. Un échantillon du DS Mohler

ID Question	Question	Réponse Modèle (RM)	Réponse Etudiant (RE)	Note Manuelle
Question 1 Assign 1	What is the role of a prototype program in problem solving?	To simulate the behaviour of portions of the desired software product	It tests the main function of the program while leaving out the finer details. 	2
			it simulates the behavior of portions of the desired software product	5
Question 1 Assign 2	What is typically included in a class definition?	Data members (attributes) and member functions.	An object and data.	2
			Data and functions	4.5
Question 1 Assign 3	What does a function signature include?	The name of the function and the types of the parameters.	input parameters and return type	3
			The portion of the function prototyp tha has the function name and the arguments but NOT the return type.	5

1.3.3. Métriques d'Evaluation [32] :

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 DS et les travaux connexes qui ont utilisé ces mêmes DS. La corrélation de Pearson ([34]**)** est la métrique la plus fréquemment utilisée par les recherches dans ce domaine. C'est le cas aussi des différents DS 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)[35]) pour quantifier la différence (ou le décalage) entre le résultat(score) obtenu par le système et celui obtenu par l'expert humain.

1.3.4. Coefficient de Pearson (CP) [34] :

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 CP. 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 :

Tableau 19. Signification des valeurs de 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

Plus le coefficient est proche des valeurs extrêmes -1 et 1, plus la corrélation linéaire entre les variables est forte ; nous employons 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 DS (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.

1.3.5. Erreur quadratique RMSE (Root Mean Squared Error (RMSE)[35]) :

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. Nous divisons ensuite le résultat obtenu par le nombre total d'erreurs individuelles, puis nous en prenons 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 des approches correspond à trouver la meilleure minimisation de l'erreur quadratique avec une maximisation du coefficient de corrélation.

2. Résultats et discussion

Dans la suite, nous présentons les résultats obtenus dans plusieurs perspectives :

- Dimensionnalité de l'espace sémantique et impact du stemming,
- Les résultats des deux modèles de similarité proposés et de leurs variantes (en termes de pondération),
- Hybridation des deux modèles,
- Hybridation avec les mesures syntaxiques et avec celles développées pour les WE,
- Impact de l'utilisation de corpus de domaine spécifique,
- Evaluation par rapport aux résultats obtenus par les travaux connexes.

3. Dimensionnalité de l'espace sémantique et impact du stemming

Généralement, les vecteurs de mots avec plus de quelques centaines de dimensions sont peu pratiques. En mathématiques, la méthode algébrique de décomposition en valeur singulière (Singular Value Decomposition (SVD) [36]) est un outil important pour factoriser des matrices rectangulaires complexes afin de réduire la taille. Cependant, cette approche est très coûteuse en termes de consommation de mémoire et peut être impraticable en particulier pour les grands corpus, où les tailles d'espace initiales peuvent être importantes. Idéalement, pour notre système proposé, l'algorithme SVD serait calculé à l'aide des vecteurs matriciels complets de l'espace sémantique construit. Cependant, ceci est difficile en termes de calcul et est inutile. De bons résultats peuvent être obtenus en utilisant plusieurs milliers de mots parmi les plus fréquents. La dimensionnalité de l'espace sémantique peut être réduite en augmentant la limite des mots peu fréquents après le retrait des Stopwords. Nous utilisons également le stemming pour minimiser les classes de mots et ensuite la dimensionnalité.

Dans le « Tableau 20 », nous explorons 3 espaces sémantiques construits avec des dimensions allant jusqu'à 20 662. Ajoutant par la suite un corpus spécifique de domaine de taille réduite et avec de meilleurs résultats pour voir la différence et mettre l'accent sur l'importance de la qualité du corpus choisi.
En augmentant la dimensionnalité pour les 3 espaces sémantiques, le système de base calculé avec le modèle de la somme vectorielle sans pondération en utilisant Mohler DS, présente le meilleur résultat avec la dimension **18 868**. Au-dessus, il y a un petit changement de performance. Les performances diminuent lentement à mesure que nous réduisons les vecteurs.

Cela confirme bien que, dans la pratique, des performances à peu près équivalentes sont obtenues en utilisant la dimensionnalité de 14 000 à 100 000 [16]. Cette constatation présente un double avantage pour le système proposé. Tout d'abord, encourager l'utilisation de corpus de domaines spécifiques car, autour d'une certaine dimension, les résultats sont comparables. Il suffit alors de construire le

domaine de corpus spécifique correspondant au cours ou aux connaissances à évaluer en utilisant le système ASAGS. Deuxièmement, il est plus facile de construire (ou de trouver) un corpus (pas nécessairement de taille gigantesque) et de ne pas avoir besoin de beaucoup de ressources machine pour l'implémentation du système ASAG.

Pour les mêmes corpus nous avons généré les espaces sémantiques correspondants en appliquant un stemming différent. Nous faisons deux constatations essentielles :

- L'impact de dimensionnalité est bien confirmé puisque les résultats pour des dimensions entre 8 347 et 20 662 sont restés comparables en corrélation (CP) et en RMSE. Le meilleur résultat (CP= **48.33%)** étant obtenu pour une dimension **18 868**. Sans prendre en considération le corpus spécifique de domaine car pour un corpus de domaine on peut avoir des résultats meilleurs avec un espace sémantique de dimensionnalité beaucoup plus réduite.

- L'impact de stemming lourd est bien visible puisqu'il ya une baisse en corrélation pour pratiquement tous les corpus en utilisant le modèle somme des vecteurs pondéré avec TFmin-max comme mesure. Ceci peut être expliqué que travailler sur des racines de mots permet mieux dans une approche statistique d'accentuer le calcul de cooccurrences sur la racine d'où un meilleur résultat quand les cooccurrences sont calculées à partir des racines de mots. Ce qui est le cas pour le stemmer SnowBall, Cependant pour le stemmer Lancaster c'est un stemmer très lourd et qui n'est pas performant car il permet de réduire beaucoup la taille du mot jusqu'à dépasser la racine et aboutir à un mot qui n'a pratiquement aucun sens, ce qui explique la mauvaise évaluation et du coup une perte de corrélation de Pearson.

Tableau 20 : Dimensionnalité des espaces sémantiques générés (modèle SV sans pondération et avec pondération) en utilisant un stemmer lourd et un autre léger.

	Stemming Lourd (Lancaster)			Stemming Léger (SnowBall)		
Corpus	Dimension des vecteurs	CP(%)	RMSE	Dimension des vecteurs	CP(%)	RMSE
Leipzig	10 660	45.81	1.89	12 905	41.00	2.44
BBC	16 017	46.23	2.06	18 868	46.35	2.08
Gutenberg	18 143	44.24	2.60	20 662	44.43	2.56
Spécifique	8 347	48.29	1.77	9 777	48.33	1.83

De plus, les temps de génération des espaces sémantiques comme nous pouvons le voir dans le « Tableau 21 » sont assez grand et ceci est dû au manque considérable de ressources matériels en considérant que ces espaces en été générés avec nos propre pc laptop avec des capacité limités ce qui parfaitement normale, mais malgré ceci, ces résultats restent raisonnables comparé à des temps d'exécution beaucoup plus importants dans des approches de machines learning surtout si nous rappelons que l'espace sémantique n'est généré qu'une seule fois.

Tableau 21 : Temps de génération des espaces sémantiques.

Corpus	Temps d'exécution
BBC	Environ 2 heures
Leipzig	Presque une demi-journée
Gutenberg	Presque une journée

Etant comparables pour les espaces sémantiques déjà construits, tous les résultats rapportés dans ce qui va suivre sont calculés en utilisant un stemming Leger.

4. Les résultats des deux modèles de similarité proposés et de leurs variantes

4.1. Les résultats du modèle CM :

Dans les tableaux suivants « Tableau 22 », nous présentons les résultats obtenus pour le modèle CM sur les différents espaces sémantiques appliquer au

dataset de mohler. Je tiens à préciser que le dataset contient des erreur d'orthographe ou des erreurs de frappe commis par les étudiants.

Nous constatons d'un côté que même si les meilleurs résultats sont obtenus pour l'espace Leipzig, les résultats pour les autres restent très proches et comparables. Pour les différents corpus, la similarité d'ordre (représentée ici par WF) n'a pas une importance significative dans notre travail puisque les meilleurs résultats sont obtenus avec wf=0.0, wf=0.1 ou Wf=0.2.

Tableau 22 : Résultats du modèle CM avec les différents espaces sémantiques.

		Calcul Matriciel (CM)					
Corpus	**WF**	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
BBC Brute	**CP (%)**	48.99	48.96	49.24	49.28	47.87	47.91
	RMSE	2.82	2.82	2.77	2.80	2.83	2.80
BBC SnowBall	**CP (%)**	49.49	49.18	49.29	48.18	47.71	47.42
	RMSE	2.52	2.60	2.58	2.65	2.69	2.60
BBC Porter	**CP (%)**	49.61	50.14	49.34	49.12	48.77	46.90
	RMSE	2.53	2.61	2.05	2.66	2.70	2.81
Leipzig SnowBall	**CP (%)**	50.10	50.34	50.98	49.37	49.40	47.57
	RMSE	2.61	2.61	2.57	2.66	2.68	2.81

En contrepartie dans le tableau « tableau 23 » on remarque que la pondération **TFmin-max** a amélioré les résultats comparés aux autres mesures de pondérations, pour tous les corpus et surtout pour les corpus spécifiques de domaine ce qui confirme ce qui a été dit précédemment.

Tableau 23 : Résultats et impact des différents mesures de pondération.

	SV pond TF-IDF		SV pond IDF		SV pond TFmin-max	
	CP (%)	**RMSE**	**CP (%)**	**RMSE**	**CP (%)**	**RMSE**
Corpus Spécifique	**43.04**	**2.20**	**43.86**	**2.08**	**47.91**	**1.86**

Dans pour les restes des résultats on va considérer la pondération TFmin-max comme mesure de pondération.

Dans les tableaux suivants « Tableau 24 », nous présentons les meilleurs résultats obtenus pour le modèle CM sur le corpus de domaine finale tout en appliquant une correction orthographique au dataset de Mohler.

De plus Comme nous avons pu remarquer la présence d'erreurs d'orthographe ou erreurs de frappe commis par les étudiants dans le dataset de Mohler, Ce qui engendre une diminution des résultats et pour palier à ce problème nous avons adopter une étape de correction orthographique qui sera appliquer sur le dataset de Mohler. Et on pourra constater qu'une grande amélioration est présente pour presque tous les espaces générés précédemment.

Tableau 24 : Résultats du modèle CM avec l'espace sémantique spécifique.

		Calcule Matriciel (CM)					
DS	**WF**	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
Corpus spécifique Finale	**CP (%)**	51.40	52.43	51.98	49.98	49.98	48.50
	RMSE	2.43	2.27	2.30	2.54	2.54	2.59

Dans le tableau suivant « Tableau 25 », nous présentons les résultats obtenus pour le modèle SV avec et sans pondération de fréquences sur les différents espaces sémantiques. Nous constatons d'un côté que même si les meilleurs résultats sont obtenus pour l'espace BBC Brute, les résultats pour les autres restent très proches et comparables.

Nous constatons essentiellement que la pondération TFmin-max a nettement amélioré les résultats BBC en état brute mais a diminué pour les autres Espaces. En considérant par exemple l'espace sémantique BBC brute, la pondération a amélioré la corrélation de Pearson de +1,01. D'où l'importance de considérer l'importance des mots (exprimée ici par les poids TFs) dans le corpus. Ce qui n'est pas le cas pour les

espaces qui ont été stemmé en remarque une diminution de la corrélation de Pearson de -1.24, -0.37, -1.14.

Tableau 25 : Dimensionnalité et résultats(modèle SV) des différents corpus traiter, avec et sans correction orthographique.

		Sum Vec No Pond		Sum Vec Pond	
	Dimension des vecteurs	CP (%)	RMSE	CP (%)	RMSE
BBC Brute	27 757	47.70	2.22	48.71	2.30
BBC Brute avec correction		48.85	2.41	48.77	2.35
BBC SnowBall	18 868	47.89	2.11	46.65	1.86
BBC SnowBall avec correction		48.12	2.11	46.65	1.86
BBC Porter	18 998	47.28	1.87	46.91	2.06
BBC Porter avec correction		47.91	2.12	47.60	2.08
Leipzig SnowBall Diminution de 4 par document	12 905	42.14	2.56	41.00	2.44
Leipzig SnowBall Diminution de 4 par document avec correction		42.67	2.56	41.27	2.53
Gutenberg avec stem Snowball Diminution de 70	20 662	44.43	2.56	42.70	2.53
Gutenberg avec stem Snowball Diminution de 70 avec correction		44.90	2.52	43.26	2.52

Nous avons mis à l'évaluation une autre technique de stemming qui est la lemmatisation « Tableau 26 », Cette technique est souvent confondue avec le stemming en terme de ressemblance mais cette dernière se base beaucoup plus sur le sens du mot et non sur la forme du mot, mais cette dernière n'a pas donner de meilleurs résultats comparé au stemming.

Tableau 26 : Résultats de la Lemmatisation pour le corpus de domaine.

		Sum vec No Pond		Sum Vec Pond	
Corpus spécifique initial avec Lemmatisation sans correction	6 202	43.81	2.02	47.91	1.86
Corpus spécifique initial avec Lemmatisation avec correction		44.73	1.79	46.25	1.99

Dans le « Tableau 27 » suivant nous présentons les résultats obtenus pour le modèle SV avec et sans pondération pour les 2 corpus de domaine généré, un corpus initial avec une taille réduire et l'autre avec une taille un peu plus abordable, avec et sans correction orthographique.

Tableau 27 : Dimensionnalité et résultats(modèle SV) des 2 corpus spécifique de domaine, avec et sans correction orthographique.

		Sum Vec No Pond		Sum Vec Pond	
	Dimension des vecteurs	CP (%)	RMSE	CP (%)	RMSE
Corpus spécifique initial avec Snowball	4 312	45.75	1.64	47.91	1.86
Corpus spécifique initial avec Snowball		46.62	1.63	48.29	1.85
Corpus spécifique Finale avec stem Snowball	9 777	46.46	1.70	48.14	1.84
Corpus spécifique Finale avec stem Snowball		46.77	1.69	48.33	1.83

Nous remarquions que les résultats obtenus par les corpus de domaine sont meilleurs avec une dimensionnalité d'espace sémantique beaucoup plus réduite.

Nous remarquions aussi l'impact du correcteur orthographique qui améliore les résultats.

Dans le tableau suivant « Tableau 28 » on va présenter les résultats pour le corpus spécifique de domaine finale avec CM tout en appliquant une correction orthographique.

Tableau 28 : Résultats du modèle CM avec l'espace sémantique spécifique.

		Calcule Matriciel (CM)					
DS	**WF**	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
Corpus spécifique Finale	**CP (%)**	51.40	52.43	51.98	49.98	49.98	48.50
	RMSE	2.43	2.27	2.30	2.54	2.54	2.59

Nous remarquions que le calcule matriciel (CM) a donné les meilleurs résultats en terme de corrélation mais ce dernier ne permet pas d'avoir une bonne erreur quadratique en comparaison avec les modèle Somme des vecteurs (SV).

4.2. Combinaison des modèles (CM et SV) :

Tableau 29 : Résultats de la combinaison CM –SV avec l'espace sémantique spécifique de domaine.

		Combinaison CM – SV					
DS	**WF**	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
SV pond–CM-Moy	**CP (%)**	50.80	50.59	50.66	50.59	49.97	49.70
	RMSE	2.06	2.12	2.18	2.18	2.20	2.23
SV pond-CM-Max	**CP (%)**	48.14	48.04	48.03	48.03	48.03	48.03
	RMSE	1.82	1.82	1.82	1.82	1.82	1.82

On remarque une baisse légère en corrélation et une amélioration dans l'erreur quadratique ce qui n'est pas mal si nous étant face à une situation d'évaluation individuel car minimiser l'erreur quadratique ainsi qu'améliorer la corrélation est notre objective.

5. Hybridation avec les mesures syntaxiques

Les résultats obtenus par les deux modèles de similarité proposés ont été combinées avec des mesures syntaxiques.

Le modèle de calcul matriciel a été développé pour une mesure syntaxique où la similarité entre deux mots est calculée dans une matrice syntaxique en appliquant l'algorithme LCS ([37][38]). Une combinaison SV et CM syntaxique a été faite ainsi qu'avec la mesure syntaxique DICE. Les résultats sont reportés dans le « Tableau 30 ».

Tableau 30 : Résultats de la combinaison SV - DICE et SV - LCS.

	Corpus spécifique finale		
	Dimension des vecteurs	CP	RMSE
Sv - LCS - Moyenne	9 777	42.65	1.98
Sv pond - LCS - Moyenne		46.41	1.92
Sv - LCS - Maximaum		48.57	1.64
Sv pond - LCS - Maximaum		46.03	1.86
	Corpus spécifique finale		
	Dimension des vecteurs	CP	RMSE
Sv - DICE - Moyenne	9 777	45.36	1.99
Sv pond - DICE - Moyenne		46.37	1.98
Sv - DICE - Maximaum		46.34	1.95
Sv pond - DICE - Maximaum		47.71	1.92

Dans les deux cas, une amélioration importante de la corrélation et de l'erreur quadratique est enregistrée en combinant avec les mesures syntaxiques

particulièrement la mesure DICE qui prend en considération le nombre de mots communs entre les deux réponses à comparer.

6. Récapitulation des résultats et discussion

Le « Tableau 31 » résume les meilleurs résultats obtenus pour notre approche et les différentes combinaisons :

Tableau 31. Récapitulatif des résultats

	Mohler DS	
	CP (%)	**RMSE**
CM – WF=01	**52.43**	**2.27**
Sv - LCS - Maximum	**48.57**	**1.64**
SV pond–CM-Moyenne	**50.59**	**2.12**
SV pond-CM- Maximum	**48.03**	**1.82**
SV Pond	**48.33**	**1.83**

Le meilleur résultat est obtenu en utilisant le calcul matriciel comme modèle de similarité. Une analyse approfondie « voir tableau 32 » des résultats obtenus depuis le corpus de Domaine final pour ce résultat montre que :

Plus de 60% pour un écart d'un maximum de 1 point est très bien. (On a fait la somme) sur une échelle de 5 points.

Plus de 83 % pour un écart d'un maximum de 2 points. Ce n'est pas mal.
On estime que les résultats sont très raisonnables moyennant la subjectivité humaine.

Tableau 32 : Analyse approfondie des résultats obtenue pour le corpus de domaine.

	Compteur	Pourcentage
égaux	101	16.03 %
Inférieur ou égale à 1	281	44.60 %
Inférieur ou égale à 2	143	22.69 %
Inférieur ou égale à 3	59	9.36 %
Supérieur à 3	46	7.30 %

- Plus de 60% (16.03 + 44.60). Pour un écart d'un maximum de 1 point sur une échelle de 5 point.
- Plus de 83 % (16.03 + 44.60+22.69) pour un écart d'au maximum de 2 points sur une échelle de 5 point.

En conclusion, c'est un résultat très correct et très raisonnable moyennant la subjectivité du processus de correction déjà dans sa nature humaine.

7. Evaluation par rapport aux résultats obtenus par les travaux connexes.

Dans le « Tableau 33 » nous comparons les résultats obtenus par rapport à ceux des travaux connexes de Mohler DS qui ne considère que la corrélation de Pearson sans considération de l'erreur quadratique. Notre résultat dépasse celui de la littérature de +**1.43%**.

Tableau 33. Evaluation par rapport aux travaux connexes sur Mohler DS

	CP (%)	*RMSE*
Mohler Texas system System [33]	**50.99**	-
Notre Approche Sémantique (CM – WF=0.1 – Corpus Spécifique Finale)	52.43	2.27

8. Evaluation par les Word Embedding (WE)

Pour mieux apprécier la qualité de l'approche ainsi que l'espace sémantique, nous avons combiné les résultats obtenus par les modèles de similarité avec ceux obtenus en utilisant les WE comme représentation des mots. Ceci a été réalisé en deux étapes :

1. Nous avons remplacé dans l'approche la représentation des mots par espace sémantique par une nouvelle représentation qui est les Word Embedding. Nous avons utilisé les WE pré-trainés « Word2Vec »[18] de Google qui sont disponibles à l'utilisation de manière gratuite. La Taille du modèle est de l'ordre 1.5 Go, ce qui nous donne une bonne appréciation de notre espace dont la taille est largement plus réduite.
2. Nous avons appliqué le modèle SV de similarité en utilisant les WE en utilisant le corpus spécifique
3. Nous avons ensuite combiné les résultats obtenus par l'espace sémantique et ceux en utilisant les WE pour le même modèle SV avec et sans pondération TF-Min-Max. La combinaison est illustrée dans le « Tableau 34 ».

Tableau 34 : Résultats de la combinaison SV – WE sur le corpus spécifique

	SV no pond - WE		SV pond - WE	
	CP (%)	RMSE	CP (%)	RMSE
Combinaison avec moyenne	**45.84**	**1.93**	**46.41**	**1.92**
Combinaison avec maximum	**45.31**	**2.01**	**46.03**	**1.86**

Discussion : La combinaison avec les WE n'ait pas donné de meilleurs résultats en comparaison avec les espaces sémantiques et ceci est dû à deux raisons principales :

- Les faibles résultats obtenus par les WE sans combinaison avec nos approches a diminué la corrélation de la combinaison.

[18] https://mccormickml.com/2016/04/12/googles-pretrained-word2vec-model-in-python/

- Les modèles des WE nécessitent des ressources matérielles gigantesques pour les générer. Nous avons utilisé le modèle de Google de taille très réduite pour qu'on puise l'exécuter dans nos machines. Nous pensons qu'avec plus de moyens nous pouvons augmenter la taille des WE et améliorer les résultats.

En conclusion, Il serait intéressant pour la généralisation de l'approche aux WE d'explorer encore les autres modèles de similarité proposés et d'augmenter la qualité des WE.

9. Discussion

Au terme de cette synthèse expérimentale, nous aboutissons à plusieurs constatations que nous discutons à travers les points suivants :

- Une amélioration des résultats est bien marquée avec les combinaisons (interne ou externe). Ce qui confirme bien que les modèles hybrides donnent de meilleurs résultats (Le meilleur de nos résultats est obtenu avec les combinaisons externes).
- L'approche adoptée pour la langue arabe est largement généralisée à l'anglais et reste indépendante de la langue.

10. Déploiement de l'approche sur la plateforme Moodle

Pour vérifier le passage à l'échelle et tester l'approche sur un environnement réel, nous avons déployé l'approche sur la plateforme Moodle en utilisant l'approche proposée par [41] dans le cadre du même projet d'évaluation automatique.

Nos motivations sont diverses quant au déploiement sur une plateforme de télé-enseignement à savoir :

- ✓ Bénéficier de toutes les fonctionnalités de la plateforme Moodle tel que l'historique du test, l'historique du Feedback, l'assignement des notes du test et bien d'autres fonctionnalités déjà disponible d'une plateforme de télé-enseignement.
- ✓ Vérifier que le PLUGIN présente un moule pour n'importe quelle approche d'évaluation automatique proposée dans le cadre du projet de recherche.
- ✓ Intégrer un PLUGIN dans le module Quiz (test) de Moodle.

- ✓ Combiner notre nouveau type de question avec les autres types de Moodle dans un test.
- ✓ Intégrer l'évaluation automatique des réponses courtes dans la plateforme de télé-enseignement basée sur les approches sémantiques.
- ✓ Offrir à l'enseignant la possibilité de construire son test de manière naturelle et uniforme en combinant différents types de questions déjà intégrées sur la plateforme (choix multiple, essais, remplir les vides, ...) avec le nouveau type d'évaluation proposée.
- ✓ Familiariser les apprenants à un environnement de test et au feedback consolidés sur la même plateforme de cours et offrir ainsi une navigation et utilisation facile et familière du PLUGIN auprès des enseignants et étudiants.
- ✓ Evaluer qualitativement notre approche déjà développée par rapport à un passage à l'échelle.

Le déploiement s'est fait sur la plateforme de l'université de Bouira (http://elearninginfo.univ-bouira.dz/test/moodle/). Nous avons installé le plugin qui implémente notre approche et réalisé deux types de tests :

- Le premier test est composé d'un seul type de question qui est le nouveau type développé (English Short Answer) « voir Figure 22 ».
- Le deuxième test est composé de 3 différents types de questions, avec deux questions de type (English Short Answer), une question à choix multiple, et une dernière question de type (remplir le vide). L'objectif ici est d'évaluer l'intégration du plugin avec d'autres plugins dans un même test « voir Figure 23 ».

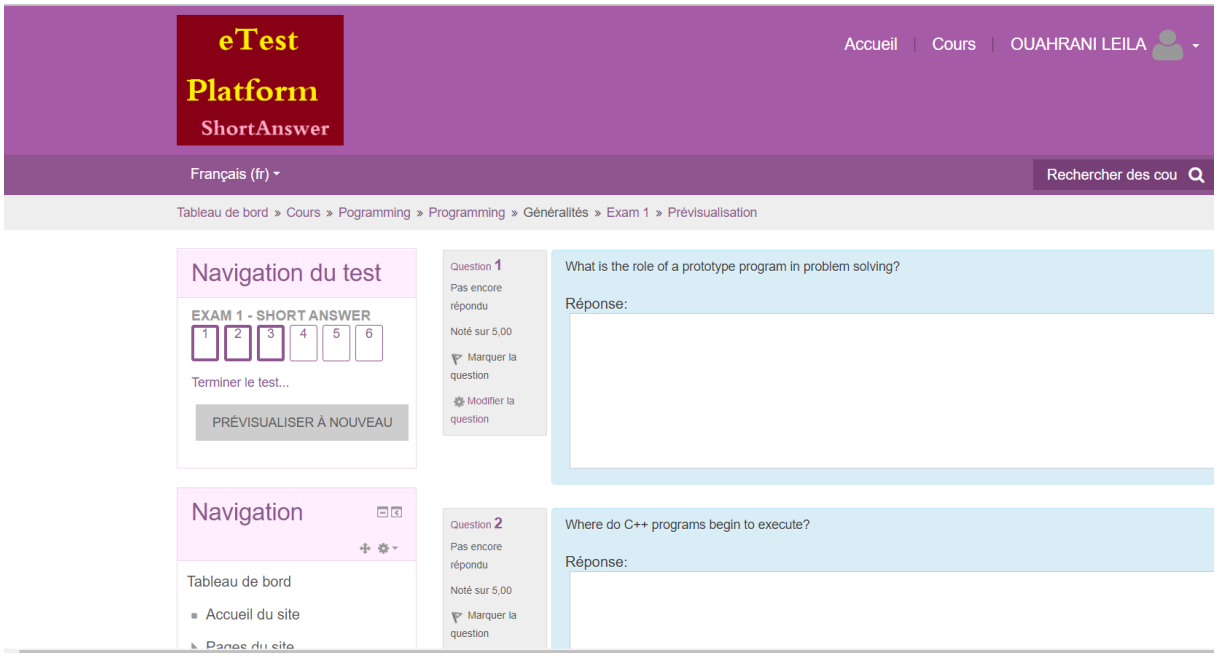


Figure 22 : Exemple d'un examen qui contient que les réponses courtes que nous avons développé

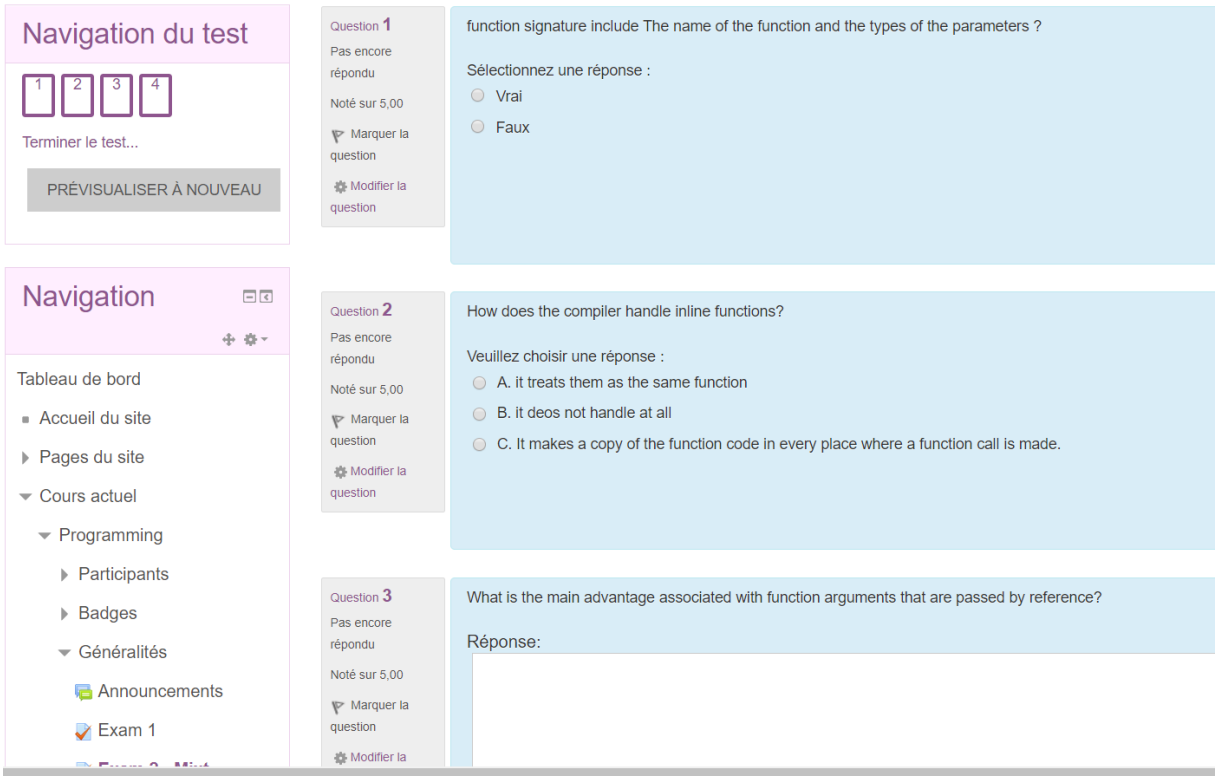


Figure 23 : Exemple d'un examen complet qui contient les réponses courtes ainsi que d'autres types de questions disponible sur la plateforme Moodle.

Nous n'avons pas eu la chance de mettre des étudiants sur ce test qualitatif vu que le déploiement ne s'est fait qu'n mois de septembre et les étudiants ne sont pas encore en période de cours pour les solliciter. Nous avons essayé nous-mêmes en jouant le rôle d'étudiants en utilisant comme banque de question le DataSet de Mohler déjà utilisé pour l'évaluation quantitative (corrélation de Pearson).

La consolidation des réponses aux questions et des remarques a permis de faire les constatations suivantes :

- Le retour étudiant est très positif à propos du test. Particulièrement le retour immédiat des notes. La notation instantanée de chaque question aide à maintenir la motivation. L'étudiant prend ainsi, le temps de bien faire les choses avant de continuer et de valider le test. Ceci en temps normal d'utilisation du plugin dans l'apprentissage du module permet, de pouvoir suivre le niveau de compréhension du cours.

- Dans un contexte d'examen, l'étudiant trouve utile le retour d'information immédiat (appréciations et des réponses de l'enseignant). En effet, le feedback instantané évite de ne pas savoir comment il évolue, et il apprécie de quitter la salle d'examen en sachant sa note.

- D'un point de vue enseignant, sans surprise l'enseignant apprécie de n'avoir que peu ou pas de tests de corrections d'examens, en particulier quand la correction concerne un nombre important de copies (C'est l'avis formulé par notre promoteur qui lui est enseignant et qui a essayé les deux plugin Arabic Short Answer et English Short Answer).

11. Conclusion : Généralisation de l'approche

Dans la Schéma suivant « voir figure 24 », nous résumons les entrées de notre approche que nous allons considérer comme une boite noire. Pour chaque langue, il suffit de changer les entrées à savoir : un corpus de textes dans la langue et un Stemmer.

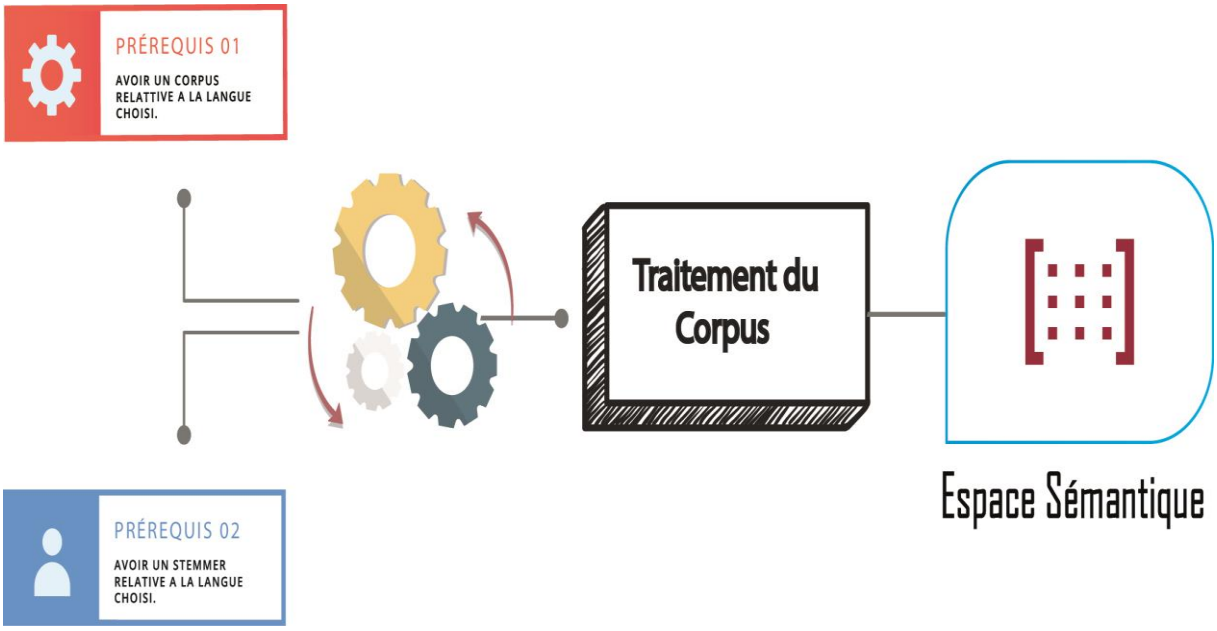


Figure 24 : Schéma qui résume les prérequis pour la généralisation de notre approche.

En effet, l'approche proposée pour la langue arabe est facilement généralisée pour une autre langue qui souffre du manque de ressources spécialement de connaissances (dictionnaires, ontologies) comme c'est le cas de la langue arabe. Pour cela il suffit d'avoir un corpus de la langue et un Stemmer (ce qui est facile à avoir aujourd'hui avec le web.

Conclusion générale

L'évaluation automatique est un sujet d'actualité qui est en train de bénéficier de l'avancement de la technologie particulièrement en termes de modèles et de ressources.

L'approche statistique (basé corpus) est la solution la plus appropriée pour les langues qui manquent de ressources linguistiques telle que la langue Arabe. Pour cela nous avons axé notre recherche sur la création de l'espace sémantique ainsi la généralisation de cette approche pour les autres langues qui souffrent notamment de manque de ressources linguistiques.

Notre thème est au milieu du carrefour de nombreux domaines, en conséquence nous avons présenté un état de l'art sur les systèmes d'évaluation, le traitement de la langue, les approches de similarité syntaxique et sémantique.

Nous avons proposé par la suite notre approche après une large recherche ainsi qu'une étude complète de ce qui a été fait dans les travaux précédents et avec comparaison des approches existantes. En résultat un système d'évaluation automatique a été développé qui se compose de plusieurs modules. Le système est combiné avec d'autres approches syntaxiques et Word Embeddings. Nous avons évalué notre système avec un dataset qui nous a été fourni (le seul de la littérature actuelle pour l'anglais). Nous avons pu en améliorer la corrélation globale.

Nous avons pu ajouter un module de correction grammaticale automatique pour les réponses des étudiants introduit car nous avons pu remarquer qu'un taux élevé d'erreurs est commis par les étudiants et cela affecte négativement l'évaluation moyennant le fait que dans le genre de questions que nous traitons le style des réponses n'est pas ciblé (seule la synthèse du contenu de la réponse est évaluée plutôt que le style).

Nous avons pu aussi intégrer notre système dans une plateforme de télé-enseignement Moodle. Ainsi nous pouvons déployer des examens réels variés avec différents types de questions ainsi que des questions à réponses courtes que nous avons développées.

Nous avons pu vérifier que l'approche déjà développée pour la langue arabe est facilement généralisable pour toute autre langue et dans ces cas indépendants de la langue (Independant-Language Approach). Sachant que les Stemmers et les corpus sont les seules ressources dépendantes de la langue prérequises.

En perspectives, nous pensons considérer entre autres les aspects suivants :

- Identifier les entités nommées ainsi que les collocations et les types de mots (verbe, adjectif, nom…) dans l'identification des réponses,

- Explorer encore plus du côté des WE de manière à maintenir le système utilisable et améliorer sa corrélation.

- Adapter le système à la possibilité de considérer plusieurs réponses modèles de l'enseignant. Ceci revient à intégrer l'automatisation de la génération du corpus de réponses modèle à notre système.

Bibliographie

[1] J. Sukkarieh and J. Blackmore, “c-rater: Automatic Content Scoring for Short Constructed Responses.,” FLAIRS Conf., pp. 290–295, 2009.

[2] Y. ATOUB, A. BENAYAD « Mesures de similarité sémantique pour un système d’évaluation automatique des réponses courtes: Application à la langue arabe». Mémoire master USDB 1. 2017/2018.

[3] M. H. Abu Mugasib and R. S. Baraka, “An Ontology-Based Automated Scoring System for Short Answer Questions,” 2015.

[4] S. Burrows, I. Gurevych, and B. Stein, The eras and trends of automatic short answer grading, vol. 25, no. 1. 2015.

[5] D. Callear, J. Jerrams-Smith, V. Soh, D. J. Jerrams-smith, and H. P. Ae, “CAA of Short Non-MCQ Answers,” in In Proceedings of the 5th International CAA conference, 2001.

[6] C. Leacock and M. Chodorow, “C-rater: Automated Scoring of Short-Answer Questions,” Comput. Hum., vol. 37, no. 4, pp. 389–405, 2003.

[7] S. Jordan, “Investigating the Use of Short Free Text Questions in Online Assessment,” Final Proj. report, Cent. Open Learn. Math. Sci. Comput. Technol. Open Univ. Milt. Keynes, United Kingdom, 2009.

[8] J. Z. Sukkarieh and S. Stoyanchev, “Automating Model Building in C-rater,” in Proceedings of the 2009 Workshop on Applied Textual Inference, 2009, pp. 61–69.

[9] L. Cutrone, M. Chang, and Kinshuk, “Auto-Assessor: Computerized Assessment System for Marking Student’s Short-Answers Automatically,” in 2011 IEEE International Conference on Technology for Education, 2011, pp. 81–88.

[10] T. Pedersen, S. Patwardhan, and J. Michelizzi, “WordNet::Similarity: Measuring the Relatedness of Concepts,” in Demonstration Papers at HLT-NAACL 2004, 2004, pp. 38–41.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update,” SIGKDD Explor. Newsl., vol. 11, no. 1, pp. 10–18, 2009.

[12] N. Madnani, J. Burstein, J. Sabatini, and T. O’Reilly, “Automated scoring of a summary writing task designed to measure reading comprehension,” Naacl/Hlt 2013, p. 163, 2013.

[13] E. Negre, “Comparaison de textes: quelques approches...,” 2013.

[14] “Indice et distance de Jaccard,” 2018. .

[15] R. A. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.

[16] D. L. T. Rohde, L. M. Gonnerman, and D. C. Plaut, “An Improved Method for Deriving Word Meaning from Lexical,” Cogn. Psychol., vol. 7, pp. 573–605, 2004.

[17] W. H.Gomaa and A. A. Fahmy, “A Survey of Text Similarity Approaches,” Int. J. Comput. Appl., vol. 68, no. 13, pp. 13–18, 2013.

[18] “WordNet — Wikipédia,” 2018. .

[19] W. H. Gomaa and A. A. Fahmy, “Arabic Short Answer Scoring with Effective Feedback for Students,” Int. J. Comput. Appl., vol. 86, no. 2, pp. 35–41, 2014.

[20] W. H. Gomaa and A. A. Fahmy, “Automatic scoring for answers to Arabic test questions,” Comput. Speech Lang., vol. 28, no. 4, pp. 833–857, 2013.

[21] P. Kolb, “Disco: A multilingual database of distributionally similar words,” Proc. KONVENS-2008, Berlin, no. 2003, pp. 37–44, 2008.

[22] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, “SemEval-2017 Task 1: Semantic Textual Similarity - Multilingual and Cross-lingual Focused Evaluation,” Proc. 11th Int. Work. Semant. Eval. (SemEval-2017), pages 1–14, Vancouver, Canada, August 3 - 4, pp. 1–14, 2017.

[23] A. ABDALLAH, K. GAROUDJA, « Mesures de similarité syntaxique pour un système d'évaluation automatique des réponses courtes : Application à la langue arabe ». Mémoire master USDB 1. 2017/2018.

[24] M. H. HANNOUFI, A. N. HENNICHE, « Les Word-Embedding pour l'évaluation automatique des réponses courtes en apprentissage en ligne : Application à la langue arabe ». Mémoire master USDB 1.2017/2018.

[25] J. B. Lovins, "Development of a stemming algorithm," Mech. Transl. Comput. Linguist., vol. 11, no. June, pp. 22–31, 1968.

[26] "SAFAR Web v2." .

[27] C. Tambellini and C. Berrut, "Pondération des données incertaines dans les systèmes de recherche d ' informations : une première approche expérimentale."

[28] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," Inf. Process. Manag., vol. 24, no. 5, pp. 513–523, 1988.

[29] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," ACM Trans. Knowl. Discov. Data, vol. 2, no. 2, pp. 1–25, 2008.

[30] H. M. HANNOUFI and N. A. HENNICHE, "Les Word-Embedding pour l'évalutaion automatique des réponses courtes en apprentissage en ligne : Application à la langue arabe.," Saad Dahleb Blida1, 2018.

[31] J. Macqueen, "Some methods for classification and analysis of multivariate observations," Proc. Fifth Berkeley Symp. Math. Stat. Probab., vol. 1, no. 233, pp. 281–297, 1967.

[32] L. Ouahrani, "String similarity for Arabic short answer grading," Intern. report, LIMPAF/118, LIMPAF Lab. Bouira Univ., 2018.

[33] M. Mohler and R. Mihalcea, "Text-to-text Semantic Similarity for Automatic Short Answer Grading," 2004.

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

[35] W. Greene, “Econométrie, Paris, Pearson Education,” 5e éd. (ISBN 978-2-7440-7097-6), 2005.

[36] M. Berry, Z. Drmac, and E. Jessup, “Matrices, Vector Spaces, and Information Retrieval,” SIAM Rev., vol. 41, no. 2, pp. 335–362, 1999.

[37] J. W. Hunt and M. D. MacIlroy, An algorithm for differential file comparison. Bell Laboratories, 1976.

[38] L. Allison and T. I. Dix, “A bit-string longest-common-subsequence algorithm,” Inf. Process. Lett., vol. 23, no. 5, pp. 305–310, 1986.

[39] « Définition : Question ouverte » Définitions marketing ». [En ligne]. Disponible sur: https://www.definitions-marketing.com/definition/question-ouverte/. [Consulté le: 15-sept-2018].

[40] P. Ryu and K. Choi, “Measuring the specificity of terms for automatic hierarchy construction,” Proc. ECAI Work. Ontol. Learn. Popul., 2004.

[41] A. MADANI, E. SNOUSSI, « Développement d’un PLUGIN d’évaluation automatique des réponses courtes pour une plateforme de télé-enseignement ». Mémoire master USDB 1. 2018/2019.

Annexe

A. Liste des stop-words :

Nous avons un totale de 179 Mots vides.

i	me	my	myself	we	our	ourselves	you
You're	You've	You'll	You'd	your	yours	yourself	yourselves
he	him	his	himself	she	She's	her	hers
herself	it	It's	its	itsself	they	them	their
theirs	themselves	what	wich	who	whom	this	that
That'll	these	those	am	is	are	does	did
doing	a	an	the	and	but	if	or
because	as	until	while	or	at	by	for
with	about	against	between	into	through	during	before
after	above	against	to	from	up	down	In
out	on	off	over	under	again	further	then
once	few	more	most	so	than	too	very
s	t	can	will	just	don	Don't	should
Should've	now	d	ll	m	o	re	ve
y	ain	aren	Aren't	couldn	Couldn't	Didn	Didn't
doesn	Doesn't	hadn	Hadn't	hasn	Hasn't	haven	Haven't
isn	Isn't	ma	mightn	Mightn't	mustn	Mustn't	needn
Needn't	shan	Shan't	shouldn	Shouldn't	wasn	Wasn't	weren
Weren't	won	Won't	wouldn	Wouldn't			

B. Nombre des mots non-trouvés par SnowBall stem :

	Corpus BBC	Corpus Leiptiz	Corpus Gutenberg	Corpus spécifique Finale
Avec répétition	665	587	1897	157
Exhaustive	**110**	**117**	**180**	**83**

Après application d'une étape de correction grammatical.

	Corpus BBC	Corpus Leiptiz	Corpus Gutenberg	Corpus spécifique Finale
Avec répétition	519	442	1811	60
Exhaustive	**46**	**57**	**122**	**26**