Page de garde

<REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLEB DE BLIDA
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

CENTRE DE RECHERCHE SUR L'INFORMATION SCIENTIFIQUE ET TECHNIQU

MEMOIRE DE FIN D'ETUDES MASTER 2

THÈME :

DETECTION DE PLAGIAT DANS LES TEXTES ARABES

Sujet proposé par :

Dr.ALLIANE Hassina , CERIST.

Réalisé par :

Setha Imene

Attaba Yasmine

Encadré par :

Dr.ALLIANE Hassina, CERIST.

Pr.BENBLIDIA Nadjia, USDB.

Président de jury : Aroussi

Examinateur : Boutoumi

Organisme d'accueil :

Centre De Recherche sur l'Information Scientifique et Technique.

Promotion : septembre 2016

Remerciement

Nous tenons tout d'abord à remercier Madame ALLIANE Hassina, pour la confiance qu'elle nous a accordée en acceptant de diriger ce mémoire et sa patience de la suivre jusqu'à son aboutissement. On la remercie profondément pour son attention, sa bienveillance et son appui sans faille qui ont été des encouragements décisifs pour mener à terme ce travail. Sans ses qualités rares au niveau humain et scientifique, le développement et l'achèvement de ce travail n'auraient été possibles.

Nous sommes sincèrement reconnaissantes à vous, Madame ALLIANE. On a un grand honneur et une grande chance d'avoir une encadreuse comme vous.

Nous tenons a remercie notre promotrice Madame BENBLIDIA Nadjia , pour sa confiance aussi sa générosité, ses conseils et orientations qui ont été une grande aide pour nous durant la réalisation de ce travail.

On voudrait également remercier les personnes qui nous font l'honneur d'accepter de participer au jury de ce mémoire.

On voudrait remercier tous ceux qui ont facilité notre tâche et nous a permis de mener à bien ce travail ainsi que ceux qui nous a aidés dans nos études, et que je n'ai pas pu citer.

Enfin, je remercie Dieu tout puissant de la patience et de la volonté qu'il nous a donné pour réaliser ce travail.

Résumé

Le plagiat est le résultat de l'action qui consiste à s'emparer délibérément ou par omission des mots ou des idées de quelqu'un d'autre et de les présenter comme siens. Nous proposons dans ce mémoire de résoudre le problème de plagiat en langue arabe par un système de détection de plagiat. Ce système, recherche en réalité les similitudes entre deux textes en arabes.

Il est basé sur l'approche Analyse Sémantique Latente qui est une méthode statistique permettant d'extraire automatiquement des relations conceptuelles entre les termes d'une collection de texte. (LSA)

Mots-clés : Plagiat, Similarité, Analyse Sémantique Latente, Arabic word net.

ملخص

الانتحال هو نتيجة عمل الامتلاك العمدي او عن طريق حذف او تغيير لكلمات او افكار شخص اخر وتقديمها باعتبارها ملكا له. نوضح في هذه المذكرة حل لمشكلة الانتحال بالنسبة للغة العربية عن طريق برنامج ضد الانتحال.

هذا البرنامج يبحث عن اوجه التشابه ما بين نصين باللغة العربية فهو يتمحور حول منهجية (LSA) التي تعتبر طريقة احصائية لاستخراج تلقائيا العلاقات المفاهمية بين المصطلحات التي تتواجد في مجموعة من النصوص.

الكلمات المفتاحية : انتحال ، تشابه ، تحليل الدلالات الكامنة ، الأنطولوجيا العربية وردنت .

Abstract

Plagiarism is the result of steeling someone else ideas or words and represent it like ours. We propose in this work to resolve this problem of plagiarism in Arabic language by a system of plagiarism detection.

Our work is based on the latent semantic approach which is a statistic method to improve our search in this system.

Keywords: plagiarism, similarity, Latent Semantic Analysis, Arabic word net.

Sommaire

Chapitre II : Détection de plagiat dans les textes en langue arabe

Chapitre VII : Implémentation et réalisation

Chapitre I : L'état de l'art

Chapitre II : Détection de plagiat dans les textes en langue arabe

Chapitre III : Conception du système de détection de plagiat

Chapitre VII : Implémentation et réalisation

Chapitre II : Détection de plagiat dans les textes en langue arabe

Chapitre III : Conception du système de détection de plagiat

Chapitre VII : Implémentation et réalisation

Chapitre II : Détection de plagiat dans les textes en langue arabe

Chapitre III : Conception du système de détection de plagiat

Introduction générale

Introduction

L'évolution rapide des techniques de l'information et la communication, l'accroissement des échanges internationaux ont modifié en quelques décennies le monde de la connaissance et les exigences en matière du plagiat.
Le plagiat n'est pas une fraude récente, cette notion est très régulièrement utilisée mais la nouveauté réside dans sa généralisation. En effet, une plus grande facilité d'accès aux documents électroniques via le web est toujours une préoccupation au sein de la communauté universitaire, aussi la facilité d'accès a une grande masse d'informations a favorisé la généralisation du phénomène du plagiat.

Dans le cadre de ce mémoire, nous tenterons de développer un système Anti-Plagiat, permettant de détecter le plagiat dans des textes en langues arabes. La plupart des recherches se base sur les approches de similarité locale et pour cela nous allons adopter une approche avancée sur la similarité globale du document qui a été beaucoup moins étudier par rapport aux autres approches et spécialement dans la langue arabe, c'est l'approche de l'Analyse Sémantique Latente qui simule l'acquisition de connaissances à partir de l'analyse entièrement automatique de corpus de textes.

L'objectif de notre travail est de :

Mener une étude afin de concevoir et implémenter un logiciel qui permet de détecter le plagiat dans les textes arabes. Ce système est basé sur une approche de calcule de similarité pour les textes en utilisant des différents Pré-traitements et méthodes.

Notre travail est composé de quatre chapitres :

- Le premier chapitre présente les notions de base et les principaux concepts liés au domaine abordé dans ce mémoire, à savoir, le domaine des traitements automatiques ddes langages. Il se focalise particulièrement sur la description de détection de similarité dans les textes arabe. D'abord il présente la définitions de plagiat avec ses formes et ses types, ainsi qu'un aperçu sur les principaux modèles de détection de similarité existants dans la littérature telle que l'analyse sémantique latente, le fingerprinting…etc. il aborde aussi les différentes mesures de similarité théorique dans le domaine de traitement du texte.

Le chapitre décrit aussi brièvement les différents logiciels dans le monde pour la détection de plagiat dans les textes de différentes langues.

- Le deuxième chapitre décrit l'aspect utilisation des différentes méthodes qui portent sur l'utilisation d'un détecteur plagiat dans les textes arabe.
Il présente les caractéristique de la langue arabe, il montre les méthodes et les systèmes déjà utilisés et créés par d'autre chercheurs et enfin une comparaison entre les différents travaux et approches utilisées.

- Le troisième chapitre propose la conception de notre système, ce chapitre commence par une architecture générale pour le système. Ensuite, il aborde tous les traitements proposés pour atteindre le but de ce travail, une explication détaillée de la méthode d'Analyse Sémantique Latente. Finalement, des différents algorithmes proposés pour avoir la détection de similarité dans les paragraphes de texte entré par l'approche de LSA.

- Le quatrième chapitre présente notre implémentation et notre réalisation. Il concerne l'intégration de tous les traitements et tous les algorithmes proposés pour atteindre le but de détection de plagiat dans les textes arabes. Ensuite nous présentons différents outils implémentés.
Les différents résultats correspondants à différentes expérimentations sont ensuite rapportés et commentés, pour enfin terminer par une conclusion sur l'apport de la détection de plagiat et quelques perspectives de son utilisation pour les recherches informatique.

CHAPITRE I

ETAT DE L'ART:

Introduction

Le plagiat est un phénomène qui explose depuis quelques années dans notre monde. Et elle est largement décriée, qu'il soit le fait d'étudiants ou d'enseignants-chercheurs. Dans le monde, peu d'études consistantes ont été consacrées à ce phénomène qui interroge les conditions de formation à l'université. Alors l'objectif que nous fixons par ce travail est la détection de documents plagiés au sein d'un corpus de langue arabe. Dans ce chapitre, nous allons tirer profit des notions existantes su le plagiat en générale et l'ensemble des concepts et méthodes utilisés dans notre travail.

1. Définition du plagiat

1.1 Définition commune du plagiat

Plagiaire [plaʒjɛʀ].n. Personne qui pille ou démarque les ouvrages des auteurs. V. Imitateur. [1]

Plagiat. n.m. Action de plagier, vol littéraire. V. Copier, imitation. [1]

Autrement dit, le plagiat est le vol d'une propriété intellectuelle appartenant à un autre. Ce qui comprend à la fois le vol des idées et des concepts non écrites, ainsi que le vol d'un texte, notes, programmes informatiques, des dessins, des images. Dans de nombreux cas, le vol de la propriété intellectuelle est intentionnelle et, dans certains cas, malveillant dans sa nature. [2]

1.2 Définition du plagiat académique

> Utilise les mots, les idées ou le travail de quelqu'un d'autre, Alors que l'on peut identifier la provenance des données sans que la personne concernée reconnaisse cette source; Dans une situation où existe une attente légitime quant à la paternité (authorship) en vue d'obtenir un avantage, du mérite, un gain. [7]

De même, selon la section « Droit d'auteur et plagiat » du site Infosphère[1] de l'Université de Montréal, plagier, c'est :

- S'approprier le travail créatif de quelqu'un d'autre et le présenter comme sien;

[1] Le site de l'université UQAM de Montréal http://www.infosphere.uqam.ca/

- Prendre des extraits de texte, des images, des données, etc. provenant de sources externes et les intégrer à son propre travail sans en mentionner la provenance;
- Résumer l'idée originale d'un auteur en l'exprimant dans ses propres mots, mais en omettant d'en mentionner la source.
- Le plagiat s'inscrit comme l'une des formes possibles de tricherie en contexte académique. [7]

1.3 Le plagiat dans le contexte numérique

Les institutions d'enseignement sont actuellement confrontées à une réalité indéniable : le développement du Web et de ses ressources a modifié radicalement la recherche documentaire et la réalisation des travaux académiques. Étudiants, professeurs et professionnels œuvrant au sein des institutions de formation, tous recourent à Internet, qui met à la disposition de ses utilisateurs un ensemble de données et d'informations d'une ampleur phénoménale.

La génération C^2, qui fréquente actuellement nos universités, est née et a grandi à l'ère numérique. Elle fait un usage généralisé du Web, s'en servant non seulement pour s'informer, mais aussi pour communiquer, créer et collaborer.

Une grande proportion des 18 à 24 ans est considérée comme de grands utilisateurs d'Internet : 40 % d'entre eux passent ainsi 21 heures ou plus sur le Web par semaine. Il va sans dire que la recherche documentaire en contexte universitaire et que la réalisation des travaux sont influencées par l'omniprésence des technologies dans la vie des étudiants. Toujours selon l'enquête menée par le CEFRIO,

- 91 % des étudiants québécois de 12 à 24 ans utilisent un ordinateur pour réaliser leurs travaux;
- 29 % des universitaires sondés en emploient systématiquement un en classe.

Par ailleurs, une étude menée en 2006 en France par les sociétés Le Sphinx Développement et Six Degrés auprès de 975 étudiants provenant de divers établissements universitaires et de domaines variés (informatique, physique, biologie, sciences humaines, etc.) a montré que 97 % des étudiants emploient Internet comme source principale de documentation. Questionnés sur

---

2

les avantages obtenus à se servir d'Internet comme ressource documentaire, les étudiants ont donné, en ordre d'importance,

- La rapidité d'accès aux informations (88 %),
- La variété des sources trouvées (77 %),
- La facilité de réutilisation des sources (36 %),
- La qualité des sources trouvées (15 %).

Ces résultats montrent clairement que le recours au Web dans la réalisation des travaux académiques occupe une place considérable dans les habitudes des étudiants d'aujourd'hui.

Si Internet peut être considéré comme une ressource utile à l'apprentissage, l'importante démocratisation de l'information qu'il a entraînée comporte des inconvénients. Notamment, il semble légitime d'affirmer que l'accessibilité des informations fournie par Internet combinée à la facilité d'utiliser la fonction copier-coller aurait amplifié le phénomène du plagiat. En effet, les étudiants pouvant accéder aisément aux données du Web, ils se sentiraient légitimes de se les approprier et ne se jugeraient donc pas coupables. [7]

1.4 Formes du plagiat académique

L'observation du monde réel sur le plagiat révèle une variété de courante formes trouvées.

Le plagiat littéral décrit la copie excessive du texte avec très peu ou pas de déguisement.

- ✓ **Copy, Paste :** Est la forme la plus commune de plagiat littérale et se caractérise par l'adoption du texte verbatim d'une autre source. [3]

Le plagiat déguisé subsume la pratique pour cacher le texte indûment copié [4]. Le plagiat déguisé couvre cinq formes dans la littérature.

- ✓ **Shake, Paste ou mosaïque :** Se réfère à la copie et la fusion des segments de texte avec de légers ajustements pour former un texte cohérent, par exemple par changer l'ordre des mots, en remplaçant les mots avec des synonymes, ou par l'ajout ou la suppression des mots «de remplissage». [3]

- ✓ **Plagiat expansif :** Se réfère à l'insertion d'un texte supplémentaire dans un ou en plus des segments copiés. [4]
- ✓ **Plagiat constrictive :** Décrit le résumé ou la coupe de matériel copié. [4]

Le déguisement technique est résumé pour cacher la détection du contenu plagié automatiquement détecté en exploitant les faiblesses du courant texte à base d'analyse par exemple par substituant des caractères avec des symboles graphiquement identiques d'alphabets étrangers ou des lettres d'insertion en blanc couleur de la police. [5]

- ✓ **Paraphrase excessive :** Définit la réécriture intentionnelle des pensées étrangères dans le vocabulaire et le style du plagiaire sans donnant du crédit en raison de dissimuler la source d'origine.
- ✓ **Plagiat Traduit :** Est défini comme une conversion manuelle ou automatisée du contenu d'une langue à une autre destinée à couvrir son origine. [3]
- ✓ **Auto-plagiat :** L'auto-plagiat consiste à réutiliser des parties de contenus dont on est l'auteur pour l'insérer dans un nouveau document. Dans ce cas également, même si l'on en est l'auteur, il faut indiquer dans le nouveau document créé l'origine des extraits, et des passages que l'on réutilise. [6]

2. La détection du plagiat

2.1 L'histoire de la détection du plagiat

Dans les premiers jours, le plagiat ne pouvait être détecté manuellement en appuyant sur la connaissance des lecteurs. Comme la cognition varie de personne à personne, et la grande quantité de matériaux est impossible à atteindre, l'identification du processus de plagiat dans le texte peut être une tâche difficile. Dans la plupart des cas, le plagiat est identifié par la lecture d'un texte qui déclenche une '' Déjà vu '' dans le lecteur, où le lecteur a reconnu-t-il.

L'inconvénient évident de la méthode manuelle est que, lorsque la quantité de l'information augmente, un lecteur est moins susceptible d'être en mesure d'identifier les similitudes. Le cerveau humain ne fonctionne pas comme un ordinateur sur le disque dur où l'information est facilement accessible à la demande.

L'une des premières méthodes de détection de plagiat a été présenté par Bird (1927), qui a enquêté sur l'application de méthodes statistiques pour détecter le plagiat réponses de choix multiples.

En (1960) une autre méthode a été développée où ils ont mis l'accent sur la détection de plagiat dans les tests à choix multiples. Les systèmes de la détection de plagiat précoce pour les textes écrits ont commencé à apparaître dans les années 1990. Ces outils utilisent les méthodes statistiques pour calculer la similarité entre les textes, et la plupart des outils sont déduis pour le plagiat de texte alors que certains concentrés uniquement sur le plagiat du code source.

Dans la dernière décennie, les systèmes commerciaux ont été prospérés grâce à l'augmentation en nombre et les affectations des étudiants. En 2000, il n'y avait que cinq systèmes établies, dont quatre ont été utilisés pour identifier le plagiat du texte et une pour identifier le plagiat du code source (Lathrop et Foss, 2000). Une décennie plus tard, en 2010, 47 systèmes ont été notés (Weber-Wul, 2010). Cette croissance substantielle suggère que le plagiat n'a pas été traité avec efficacité, ainsi de nombreux outils ont été développés pour répondre à l'augmentation de la demande du marché.

Le plus grand défi dans le domaine de la détection de plagiat est que la plupart des approches sont insuffisantes pour détecter des textes avec des changements sémantiques et syntaxiques importantes.

Pour un être humain, il est facile de comprendre des textes qui portent un sens similaire, même quand ils sont réécrits en utilisant des mots et structures différents. Cependant, les ordinateurs sont incapables de comprendre des textes d'une manière similaire, en particulier lorsque la détection repose automatiquement sur le texte exact correspondant. Une solution possible à ce défi réside dans le domaine de la recherche de la linguistique informatique, qui fournit des techniques pour aide à l'analyse linguistique plus profonde. L'utilisation de ces techniques est encore une zone inexplorée dans le champ de détection de plagiat. Afin de faire la lumière sur les approches existant de détection de plagiat. [10]

2.2 Les types de détection de plagiat

L'historique de l'informatique a marqué au fil du temps un développement très croissant des systèmes anti-plagiat. Chaque système est spécifie à un modèle spécial. Pour cela, nous allons citer les différents modèles de la détection de plagiat.

2.2.1 Détection lexicale

Les changements lexicaux comprennent l'addition, la suppression ou le remplacement des mots dans le texte. Un brusque changement de vocabulaire, telles que l'utilisation excessive d'une nouvelle terminologie dans un document, est généralement une bonne indication de copier-coller le plagiat. Un autre exemple est le remplacement mot par mot par des synonymes. Ce type de plagiat est indétectable en utilisant l'approche traditionnelle String-Matching. La détection exige une analyse de l'information lexicale dans tout le texte. [10].

2.2.2 Détection sémantique

Cela implique des changements plus radicaux dans le texte, normalement basé sur la paraphrase lourde qui peuvent inclure des changements à la fois lexicales et syntaxiques. Détecter ce type, nécessiterait l'analyse de l'information sémantique pour juger si deux textes partagent la même signification. Encore une fois, ceci est indétectable avec l'approche traditionnelle. [10]

La majorité des systèmes de détection de plagiat compare les mots et les phrases syntaxiquement, tandis que la possibilité d'échanger les mots par leur synonymes et cela pour introduire le même sens sous une sémantique différente. En utilisant WordNet[3] pour récupérer les synonymes, pour cela le problème de la substitution pourrait être traitée, mais le sens des mots sont ambigus alors la sélection du mot correct est no trivial. [8]

Pour plus de motifs en plagiat sémantique, le système Checker[4] qui calcule la quantité de données copiées à partir du document original en fonction des modèles de plagiat linguistique.

Une autre méthode d'analyse linguistique a été proposée, pour la détection de plagiat en utilisant une analyse syntaxe-sémantique comme la méthode Fingerprinting.

L'analyse syntaxique est effectuée par l'utilisation d'un analyseur pour identifier les règles de grammaire dans les textes et déterminer les structures des textes puis ces structures sont comparées par les règles de grammaire.

[3] https://wordnet.princeton.edu/

[4] https://www.plagramme.com/?gclid=CJHy1byG6cwCFc0y0wodrOAF2g

Ces systèmes tout comme PPChecker utilise WordNet pour retourner les synonymes. Certaines méthodes utilisent des informations statistiques telles que les positions des mots dans le document pour mesurer leur similitude. [8]

2.2.3 Détection syntaxique

Les changements dans les informations syntaxiques sont bien observés dans le réarrangement de la structure de texte. Par exemple : Le réordonnancement des mots ou d'une clause, la transformation des phrases entre la voix active et la voix passive.
Encore une fois il est indétectable en utilisant l'approche traditionnelle String-Matching. La détection nécessiterait l'analyse de la structure syntaxique du texte. [10]

Contrairement à la détection sémantique, la méthode syntaxique ne prend pas en considéra tion le sens des mots, des phrases, ou d'une phrase.
Par exemple, les deux mots « Exactement » et « Egalement » se considère comme un seul mot mais étant deux mots différents. [8]

3. Les mesures de similarité sémantique

La notion de similarité est importante dans presque tous les domaines scientifiques. Ce titre se concentre principalement sur les techniques utilisées pour mesurer la similarité.

3.1 Définition des mesures de similarité sémantique

L'objectif des mesures de similarité sémantique est d'évaluer la proximité sémantique entre les concepts. Le calcul de similarité entre deux concepts permet de déterminer s'ils sont similaires c'est-à-dire s'ils atteignent un certain niveau de ressemblance ou dissimilaire qui peuvent être également liés sémantiquement par des relations lexicales. [11]

Dans le domaine de la recherche de l'information, les modèles de l'espace vectoriel sont largement adoptés. Ces approches utilisent un vecteur caractéristique, dans un espace dimensionnel, pour représenter chaque objet et calculent la similarité en se basant sur la mesure de cosinus ou la distance euclidienne. Le modèle de l'espace vectoriel est employé

pour un arrangement des objets complexes en le représentant comme des vecteurs de k-dimensions. La définition de la similarité entre deux vecteurs d'objets est obtenue par leurs contenus internes. Parmi les approches citées dans la littérature on peut citer : [13]

3.1.1 Similarité de Jaccard

La mesure de similarité de Jaccard est définie par le nombre des objets communs divisé par le nombre total des objets moins le nombre d'objets communs ; sa formule est de :

$$\frac{x.y}{||x||^2 + ||y||^2 - x.y}$$

3.1.2 Similarité de Cosinus

Cette mesure utilise la représentation vectorielle complète, c'est-à-dire la fréquence des objets (mots). Deux objets (documents) sont similaires si leurs vecteurs sont confondus. Si deux objets ne sont pas similaires, leurs vecteurs forment un angle (X, Y) dont le cosinus représente la valeur de la similarité. La formule est définie par le rapport du produit scalaire des vecteurs x et y et le produit de la norme de x et de y.

$$\frac{x.y}{||x||^2.||y||^2}$$

La mesure de Cosinus quantifie donc la similarité entre les deux vecteurs comme le cosinus de l'angle entre les deux vecteurs.

3.1.3 Similarité Euclidienne

La similarité euclidienne est basée sur le ratio de la distance euclidienne augmenté de 1. La distance euclidienne est définie par la formule suivante : $\mathbf{dE= ||x-y||^2}$

La mesure de similarité est donc définie par :

$$\frac{1}{1 + dE}$$

3.1.4 Similarité de Dice

La similarité de Dice est définie par le nombre des objets communs multipliés par 2 sur le nombre total d'objets. La mesure de Dice est donc définie par la formule suivante :

$$\frac{2\,x.y}{||x||^2 + ||y||^2}$$

4. Les approches de la détection de plagiat

Cette section nous permet de voir et de classifier les approches de la détection de plagiat. Nous ordonnons ces approches par leurs types de similarité, globale ou locale.

Le schéma suivant nous montre la classification de ces approches.

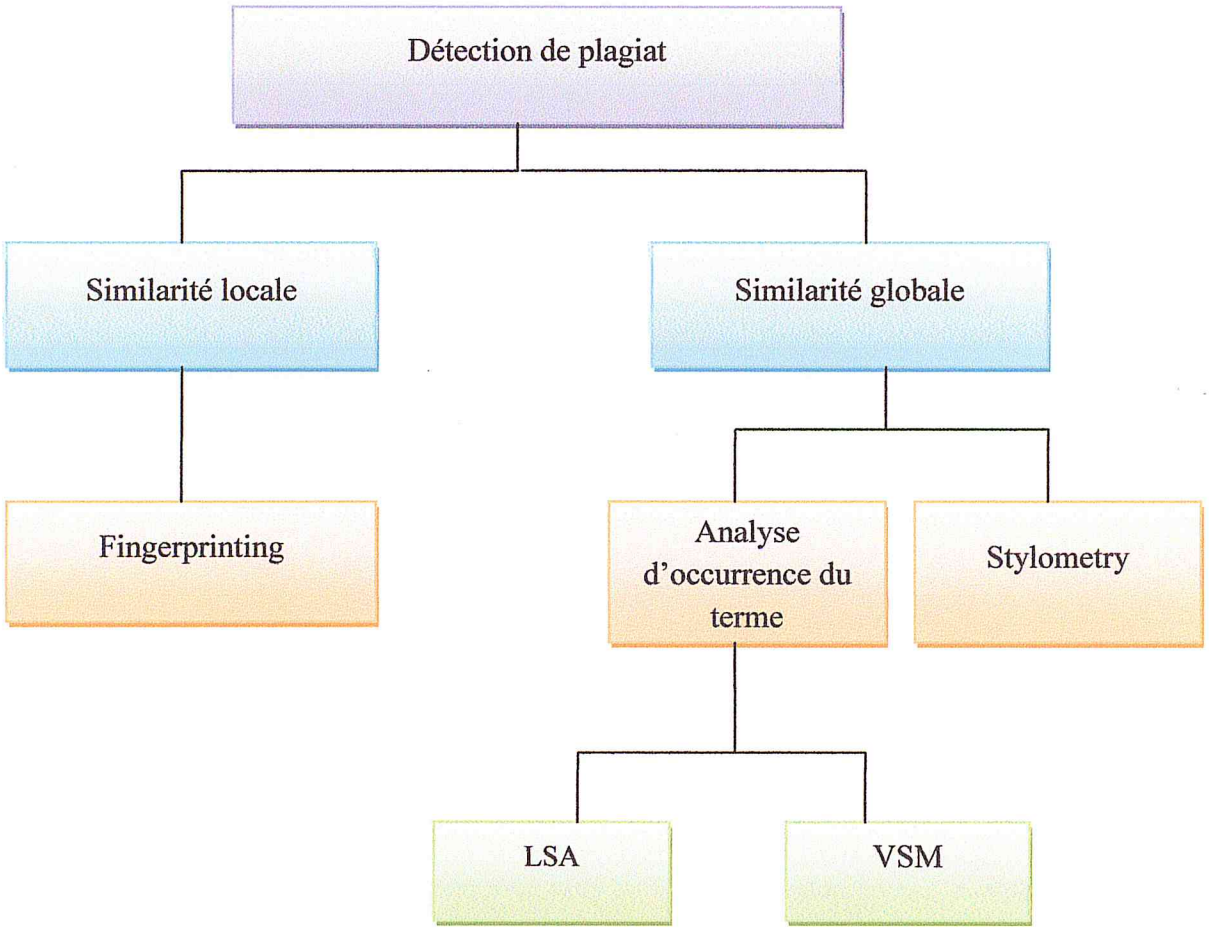


Schéma 1 : *Classification des approches de la détection de plagiat*

4.1 Détection de plagiat par l'évaluation des similarités locales

Les approches d'évaluation de similarité locale analysent les matches des segments de texte dans les textes suspects. L'approche la plus utilisée est le Fingerprinting. [2]

Fingerprinting est un ensemble de nombres entiers créé par hachage des sous-ensembles d'un document pour représenter son contenu essentiel.

Techniquement pour générer des empreintes digitales sont principalement basées sur k-grammes qui servent de base pour la plupart des méthodes d'empreintes digitales. Les fingerprintings sont sélectionnés selon des schémas différents, y compris « i eme hash » [6].

Dans le schéma de « i eme hash », chaque nième hachage d'un document est sélectionné. Ce procédé est facile à mettre en œuvre, mais pas robuste dans le cas d'une insertion, une suppression ou ré ordonnancement. Par exemple, si une lettre est insérée dans le texte alors les empreintes digitales serrent décalée par une lettre. Ce qui rend l'altération et les documents originaux ne partagent aucune empreinte digitale. Par conséquent, la copie ne sera pas détecter. **[6].** Cette méthode est également facile à mettre en œuvre, mais faible en termes des cas de détection de plagiat.

4.2 Détection de plagiat par l'évaluation des similarités globales

L'évaluation de la similarité des approches globales examine les caractéristiques de section des plus long textes, ou le document complet, et exprimer le degré auquel deux documents sont semblables les uns aux autres dans leur ensemble [2].
La détection de plagiat dans l'approche globale englobe les sous approches comme Vector Space Model et Stylométrie.

4.2.1 Analyse d'occurrence des termes

Nous distinguons deux types d'approches dans l'analyse d'occurrence des termes, comme suite l'analyse sémantique latente et modèle d'espace vectoriel, que nous définissons dans les sections qui suit.

4.2.1.1 L'analyse Sémantique Latente (LSA)

L'analyse Sémantique Latente ou Indexation Sémantique Latente est développé à la fin des années 80, dans l'laboratoire de Bell (Deerwester et al., 1990). Est un procédé de traitement des langues naturelles, dans le cadre de la sémantique vectorielle. [14]

La méthode LSA qui s'appuie sur l'hypothèse « harrissienne », est fondée sur le fait que des mots qui apparaissent dans le même contexte sont sémantiquement proches. Le corpus est représenté sous forme matricielle. Les lignes sont relatives aux mots et les colonnes représentent les différents contextes choisis (un document, un paragraphe, une phrase, etc.). Chaque cellule de la matrice représente le nombre d'occurrences des mots dans chacun des contextes du corpus. Deux mots proches au niveau sémantique sont représentés par des vecteurs proches. La mesure de proximité est généralement définie par le cosinus de l'angle entre les deux vecteurs.

La théorie sur laquelle s'appuie LSA est la décomposition en valeurs singulières (SVD). Une matrice A = [a ij] où aij est la fréquence d'apparition du mot i dans le contexte j, se décompose en un produit de trois matricesUSV^T. U et V sont des matrices orthogonales et S une matrice diagonale. La figure 1 représente le schéma bien connu d'une telle décomposition où r représente le rang de la matrice A.
Soit S_K où k < r la matrice produite en enlevant de S les r – k colonnes qui ont les plus petites valeurs singulières. Soit U_K et V_k les matrices obtenues en enlevant les colonnes correspondantes des matrices U et V. La matrice $U_K\ S_K\ V_T^K$ peut alors être considérée comme une version compressée de la matrice originale A.

Il est coutume de dire que LSA est une méthode statistique ou numérique car elle s'appuie sur une théorie mathématique bien connue. Cependant, on peut également dire que LSA est une méthode géométrique car seuls des résultats d'algèbre linéaire sont utilisés. Nous précisons qu'avant d'effectuer la décomposition en valeurs singulières, nous effectuons une première étape de normalisation de la matrice d'origine A. Cette normalisation consiste à appliquer un logarithme et un calcul d'entropie sur la matrice A. Ainsi, plutôt que de se fonder directement sur le nombre d'occurrences de chacun des mots, une telle transformation permet de s'appuyer sur une estimation de l'importance de chacun des mots dans leur contexte. De manière similaire aux travaux de Turney (2001), cette étape de normalisation peut également s'appuyer sur la méthode du tf×idf, approche bien connue dans le domaine de la Recherche d'Information (Salton (1991)). [14]

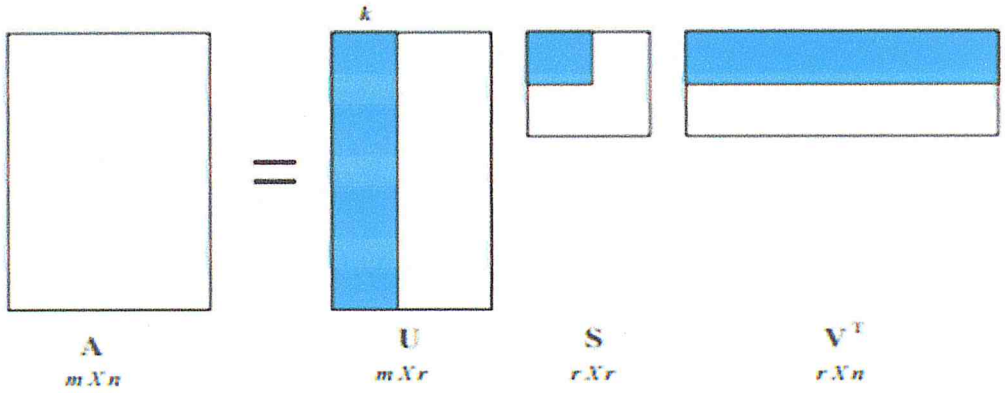


Figure 01 : *Décomposition en valeurs singulières.*

La matrice A représente le corpus d'origine de m lignes (mots du corpus) et n colonnes (contextes).

4.2.1.2 Modèle d'espace vectoriel (VSM)

Le modèle vectoriel représente un document, ainsi qu'une requête, par un vecteur dans un espace dont chaque dimension correspond à un descripteur atomique. Chaque coordonnée dans cet espace dénote l'importance du descripteur dans le document considéré. Le traitement d'une requête est alors basé sur la comparaison des vecteurs documents et requête.

Dans ces modèles, la pertinence d'un document vis-à-vis d'une requête est définie par des mesures de distance dans un espace vectoriel.
Le modèle vectoriel représente les documents et les requêtes par des vecteurs d'un espace à n dimensions, les dimensions étant constituées par les termes du vocabulaire d'indexation.

L'index d'un document dj est le vecteur = (w1,j ,w2,j ,w3,j , ..,wn,j), où wk,j ∈ [0 1] dénote le poids du terme tk dans le document dj .
Une requête est également représentée par :

Un vecteur = (w1, q, w2, q, w3, q,.., wn, q), où wk, q est le poids du terme tk dans la requête q. La fonction de correspondance mesure la similarité entre le vecteur requête et les vecteurs documents.

Le poids d'un terme représente à la fois son importance dans le document (ou dans la requête), et le fait qu'il est discriminant ou non. Ces critères sont souvent exprimés par le calcul des valeurs de tf ou fréquence de terme, et d'idf ou fréquence documentaire inverse. Par exemple :

$$\mathbf{TF\ i,\ j} = \frac{freq\ i,j}{\max i\ freqi,j} \qquad \mathbf{idf\ i} = \log \frac{N}{ni}$$

Où dans l'équation (1) Freq (i, j) est le nombre d'occurrences du terme t (i) dans le document d(j) et max(i) est le nombre d'occurrences du terme le plus fréquent dans le document d(j). Et dans l'équation (2) N est le nombre de documents dans le corpus et n(i) le nombre de documents dans lesquels le terme t(i) apparaît.

Le tf mesure l'importance d'un terme dans un document, et l'idf donne une mesure de la discriminance d'un terme. Il existe différentes manières de calculer les poids des termes. Les méthodes de pondération les plus courantes et les plus efficaces pour les documents textuels utilisent une combinaison des tf et idf comme les formules w(i,j) = tf (i,j) * idf (i) pour les documents, et w(i, q)= idf(i) pour les requêtes, éventuellement complétées par un facteur de normalisation appliqué aux vecteurs. [26]

Une mesure classique utilisée dans le modèle vectoriel est le cosinus de l'angle formé par les deux vecteurs : RSV (q, d) = cos q d.

Plus les vecteurs sont similaires, plus l'angle formé est petit, et plus le cosinus de cet angle est grand. A l'inverse du modèle booléen, la fonction de correspondance évalue une correspondance partielle entre un document et une requête, ce qui permet de retrouver des documents qui ne reflètent pas la requête qu'approximativement. Les résultats peuvent donc être ordonnés par ordre de pertinence décroissante. [9]

Le modèle vectoriel classique possède les caractéristiques suivantes :

- Il ne suppose pas que les termes du vocabulaire sont liés,
- Un document - ou une requête - est indexé par un et un seul vecteur,

- Le processus d'obtention des termes du vocabulaire des documents et requêtes sont similaires, même si la pondération est différente. [26]

4.2.2 La Stylométrie

A la croisée de la linguistique et de la statistique, la stylométrie tente d'identifier le style d'un texte, voire d'un auteur. Les développements de l'informatique ont renouvelé l'intérêt pour cette discipline ancienne. La stylométrie peut être utilisée pour analyser des textes anciens, dont l'origine est incertaine.

L'approche intrinsèque pour la détection de plagiat construit et compare des modèles qui permettent de quantifier les caractéristiques du style d'écriture de l'auteur pour les segments individuels d'un texte. Le but est d'identifier les sections qui sont stylistiquement différent des autres sections, et les indicateurs ainsi potentiels de plagiat. [3]

Les principales caractéristiques stylométrie linguistiques sont :

- Statistiques de texte qui fonctionnent au niveau de caractère (Nombre de virgules, points d'interrogation, mot longueurs, etc.)
- Caractéristiques syntaxiques pour mesurer le style d'écriture au niveau d'une phrase (la longueur des phrases, l'utilisation de la fonction mots, etc.)
- Caractéristique « part of speech » pour quantifier l'utilisation des adjectifs et les pronoms.
- Compter les mots spéciaux (Nombre de mots d'arrêts, les mots étrangers, les mots difficiles).
- Les caractéristiques structurelles qui reflètent l'organisation du texte (longueurs de paragraphe, les longueurs de chapitre, etc.).

L'approche stylométrie est pas couramment utilisée, cela est parce qu'il est difficile de prouver le plagiat sans preuves à partir des documents sources. Néanmoins cette approche pourrait fournir une indication à laquelle les documents sont susceptibles d'être plagié et donc utilisé pour comparaison ultérieure. [5]

5. Les systèmes existants pour la détection de plagiat

Pour lutter contre ce phénomène grandissant, plusieurs logiciels sont apparus ces derniers temps. Ils ont des caractéristiques sensiblement identiques, mais leur efficacité varie de façon considérable. Pour permettre une évaluation des logiciels anti plagiat, il semble nécessaire de faire plusieurs distinctions. Premièrement par leurs types de fonctionnement :
Ceux qui travaillent sur un serveur distant et ceux qui peuvent être installés directement sur la machine de l'utilisateur, et utilisés en local.

Premièrement les logiciels semble à priori le plus efficace car il dispose bien souvent d'une base de données de référence gigantesque, qui s'enrichit à chaque fois qu'un nouveau document lui est soumis pour analyse, le serveur l'incorporant alors à ses documents de référence. A noter aussi l'intérêt d'avoir la possibilité pour le professeur de demander à ses étudiants de rendre les devoirs sur une adresse mail spécifique : cela facilite grandement le travail du professeur qui, une fois le délai dépassé, reçoit un compte rendu global. [7]

Ensuite par la langue de fonctionnement et le coût du système, où on trouve la langue française ainsi que d'autre langue étrangère. Parmi les logiciels gratuits, On trouve par exemple : CopyTracker, Plagium et Noplagia. Pour ceux qui sont payants, nous citons : Compilatio, Urkund, Essay Verification Engine (EVE2) et Turnitin.

Nous allons présenter quelques logiciels pour la détection de plagiat.

5.1 Plagium

Est un logiciel pour la détection du plagiat dans ces différentes langues tels que : anglais, français, espagnole, portugais, italien et Deutsch. Il est connu par sa gratuité, il est bien installé sur un serveur externe. Le logiciel de détection de plagiat « Plagium » prend 5000 caractères limité.

On représente dans la figure suivante la fenêtre de Plagium. [7]

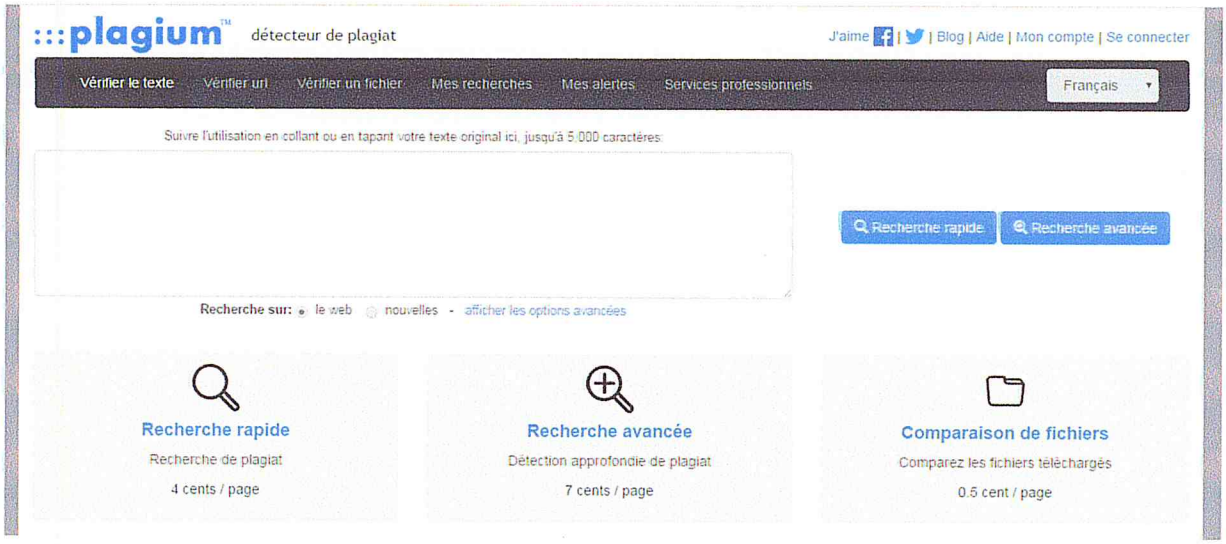


Figure 02 : *L'nterface du logiciel de detection de plagiat « Plagium »* [7]

5.2Noplagiat

Le logiciel de détection de plagiat « noplagiat » est un logiciel libre écrit en Perl. Ce logiciel détecte le plagiat en langue française. Il est installé sur un serveur externe. La figure suivante représente l'interface de ce logiciel.

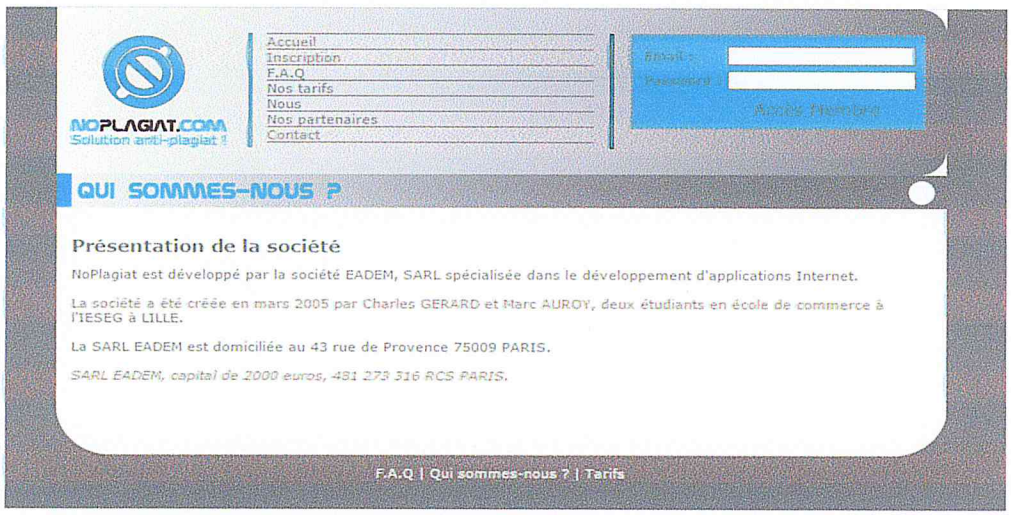


Figure 03: *L'nterface du logiciel de detection de plagiat « noplagiat »* [7]

5.3 Compilatio.net

Est un logiciel de détection de plagiat pour les langues suivantes Française, Anglaise, Deutsch, Italienne, Espagnole. Il est parmi les logiciels payants de détection de plagiat. Il est installé sur un serveur externe.

Ce logiciel est dédié pour les enseignants et les étudiants dans les universités ainsi que pour les entreprises et les auteurs. La licence nous donne un login et un mot de passe, Il suffit alors d'entrer les documents à analyser dans la base de données.

Puis après avoir sélectionné les documents que l'on souhaite analyser, on lance l'analyse. Le logiciel « Compilatio.net » analyse tous types de documents (".doc", ".rtf", ".ppt", ".pdf", ".xls", ".txt", ".odt", ".html", etc.) Et génère un rapport indiquant entre autre :

- les passages identifiés comme «copiés sur Internet»
- l'ensemble des sources de plagiats possibles. [7]

Nous allons par la suite représenter la figure qui montre l'interface du logiciel de détection de plagiat « Compilatio.net »

Figure 04 : *L'interface du logiciel de detection de plagiat « Compilatio.net »* [7]

5.3 Essay Verification Engine (EVE2)

EVE2 est un outil très puissant qui permet aux professeurs et enseignants à tous les niveaux du système d'éducation afin de déterminer si les élèves ont plagié du World Wide Web. EVE2 accepte des essais en texte brut, Microsoft Word ou Corel Word Perfect et renvoie des liens vers des pages Web à partir de laquelle un étudiant peut avoir plagié. EVE2 a été développé pour être assez puissant pour trouver du matériel plagié sans surcharger le professeur avec de faux liens.

EVE2 aussi proche que possible en utilisant des outils de recherche les plus avancées pour localiser les sites suspects. Non seulement elle trouver ces sites suspects, mais il fait alors une comparaison directe de l'essai présenté au texte figurant sur le site suspect. Si elle trouve des preuves de plagiat, l'URL est enregistrée. Une fois la recherche terminée, l'enseignant reçoit un rapport complet sur chaque document qui contenait le plagiat, y compris le pourcentage de l'essai plagié, et une copie annotée du document montrant tout le plagiat en surbrillance rouge. La figure suivante montre l'interface du logiciel de détection de plagiat « EVE2 » [15].

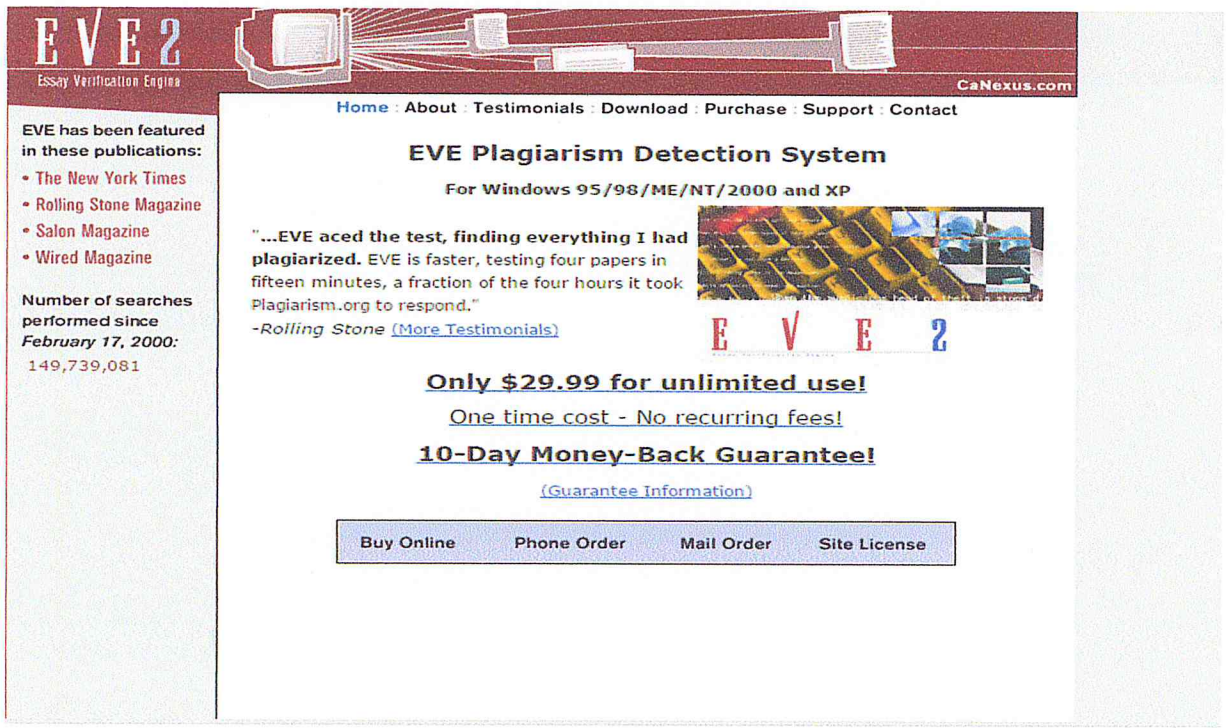


Figure 05 : *L'interface du logiciel de detection de plagiat « EVE2 »* [7]

3. Conclusion

Durant ce chapitre nous avons pu recenser toutes les informations nécessaires et indispensables sur les approches pour la détection de plagiat.

Ces informations tirées entre autre à partir de l'étude de ses approches qui nous permettrons de cerner les principaux objectifs de choisir la technique Latent Semantic Analysis adoptée dans le suivant chapitre conception.

CHAPITRE II

La détection du plagiat dans les textes arabes :

Introduction :

La langue arabe a connu une augmentation très importante dans le volume des collections de documents, par conséquent cela nécessite des systèmes de recherche d'information efficaces afin de pouvoir détecter de plagiat dans cette langue. Mais la question qui se pose est :est-ce que le changement de la langue pour un système de détection plagiat implique une différence au niveau du code et de la méthode appliquer ?

Pour répondre à cette question nous allons étudier les différentes caractéristiques de la langue arabe.

Alors, ce chapitre est organisé comme suit : nous commençons par la description des caractéristiques et particularités de la langue arabe afin de dégager les problèmes de cette langue.

Nous abordons par la suite, les différents travaux concernant la détection de plagiat dans la langue arabe.

1-Les caractéristiques de la langue arabe :

L'arabe, une des six langues officielles des Nations Unies, est la langue maternelle de plus de 300 millions de personnes. [16]

L'Arabe, langue sacrée du Coran, connaît une grande stabilité dans un créneau bien précis qui est celui de la littérature classique, des milieux de l'enseignement, la culture officielle et de la presse. C'est l'Arabe standard ou littéraire, universellement partagé par les lettrés de tous les pays arabes. Par contre, parallèlement à cette lignée, il existe de nombreuses branches s'écartant plus ou moins de la norme. L'Arabe dialectal dans toutes ses variétés, essentiellement oral, et le moyen Arabe (état intermédiaire entre le dialectal et le classique) essentiellement écrit, sont autant de réalisations différentes d'une même source. Suffisamment proches pour constituer une seule et même langue, suffisamment éloignées pour ne pas s'intégrer dans les mêmes systèmes de traitement automatique. [17]

1.1-Particularité de la langue arabe :

La langue arabe est composée de 28 lettres (voir tableau1) (25 consonnes et 3 longues voyelles), les voyelles courtes n'étant pas représentées par des lettres mais par des

diacritiques, placées sur ou sous les consonnes. Les lettres sont monocamérales, dans le sens où il n'existe pas de minuscule et de majuscule.

Lettre Arabe	Correspondant français	Lettre Arabe	Correspondant Français	Lettre Arabe	Correspondant français
أ	A	ر	R	غ	GH
ب	B	ز	Z	ك	K
ت	T	س	S	ق	Q
ث	TH	ش	SH	ل	L
ج	J	ص	S	م	M
ح	H	ض	D	ن	N
خ	KH	ط	T	ه	H
د	D	ظ		و	W
ذ	DH	ع	A	ي	Y
ف	FA				

Tableau 1: *Les 28 lettres arabes.*

L'Arabe s'écrit de droite à gauche avec la particularité que les lettres épousent des formes différentes selon qu'elles soient au début, au milieu ou à la fin du mot. Le tableau 2 illustre le script de quelques lettres dans les trois cas de graphie. Cependant, Il faut noter que certaines lettres ne s'attachent pas à celles qui la succèdent comme. {د، ر، س، ص}

A la fin du mot	Au milieu du mot	Au début du mot
أ ، ؤ، ئ، ء	ـئـ	أ
ب، ـب	ـبـ	بـ
ه، ـه	ـهـ	هـ
م، ـم	ـمـ	مـ
ي، ـي	ـيـ	يـ
غ، ـغ	ـغـ	غـ

Tableau 2: *Etat de transcription des lettres arabes.*

Des petits points noirs ont été utilisés comme marques de différenciation entre des lettres qui partageaient une forme identique. Ces points sont placés au-dessus et au-dessous de la lettre en un, deux ou trois. Exemples : خ , ج ; ف , ق ; ب, ت , ث , etc.

Pour une meilleure précision de la prononciation, des signes ont été inventés. Il s'agit de trois voyelles brèves et de sept signes orthographiques qui s'ajoutent aux consonnes. Ces trois voyelles brèves sont :

- Fatha «ó», elle surmonte la consonne et se prononce comme un «a» français.
- Damma «ó», elle surmonte la consonne et se prononce comme un «ou» français.
- Kasra « ِ», elle se note au-dessous de la consonne et se prononce comme un « i » français.

Les sept signes orthographiques sont :

- Sukun «ْ » : ce signe indique qu'une consonne n'est pas suivie (ou muet) par une voyelle. Il est noté toujours au-dessus de la consonne ;

Les trois signes de tanwin : lorsque (la Fatha, la Kasra et la Damma) sont doublées, elles prennent un son nasal, comme si elles étaient suivies de « n » et on les prononce respectivement :

- an «ً» pour les Fathatan ;
- in «ٍ» pour les Kasratan ;
- un «ٌ» pour les Dammatan.
- Chadda «ّ » comme dans le français, l'arabe peut renforcer une consonne

Quelconque.

Wasla « ٱ » : quand la voyelle d'un Alif au commencement d'un mot doit être absorbée par la dernière voyelle du mot qui précède ;

- Madda « آ » : la madda (prolongation) se place sur l'Alif pour indiquer que cette lettre tient lieu de deux alifs consécutifs ou qu'elle ne doit pas porter le Hamza.

Cependant, les textes courants rencontrés dans les journaux et les livres ne comportent habituellement pas de voyelles. De plus, certaines lettres comme Alif « ا » peuvent symboliser le « أ », « إ », « آ » ou « ٱ »; de même que pour les lettres « ى » et « ه » qui symbolisent respectivement « ي » et « ة ». [18]

1.2. La structure morphologique d'un mot arabe :

La définition du mot du point de vue du traitement automatique se heurte à des considérations syntaxiques et sémantiques. Dans le domaine des langages formels, la transformation du flux de caractères représentant un texte en une suite d'unités mieux adaptées aux traitements ultérieurs, est habituellement appelée segmentation (tokenization), et les unités produites les segments (tokens) sont construites sur la base de définitions purement orthographiques. En arabe cette séquence de lettres est appelée le mot graphique (MG). « Le mot graphique est facile à identifier : c'est ce qui s'écrit en un seul bloc entre deux blancs. » [19]

Un MG Arabe (MGA) peut être soit simple, soit complexe. Un MGA simple est un mot attesté de la langue, il est formé par la concaténation d'une base avec d'éventuels affixes (préfixes et suffixes). Il ne constitue pas un mot attestable de la langue sans les affixes. [20]

MGA simple = Préfixes + Base + Suffixes

Un MGA complexe est formé par la concaténation d'un mot simple et un ensemble de clitiques (proclitiques et enclitiques).

MGA complexe = Proclitiques # mot simple # Enclitiques

MGA complexe = Proclitiques # Préfixes + Base + Suffixes # Enclitiques

Ou : MGA complexe = Prébases + Base + Postbases

Avec : Prébases=Proclitiques # Préfixes ; Postbases=Suffixes # Enclitiques.

L'Arabe est une langue générative, les noms et les verbes sont dérivés d'une racine, généralement, trilitère. Nous pouvons engendrer jusqu'à 150 mots différents à l'aide de schèmes et à partir d'une même racine. Le tableau 3 donne quelques schèmes du mot «شهد ».

Schème	شهد	
فَعَلَ	شَهَدَ	Il a témoigné
فَعِلَ	شَهِدَ	Il a assisté
فَاعَلَ	شَاهَدَ	Il a regardé
فَاعِلٌ	شَاهِدٌ	Témoin
مَفعِلٌ	مَشهِدٌ	Scène
فُوعِلَ	شُوهِدَ	Il a été vu
فَعالَةٌ	شَهَادَةٌ	Témoignage. Certificat
فَعِيلٌ	شَهِيدٌ	Martyr

Tableau 3: *Exemple des schèmes.*

Plusieurs types d'affixes sont agglutinés au début et à la fin des mots : antéfixes, préfixes, suffixes et post fixes (voir tableau4).

Antéfixe	Préfixe	Noyau	Suffixe	Post fixe

Tableau 4 : *Structure d'un mot.*

1.2.1. Les antéfixes

Les antéfixes sont généralement des prépositions agglutinées au début des mots. Ils se combinent entre eux pour donner les traits syntaxiques, coordonnant, terminant ...etc.

Voici une liste non exhaustive des antéfixes simples.

- La coordination par les coordonnants « ف » fa et « و » wa.
- L'interrogation par le morphème « أ » a.
- La marque du futur « س » sa.
- L'article « ا ل » al.
- Les prépositions par les lettres « ب » bi et « ل » li.
- Les particules du subjonctifs النصب « ف » fa, « ل » li, et « و » wa.

Le marqueur de comparaison par les lettres « ك » ka.

- Le marqueur de corroboration « ل » la.
- La particule du jussif (الجزم) par la lettre « ل » li.

1.2.2. Les préfixes

Les préfixes (voir tableau 5), habituellement représentés par une seule lettre, indiquent la personne de conjugaison des verbes au présent.

Numéro de préfixe :	Préfixe :
1	أَ
2	أَ
3	تَ
4	تُ
5	نَ
6	نُ
7	يَ
8	يُ

Tableau 5: *listes des préfixes arabes.*

1.2.3. Les suffixes

Les suffixes sont les terminaisons de conjugaison des verbes et de marques duelles/plurielles/femelles pour les noms y compris les adverbaux. Ils ne se combinent pas entre eux. Voici la liste (tableau 6) exhaustive de tous les suffixes :

ا	ت	ك	و	يات
ات	ة	كم	وا	ية
اتكم	تان	ما	ون	يتنا
اتنا	تم	نا	ونن	يتها
اته	تموها	ني	ونه	ين
اتها	تنا	هـ	وه	يه
اتهم	ته	ها	وها	يها
انية	تها	هم	وهم	يون
اها	تين	هن	يا	يين
ان	تهم	هما	ي	

Tableau 6: *listes des suffixes arabes.*

1.2.4. Les post fixes

Finalement, les post fixes (voir tableau 7) représentent des pronoms attachés à la fin des mots. Ils peuvent se combiner entre eux. Voici dans le tableau suivant une liste des post fixes:

N° de post fixe :	Post fixe :	Description :
1	كُمَا	2eme Personne, Masculin/Féminin, Duel
2	كُمْ	2eme Personne, Masculin, Pluriel
3	كُنَّ	2eme Personne, Féminin, Pluriel
4	هُ	3eme Personne, Masculin, Singulier
5	هَا	3eme Personne, Féminin, Singulier
6	هُمَا	3eme Personne, Masculin/Féminin, Duel
7	هُمْ	3eme Personne, Masculin, Pluriel
8	كِ	2eme Personne, Féminin, Singulier
9	كَ	2eme Personne, Masculin, Singulier
10	نَا	1ier Personne, Masculin/Féminin, Duel/Pluriel

Tableau7: *listes des post fixes arabes.*

1.3-Les catégories du mot

Il existe trois catégories pour un mot arabe : nom, verbe et particule.

1.3.1. Le verbe :

Le verbe est une entité qui exprime un sens variant en nombre, en personne et en temps, exemple : « كتب » sa conjugaison dépend du temps, du nombre, du genre, de la personne et du mode, il peut donc être exprimé à l'accompli ou l'inaccompli, au singulier, duel ou pluriel, au masculin ou au féminin, au premier, deuxième ou troisième type et être au mode actif ou inactif.

Nous pouvons classer les verbes arabes selon plusieurs critères : Selon le nombre et la nature des consonnes de leurs racines, et selon leurs modèles. En classant les verbes selon le nombre des consonnes de la racine, nous aurons soit des verbes trilitères qui ont trois consonnes, soit des verbes quadrilatères, peu nombreux, qui ont quatre consonnes.

Selon le modèle et le nombre de consonnes qui constituent la structure verbale, nous avons soit des verbes nus (مجرد) qui sont composés seulement par les consonnes de leurs racines et des voyelles brèves, soit des verbes augmentés ou dérivés (مزيد) qui sont dérivés de trois consonnes de la racine par modification des voyelles, par redoublement de la deuxième lettre de la racine, par adjonction et même par intercalation d'affixes.

La conjugaison des verbes dépend de plusieurs facteurs :

- Le temps (accompli, inaccompli).
- Le nombre du sujet (singulier, duel, pluriel).
- Le genre du sujet (masculin, féminin).
- La personne (première, deuxième et troisième).
- Le mode (actif, passif).

1.3.2. Le nom

Le nom est un élément désignant un être ou un objet qui exprime un sens indépendamment du temps, exemple : « مكتب ». Le nom peut être propre, commun ou dérivé d'un verbe. Il s'exprime au singulier, au duel ou au pluriel, au féminin ou au masculin. Il peut être agent, objet, instrument ou lieu.

1.3.3. La particule

La particule est une entité qui sert à situer les événements par rapport au temps et par rapport à l'espace. Elles peuvent être des conjonctions de coordination « ... و, أو, أم » ou de subordination « .... إذا, لأن ». Les particules sont généralement des mots outils, bien que jouant un rôle important dans la cohésion d'une phrase, sont souvent associées à des mots vides qui ne véhiculent pas un sens spécifique à un domaine donné. On distingue plusieurs types :

- Préposition : exemple (عن, حتى)

- Particules de coordination : exemple (و, أو, ثم)
- Particules interrogatives : exemple (هل,ما)
- Particules d'affirmation : exemple (أجل, بلى, نعم)
- Particules de négation : exemple (لا, لن, لم)
- Particules distinctive : exemple (أي)
- Particules relatives : exemple (ما)
- Particules de futur : exemple (سوف)
- Particules conditionnelles : exemple (لو, إن)

Ces particules seront très utiles pour notre traitement, elles font partie du dictionnaire qui regroupe les mots vides.

Les particules peuvent avoir des préfixes et suffixes ce qui rajoute une complexité quant à leur identification.

2- Les problèmes liés au traitement automatique de l'arabe

Vu ses particularités, le traitement automatique de l'Arabe, fait face à un certain nombre de problèmes, les plus importants sont le problème de la voyellation, l'agglutination et l'extraction de la racine.

2.1- Le problème de la voyellation

L'absence de la voyellation est très souvent une grande source d'ambiguïté pour l'analyse morphologique, syntaxique, sémantique et même pragmatique. La majorité des textes écrits, exception faite pour les textes sacrés et quelques ouvrages pédagogiques, sont non voyellés.

Cette ambiguïté réside dans le fait que 74% des mots qui composent le vocabulaire arabe, acceptent plus d'une voyellation lexicale, et 89,9% des noms qui le constituent acceptent plus d'une voyellation casuelle. La proportion des mots ambigus passe de 90,5% si les comptages portent sur leurs voyellations globales. [21]

Si le problème est aussi commun au Français où 28 % des mots sont ambigus à cause de l'absence d'accentuation, en arabe la proportion est bien plus grande, en effet, l'ambiguïté touche 95% des mots. [22]

2.2. Le problème de l'agglutination

Une grande partie des mots arabes sont générés en agglutinant des proclitiques et des enclitiques à un radical. Pour déterminer un nom, par exemple, on ajoute (ا ل = al), comme dans le mot « الشمس » (Le soleil). Les pronoms personnels peuvent se rattacher aux noms

(آياته =ses signes), comme aux verbes (أنزله = il l'a révélé). Les particules aux noms (كالمجرمين =sur le même pied d'égalité que les criminels), les conjonctions de coordination aux verbes (فتولى =et il se retira). Le problème, dans le cadre du traitement automatique de l'Arabe, est de pouvoir bien décomposer le mot en ses différentes parties. [23]

2.3. L'extraction de la racine

Afin d'obtenir la racine d'un mot, il faut d'abord connaître le schème par lequel il a été dérivé, supprimer les éléments flexionnels (antéfixes, préfixes, suffixes, post fixes) qui lui sont attachés. En général des tables de préfixes et de suffixes sont utilisées. La nature agglutinative de l'Arabe rend cette tâche, assez difficile. Cette difficulté est encore plus accrue, lorsqu'il s'agit de textes non voyellés.

L'analyse morphologique devra donc découper le mot et identifier des préfixes comme les conjonctions (و = et) et (ك = puis), des prépositions comme (ب = avec) et (ل =pour), l'article défini (ال = le, la, les) et des suffixes de pronom possessif (ه =à lui, ها = à elle, هم =à eux, هن = à elles) etc.

La phase d'analyse morphologique détermine un schème possible. Les préfixes et suffixes sont trouvés en enlevant progressivement des préfixes et des suffixes et en essayant de faire correspondre toutes les racines produites par un schème afin de retrouver la racine ». [24]

2.4. La terminologie

Le problème de terminologie dans la langue arabe cherche toujours sa solution. Il suffit de prendre comme exemple quelques termes linguistiques et informatiques improvisés sous plusieurs équivalents dans les différents pays arabes. Il est clair que ce problème engendre une autre difficulté dans le traitement automatique de l'Arabe [25].

3-Les travaux de la détection de plagiat dans langue arabe

La langue arabe est connue comme une langue riche sémantiquement, un mot peut avoir plusieurs sens selon leur contexte d'utilisation, à cet effet, la détection de similarité devient une tâche très importante mais très difficile à réaliser. Dans ce domaine, nous trouvons peu de travaux de détection plagiat arabe. Nous pouvons citer :

3.1- Méthode de l'outil Aplag (Mohamed EL-BACHIR 2012) [26]

Cette méthode est créée par « MOHAMED ELBACHIR ELMANAI » elle se base sur la

comparaison entre deux textes dans la langue arabe : un texte source et un autre texte qui peut contenir du plagiat. Cette méthode passe par trois phases:

3.1.1 -pré-processing :

Ils ont proposé un traitement en suivant ces étapes :

tokenzation : dans cette étape ils ont divisé le texte et séparé les mots par les virgules.

stop-word removal : utilisé dans le texte pour la suppression des stop-Word.

Rooting : consiste à retourner les mots à son origine, ils utilisent l'outil de khoja [21] qui supprime les préfixes et les suffixes.

Synonyme : utilise le WORD-NET (les mots de même sens).

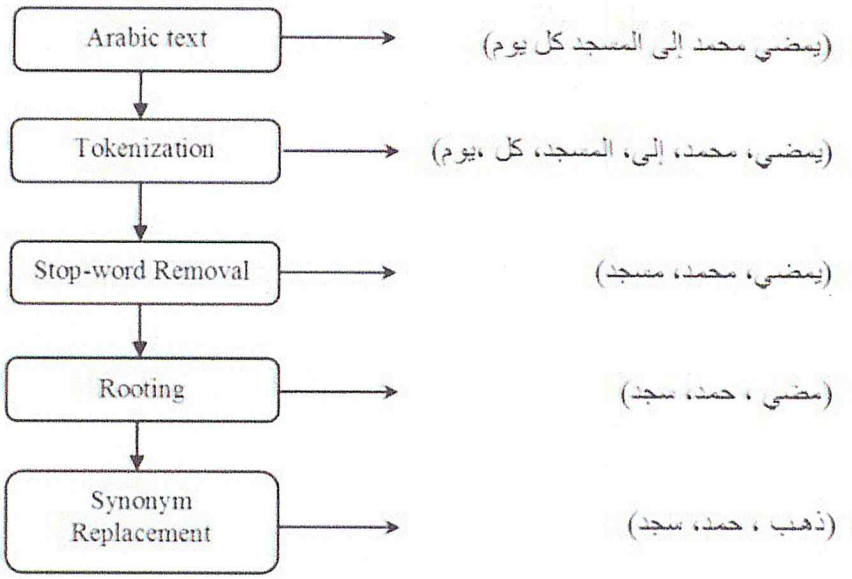


Figure 1: Un exemple de pre-processing d'APLAG [27].

3.1.2 Le fingerprinting :

Le fingerprinting est utilisée à la base de la de la fonction « chunking » qui permet de découper le document à petites pièces. L'unité de segmentation peut être une phrase ou un mot.

Exemple :

On considère la phrase l'unité de la pièce, le paramètre N où N=4

X1 x2 x3 x4 x5 x6 x7 les pièces sont x1 x2 x3 x4 ,x2 x3 x4 x5 ,x3 x4 x5 x6 , x4

x5 x6 x7.

Et puis une fonction de hachage pour minimiser les collisions entre les différents chunks hachées, dans le cas de APLAG ils ont utilisés la fonction « BKDR : Brian Kernighan And Dennis Ritchie » qui retourne la somme des multiplications de chaque caractère par une valeur spéciale qui s'appelle « Seed Value ».

Pour la métrique de similarité en finregprinting ils ont utulisé le LCS :longest common substring qui consiste a trouver la plus longue et commune chaine entre deux chaine.

Une valeur de hachage sera créée pour chaque chaine de caractère en comparant avec la chaine de texte donné.

3.1.3- Calcul de similarité:

Dans APLAG on utilise L'algorithme heuristique pour tester l'existence de similarité pour chaque niveau d'arbre

Pour les documents :

```
Algorithme1 : détection dans le document
Debut
    DocMinSize ← min (|DocA|,|DocB|) ;
    DocIntersectionSize ← |DocA ∩ DocB| ;
    Si (DocIntersectionSize ≥ DocMinSize*DocThershold) Alors
        Similarity ← vrai ;
    Sinon
        Similarity ← faux ;
    Finsi ;
    Retourner(similarty) ;
Fin.
```

Algorithme 01 : *détection de similarité dans un document.*

Où DocA et DocB sont deux documents, l'un est la source et l'autre est le soupçonné

DocMinSize:la taille minimum entre deux documents

DocIntersectionSize:la taillade document commun des DocA et DocB

DocThreshold :(0<threshold <1)

si Similarité=Vraie alors il passe à l'algorithme 2 ;

Pour les paragraphes :

```
Algorithme1 : détection dans le paragraphe
Debut
        ParMinSize ← min (|ParA|,|ParB|) ;
        ParIntersectionSize ← |ParA ∩ ParB| ;
        Si (ParIntersectionSize ≥ ParMinSize*ParThershold)
        Alors
                Similarity ← vrai ;
        Sinon
                Similarity ← faux ;
        Finsi ;
        Retourner(similarty) ;
Fin.
```

Algorithme 02 : *détection de similarité dans un paragraphe.*

Où ParA et ParB sont deux paragraphes, l'une est la source et l'autre est le soupçonné,

ParMinSize: la taille minimum entre deux paragraphes

ParIntersectionSize: la taille de paragraphe commun des ParA et ParB

Threshold :(0<threshold <1)

Si similarité= Vraie alors il passe à l'algorithme 3;

Pour les phrases :

Algorithme1 : détection dans la phrase

Debut

PhMinSize ← min (|PhA|,|PhB|) ;

PhIntersectionSize ← |PhA ∩ PhB| ;

Si (PhIntersectionSize ≥ PhMinSize*PhThershold) Alors

Similarity ← vrai ;

Sinon

Similarity ← faux ;

Finsi ;

Retourner(similarty) ;

Fin.

Où: PhA et PhB sont deux phrases, l'une est la source et l'autre est le soupçonné,

PhMinSize:la: la taille minimum entre deux phrases.

PhIntersectionSize:la taille de phrase commun des PhA et PhB

LongestCommonSeq : la langue substring commun entre deux phrases.

Threshold : (0<threshold <1).

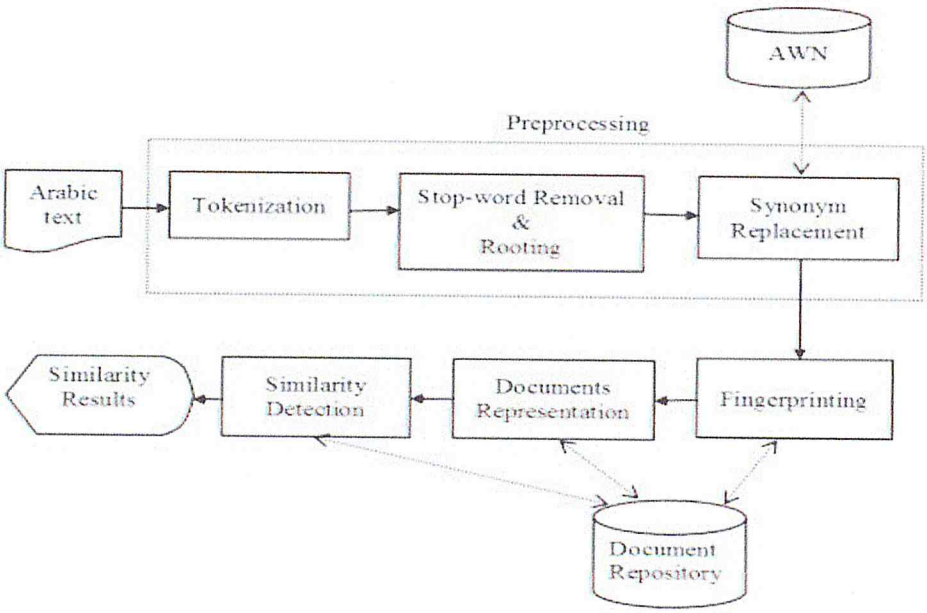


Figure 02: *L'architecture de Aplag.[28].*

3.2-méthode d'IQTEBASE0.1 plagiat in Arabic base-document (Ameera JADALLAH et Ashraf ALNAGAR)[14]

Le but de ce travail est de créer un system qui permet de détecter les phrases plagiées dans un document-text arabe à partir d'une base de données qui contient d'autres fichiers texte.

3-2-1-text processing :

C'est une phase très importante dans cette technique. Il faut faire un traitement du texte pour faciliter la détection du plagiat.

Stoplist : c'est une liste créer par (Willbur et al) qui contient 168 stop word de mot arabe et peut être éditée par l'utilisateur, elle permet d'enlever tous les mots de ce texte qui appartient a la liste de stop-word.

Stemming: c'est le processus de transformer le mot a son origine dans cette méthode ils ont appliqué KHOJA's stemmer.

Sentence segmentation:

La segmentation du texte en général est le processus consistant à diviser un texte en phrases, la longueur de phrases ne dépasse pas 35 stems.

3-2-2-Fingerprinting:

Cette étape consiste à diviser le texte en petits morceaux et ce dernier choisis le paragraphe ou la phrase ou bien le mot comme l'unité de division.

Et utilisant une fonction de hachage pour minimiser les collisions entre les différents chunks hachées on utilise la fonction de Karp-Rabin qui permet d'effectué le calcul de N-Gram.

$$N_{n\text{-}gram}(D) = L(D) - n + 1$$

L (D) = la taille de document

3-2-3-L'algorithme de winnowing:

Est une stratégie de sélection proposée par Schleimer et AL. L'idée est simple elle consiste à générer les CHUNKS des documents en utilisant la méthode de N-Gram.et puis, utiliser une fonction de hachage qui permet de produire des représentations numériques (hash values).

3-2-4-post-processing:

La dernière étape pour cette méthode est le calcule de la valeur de similarité entre les phrases, cette valeur est toujours compris entre [0,1] (0 (Pas d'originalité) et 1 (tout à fait originale)).

Le calcule se fait a partir de la mesure de similitude asymétrique Inclusion Proportion Modèle (IPM).

$$\text{Inclusion}(A, B) = \frac{|F(A) \cap F(B)|}{|F(A)|}$$

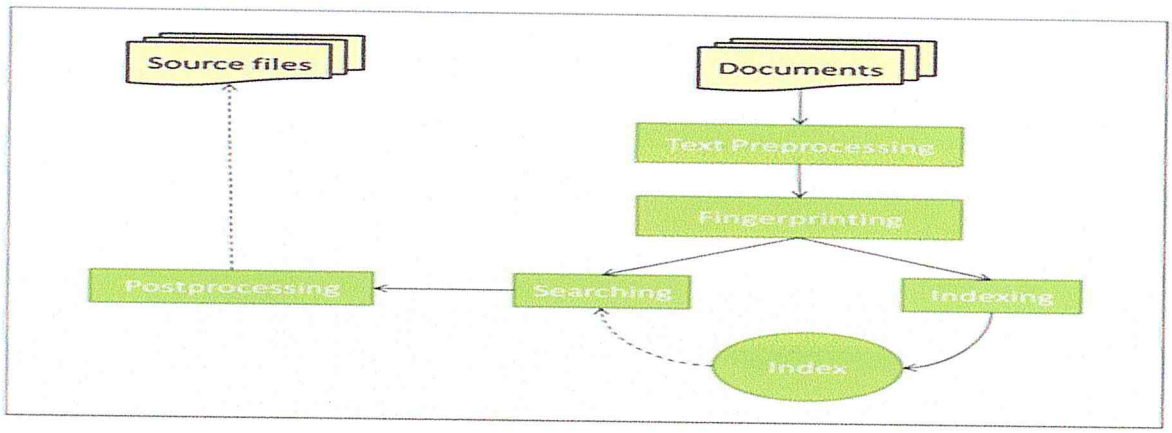


Figure 03: *L'architecture IQTEBAS01[15].*

3-3 la méthode de Fuzzy Information Retrival (Mohamed AL ZAHRANI et NAOMI Salmi) [27]

Cette méthode est créée par Salah Elzahrani et Naomie Salim, elle se base sur le comptage de pourcentage du plagiat dans un corpus.

Cette méthode compare entre deux phrases et contient les phases suivantes :

3-3-1 Pre- processing :

Stop words-removal : enlever tout les stop-words pour éviter les comparaisons qui ne sont pas effectués dans le processus de plagiat.

Stemming:

C'est la technique qui permet l'extraction des suffixes et préfixes pour rendre le mot a son origine en utilisant khoja's steemer ,

par exemple dans cette phrase :

كعبة بناء موجود ارض مكة منطقة حجاز سعودي ربط ركن اسلم حج زيارة
كعبة طاف جزء فرض حج مسلم

[28]

Est obtenu a partir de celle la:

الكعبة هي بناء موجود بأرض مكة بمنطقة الحجاز في السعودية يرتبط بأركان الاسلام وخاصة الحج ، و يعد زيارة الكعبة والطواف حولها، جزءا من فريضة الحج على كل مسلم.

[31]

3-3-2-Building corpus collection:

Le corpus de cette étude vient de ARABE WIKIPIDIA ils ont choisi 100 documents avec 4367 phrases,ils ont testé l'approche sur 5 document QueryDocs a partir des documents selectionnée CollctDocs.

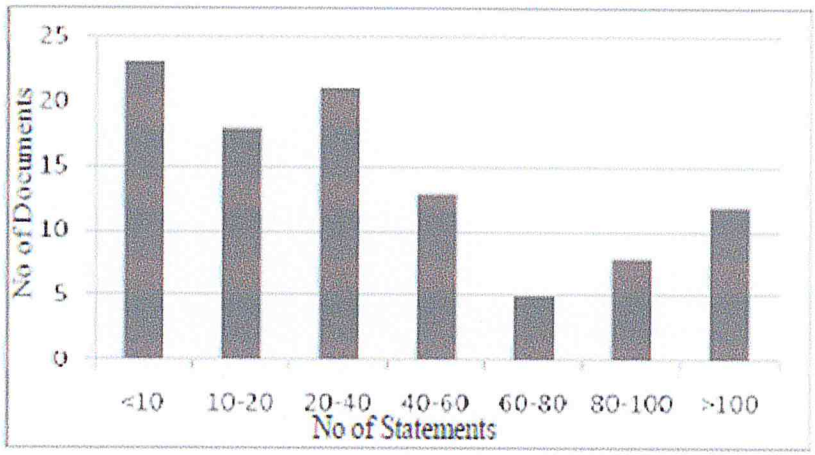


Figure 04 : la *distribution des phrases dans la corps collection.[32]*

3-3-3- l'approche Fuzzy-Set IR model:

Le calcule de similarité commence par générer les différents paire de phrases de documents CDOcs et QDOCS.

Le pourcentage du plagiat se compte par l'équation suivante:

$$Sim(Pq,Px)=(\mu 1,x+ \mu 2,x+ \ldots\ldots\ldots\ldots+\mu q,x+\ldots\ldots+ \mu n,x)/n.$$

3.4- Détection de plagiat dans les textes arabes en utilisant la corrélation du Word dans N-Gram et l'approche K- de chevauchement 2015. [20]

Dans ce système ils ont testé sur document soupçonné *Dq* et une large base de données qui contient des documents D sur le même sujet pour trouver toute les parties plagiées *dq (dq €Dq)* ce système se base sur l'algorithme de N-gram et l'approche de K-overloaping.

Les étapes de ce système sont :

1-faire une extraction des parties *dqϵDq* et *dϵD* de document suspecte.

2-trouver une liste de documents *Dx* c*D* qui sont reliée avec le document principal ou bien dans le même domaine, puis faire le calcule de similarité en utilisant les méthodes de N-gram et le fingerprinting et le coefficient de similarité de Jaccard.

3-effectuer les N-gram avec le K-overlapping segmentation.

4-effectuer les opérations de post-processing sur les phrases et les paragraphes de ce document.

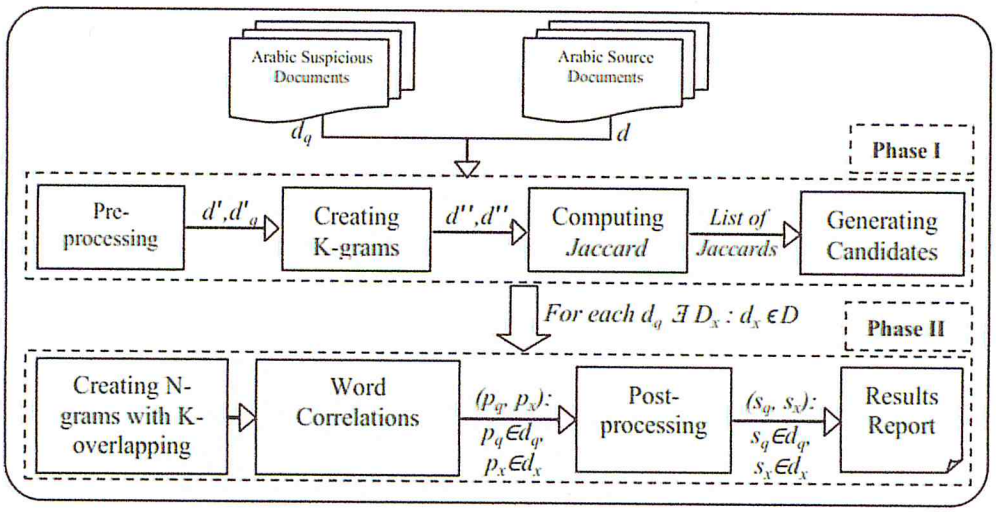


Figure 05*: l'architecture de systeme.*[33].

3.5- Plagiarism detection in arabic document [29]

La création d'un système de détection plagiat basé sur les ressources web, ce modèle dépend de Wordsteeming, fingerprinting et les algorithmes heuristiques et il a était testé sur plusieurs documents arabe. En gros c'est une évaluation de systeme1 sur IJACT journal (Aplag) une evaluation des travaux sur le plagiat dans la langue arabe ce chercheurs a basé son travail sur toutes les méthodes déjà mentionné au dessus.

4-Comparaison entre les différentes méthodes:

Le tableau suivant représente une comparaison entre les différentes méthodes utilisées pour la détection de plagiat pour la langue arabe.

Le nom de la méthode	**L'approche utulisée**	**Le pré-traitement**	**La source des documents**	**Le résultat**	**La précison**
Aplag	Le fingerprinting	1-Tokenization 2-Stopwords removal 3-Rooting 4-synonymes	Un document soupçonné	les phrases plagiées	90%
Arabic quoetes IQTEBASE0.1	Le fingerprinting avec winnowing algorithmes	1-Stoplist 2-steeming 3-scentence segmentation	Une base de donnés qui contient des fichiers .TXT	les phrases plagiées	67%
Fuzzy Information Retrival	Fuzzy-Set IR model	1- Stopwords removal 2- steeming 3- Building corpus collection	Un document soupçonné	Le pourcentage de plagiat dans le dcument	
Détection de plagiat dans les textes arabes en utilisant la corrélation du Word dans N-Gram et l'approche K- de chevauchement 2015	le K-overlapping avec le N-Grams algorithmes	1-Tokenization 2-Stopwords removal 3-Rooting 4-synonymes	Une base de données qui contient des documents de meme sujet que le document soupçonné	les phrases et les paragraphes plagiées	97%
Plagiarism detection in Arabic document	Fingerprinting et les k-gram	1-Tokenization 2-Stopwords removal 3-Rooting 4-synonymes	Les resources de web	les phrases et les paragraphes plagiées	

Tablau 08 :*Comparison entres les méthodes de détection de plagiat arabe.*

Conclusion :

Dans ce chapitre nous avons décrit les caractéristiques de la langue arabe ainsi que les problèmes de la recherche sémantique liés à cette langue. Nous avons terminé le chapitre par une présentation des différents travaux utilisant les systèmes de détection plagiat arabes.
Dans le chapitre suivant, nous présenterons notre conception dans le cadre de ce travail.
Notre but étant de proposer une approche plus performante pour l'analyse des documents et requêtes dans un système de détection de plagiat pour les textes arabes.

CHAPITRE III

CONCEPTION DU SYSTEME DE DETECTION DE PLAGIAT:

Introduction :

Le but final que nous nous fixons consiste à détecter automatiquement les paragraphes plagiés dans un corpus en langue arabe.

Dans ce chapitre nous présentons une méthodologie de conception qui permet de décrire un ensemble de méthode et d'outils permettant de prendre en charge la détection de plagiat pour les textes arabe, nous commençons par une architecture globale du système qui nous permet de décrire les caractéristiques et les étapes de notre système qui se base sur le traitement de texte arabe et l'application de la méthode Analyse sémantique Latente.

Nous proposons aussi d'autres algorithmes que nous fixons pour atteindre notre but final.

1-L'architecture de système :

Un système de détection de plagiat intègre un ensemble de techniques dédiées à sélectionner dans un document des informations volées, copiées, ou bien citées sans références.
Un système de détection de plagiat se compose de deux grandes fonctions principales représentées schématiquement par le schéma suivant :

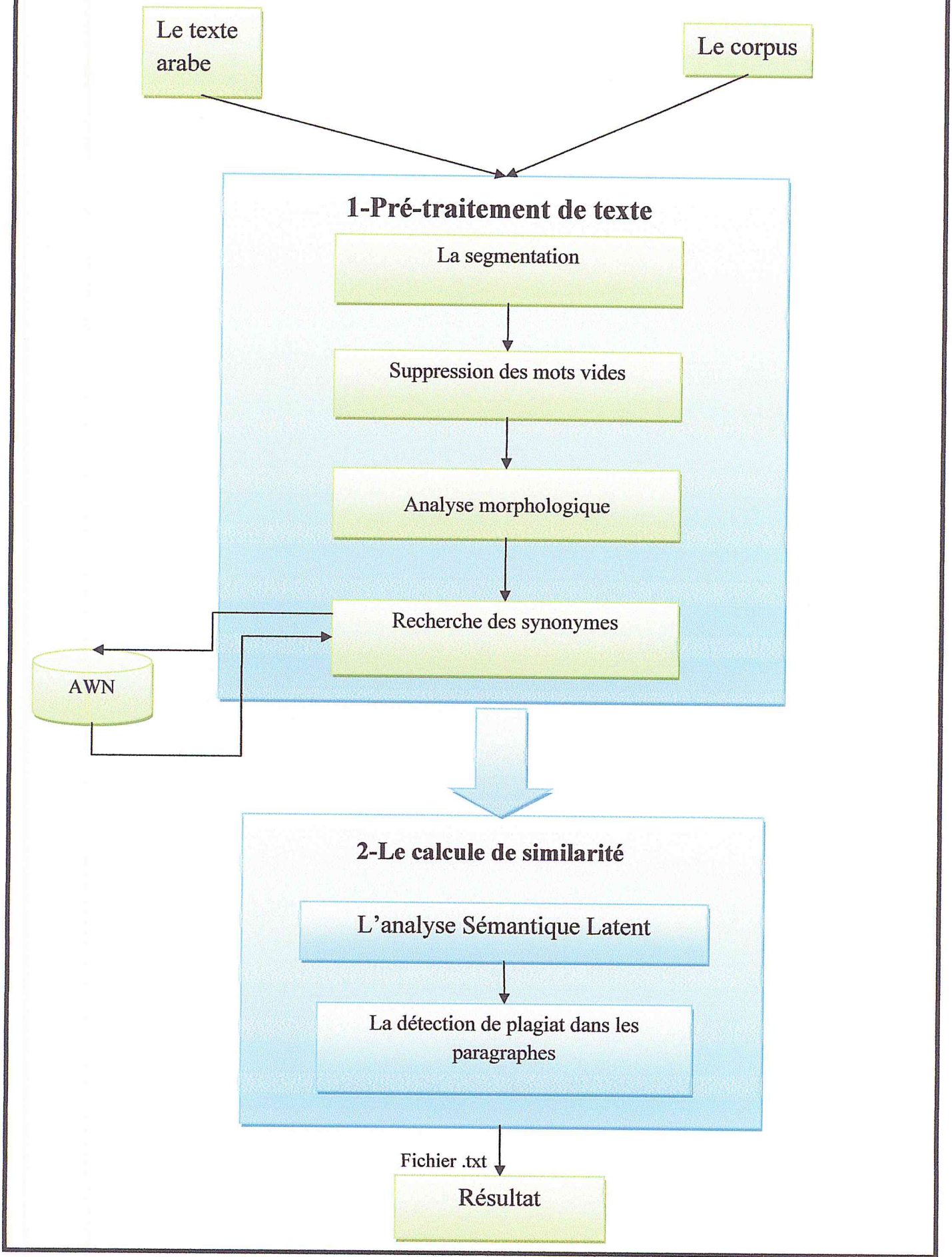


Figure 1. L'architecture globale de système.

Pour l'entrée de notre système, nous avons les informations accessibles dans le système. Elles représentent en général les collections de documents traitant d'un même domaine ou des domaines proches, et le texte choisi.

Pour le fonctionnement, nous commençons par un certain nombre de traitement sur le texte entré pour trouver les informations pertinentes de notre système. Puis nous avons notre approche et méthodologie de la détection de similarité Analyse Sémantique Latente ; En l'appliquant à l'aide d'un autre algorithme de détection des paragraphes.

Enfin, pour combiner le tout nous avons un affichage de résultat final dans un fichier texte.

1-1- le pré-traitement :

Un traitement de texte consiste à analyser chaque texte du fond documentaire afin de produire une liste de mots clés (descripteurs) qui seront utilisés ultérieurement dans le processus de détection de similarité. Dans cette partie nous allons décrire les méthodes proposés et implémentés pour cette étape.

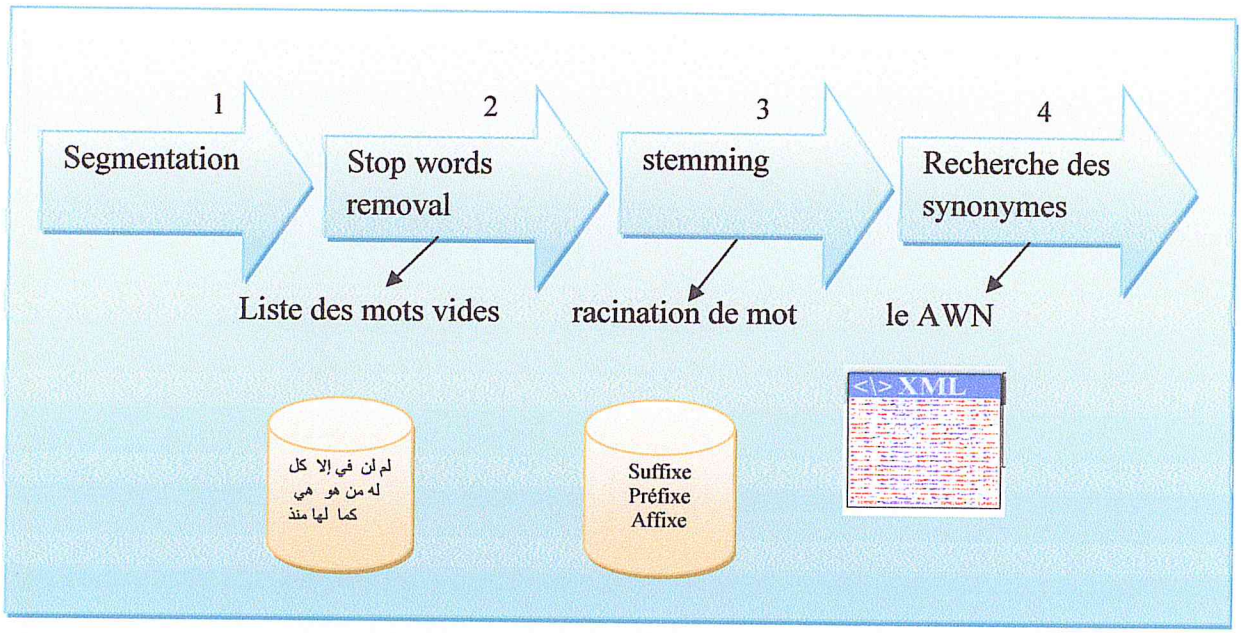


Figure02 *: les étapes de prétraitement.*

1.1.1-La segmentation :

La segmentation, sert à isoler les parties pertinentes des parties non pertinentes des textes,

Il existe plusieurs techniques de segmentation, alors nous avons étudié et implémenté l'algorithme de **« tokenization » :**

La **tokenization** désigne le processus consistant à décomposer une chaîne de caractères en une liste de composants, appelés **tokens**. Selon le contexte, un token peut être un mot ou une phrase.

Exemple :

ذَلِكَ الْكِتَابُ لا رَيْبَ فِيهِ هُدًى لِلْمُتَّقِين.

ذَلِكَ، الْكِتَابُ، لا، رَيْبَ، فِيهِ، هُدًى، لِلْمُتَّقِين.

1.1.2-La suppression de Stop-Word :

Un "mot vide" est un mot qui ne doit pas être indexé. Ceci peut économiser jusqu'à 50% de la taille d'index et par conséquent, réduire la dimension d'un modèle de représentation de documents.

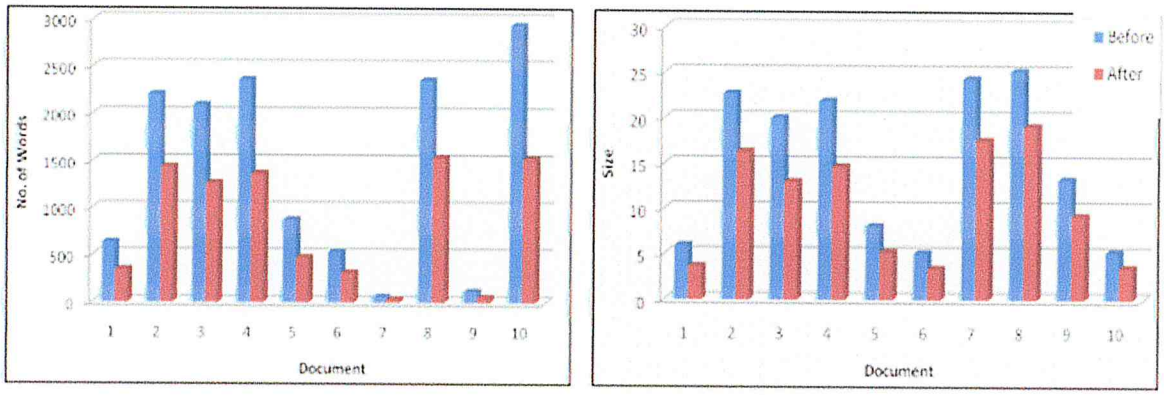


Figure 01 : *L'effet d'éliminer les Stop-Word arabe par rapport la taille des documents. [1]*

Nous avons construit un anti-dictionnaire (Stop-List) pour supprimer les mots vides de notre texte car il est évident que chaque langue dispose de son propre anti-dictionnaire. De plus, faut-il l'adapter à la nature de la collection elle-même et le domaine traité.

1.1.3-Le Stemming :

Le stemming est un processus d'enlèvement de suffixe et /ou de préfixe, qui ont été défini dans la section 1.2 dans le deuxième chapitre), Pour détecter la racine d'un mot, il faut connaître le schème par lequel il a été dérivé et supprimer les éléments flexionnels (antéfixes, préfixes, suffixes, post fixes) qui ont été ajoutés.

Le stemming détermine un schème possible. Les préfixes et suffixes sont trouvés en enlevant progressivement des préfixes et des suffixes et en essayant de faire correspondre toutes les racines produites par un schème afin de retrouver la racine [30].
Lorsqu'un mot peut être dérivé de plusieurs racines différentes, la détection de la racine est encore plus difficile, en particulier en absence de voyelles [26].

Exemple : pour le mot arabe ايمان AymAn les préfixes possibles sont : "ؤ","A ا" et "Ay اي " et les suffixes possibles sont : "ؤ" et "An ان " (Tableau), sans compter que ce mot peut aussi représenter un nom propre إيمَان *Imène*.

Stem	préfixe	Stemm	suffixe	racine	Signification
AymAn ايمان	Ø	R1yR2aR3	Ø	Amn امن	Croyance
ymAn يمان	ا A	R1R2aR3	Ø	Ymn يُمن	Convenant
mAn مان	أيAy	R1R2R3	Ø	mAn مَان	Va-t-il Approvisionner
Aym ايم	Ø	R1R2R3	An ان	Aym ايم	Deux veuves

Tableau01 : *Les stems possibles pour le mot « ايمان ».*

1.1.4-La recherche des synonymes :

Notre recherche se base sur le résultat de l'analyseur morphologique et le module de recherche des concepts à partir de Wordnet. On utilise les ressources de données dans un fichier XML de Word net Arabe qui contient les mots, les synsets et les differents formes d'un mot arabe .

Mot	Synonyme
جيد	حسن ، صالح
امتد	طال ، اتسع ، انبسط
ثبت	دائم ، باقي ، راسخ ، ثابت
نجح	توفق ، افلح ، نجاح

Tableau02 : *Les synonymes possibles pour un mot arabe.*

La recherche par le mot le plus fréquent :

A cause des synonymes des termes dans la matrice initiale il est obligé de capturer un seul terme comme entrées pour effectué la recherche de ce terme dans les autres documents pour cela nous proposons de construire un algorithme qui cherche le terme le plus fréquent dans notre collection des documents, c-à-d si un terme apparait dans les deux documents ou plus nous choisissons le concept le plus utilisé entre ces synonymes s'il existe, sinon, l'algorithme choisi le terme trouvés dans le document.74

Algorithme1 : détection dans le document

```
Def get_word_occu(ls) :
    Dic = {}
    For i in ls:
        If not dic.has_key(i):
        s = ls.count(i)
        dic [i] = s
        remove_all(i,ls)
return dic
```

Algorithme 01 *: le terme le plus fréquent.*

Le déroulement de cet algorithme est expliqué par cette figure :

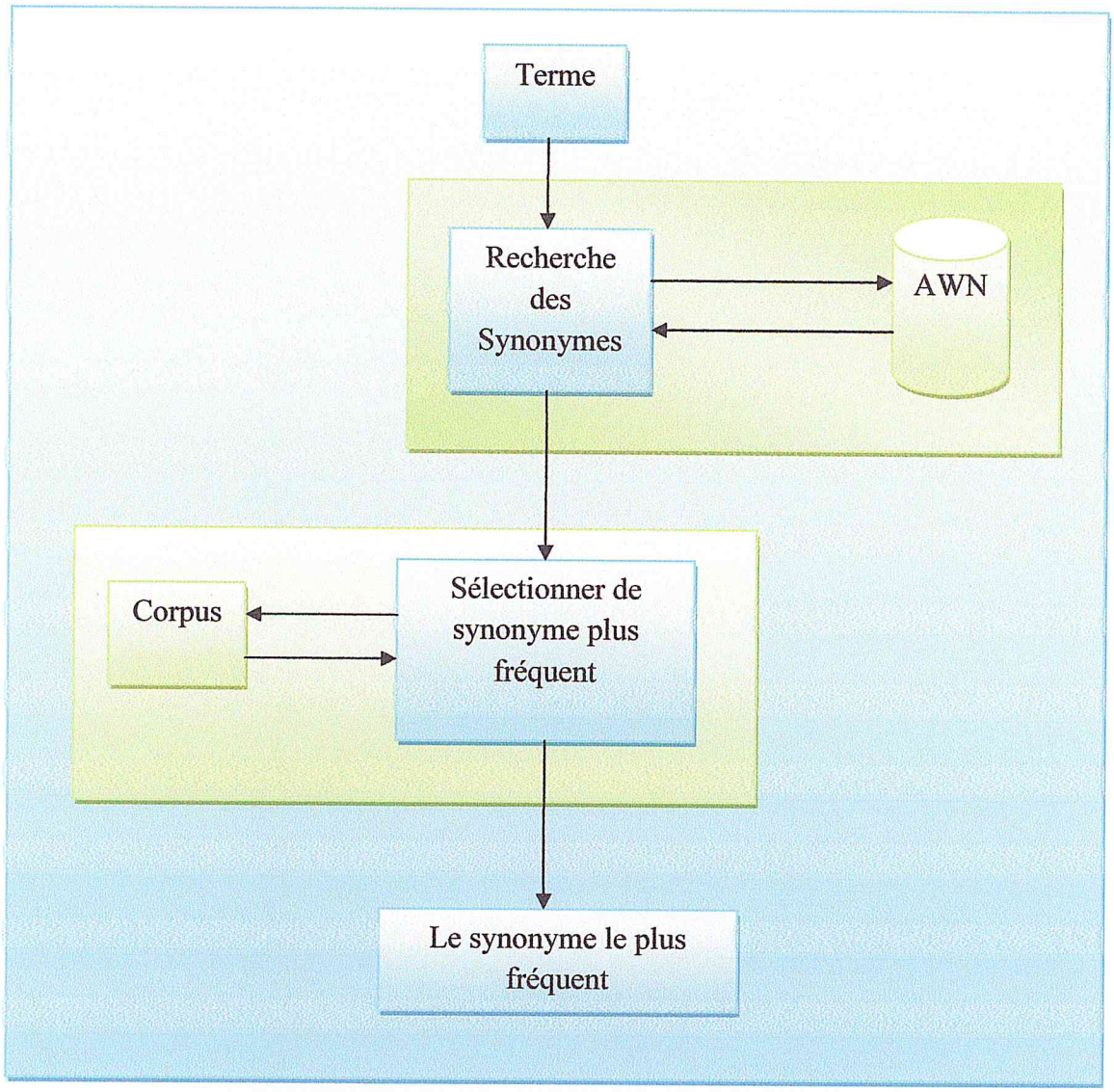


Schéma 03: *recherche du terme le plus fréquent.*

Par exemple :

"سواء كانت حالة فقدان الذاكرة بشكل مفاجئ او ببطء فذلك يعتمد على اسباب حدوث فقدان الذاكرة. ان عملية تقدم العمر قد ينتج عنها صعوبة في تعلم او ادراك الأشياء الحديثة على الشخص أو يمكن ان تتسبب في استغراق وقت اطول من قبل الشخص المسن في تذكر او استدعاء الاشياء الحديثة عليه (و لكن التقدم في العمر لا يكون سبب في فقدان الذاكرة إلا اذا كان هذا التقدم مصحوبا بمرض معين ساعد في حدوث هذه الحالة)."

Termes	synonyme sélectionnées à partir d'AWN	Entrée de synonyme (index choisi)
حدوث	{حَدَث, حُصُول, حُدُوث, ظُهُور, وُقُوع}	حُصُول
	{حُدُوث, حُصُول, حَادِثَة, حَدَث, وَاقِع}	
استدعاء	{ذِكْرَى, اِسْتِدْعاء, تَذَكُّر}	تَذَكُّر

	{إِسْتِدْعاء, حُضُور طَلَب}	
تذكر	{ذَاكِرَة, تَذَكُّر}	ذَاكِرَة
	{ذِكْرَى, إِسْتِدْعاء, تَذَكُّر}	
جاء	{جَاءَ, ظَهَرَ} {أَتَى, جَاءَ}	أَتَى
	{قَدِمَ,جَاءَ,حَضَرَ,أتى}	

Tableau 03 : *Exemple de sélection des concepts à partir d'AWN par la méthode du concept Plus fréquent.*

1.1.5-L'élimination des voyelles :

Comme nous avons cité dans le (chapitre 2.section 2.1), le problème de voyellation est une grande source d'ambiguïté pour l'analyse morphologique, syntaxique, sémantique de texte mais il faut transformer le texte du document original en un format standard plus facile et manipulable avant de faire le stemming et la recherche des synonymes.

Cette étape est nécessaire à cause des variations qui peuvent exister lors de l'écriture d'un même mot arabe.

Par exemple :

كتب → peut être lu :

كَتَبَ : Kataba. كُتُبٌ : kotobone .

كَتَّبَ : kattaba. كُتِّبَ : kottiba.

Et pour cela on été obligé de supprimer tous les voyelles des mots pour pouvoir accéder a tous les synonymes et les formes possible d'un terme. L'extraction de terme se fait à partir du texte original ce qui permet de préserver l'intégralité de l'information.

Un exemple de prétraitement du texte arabe :

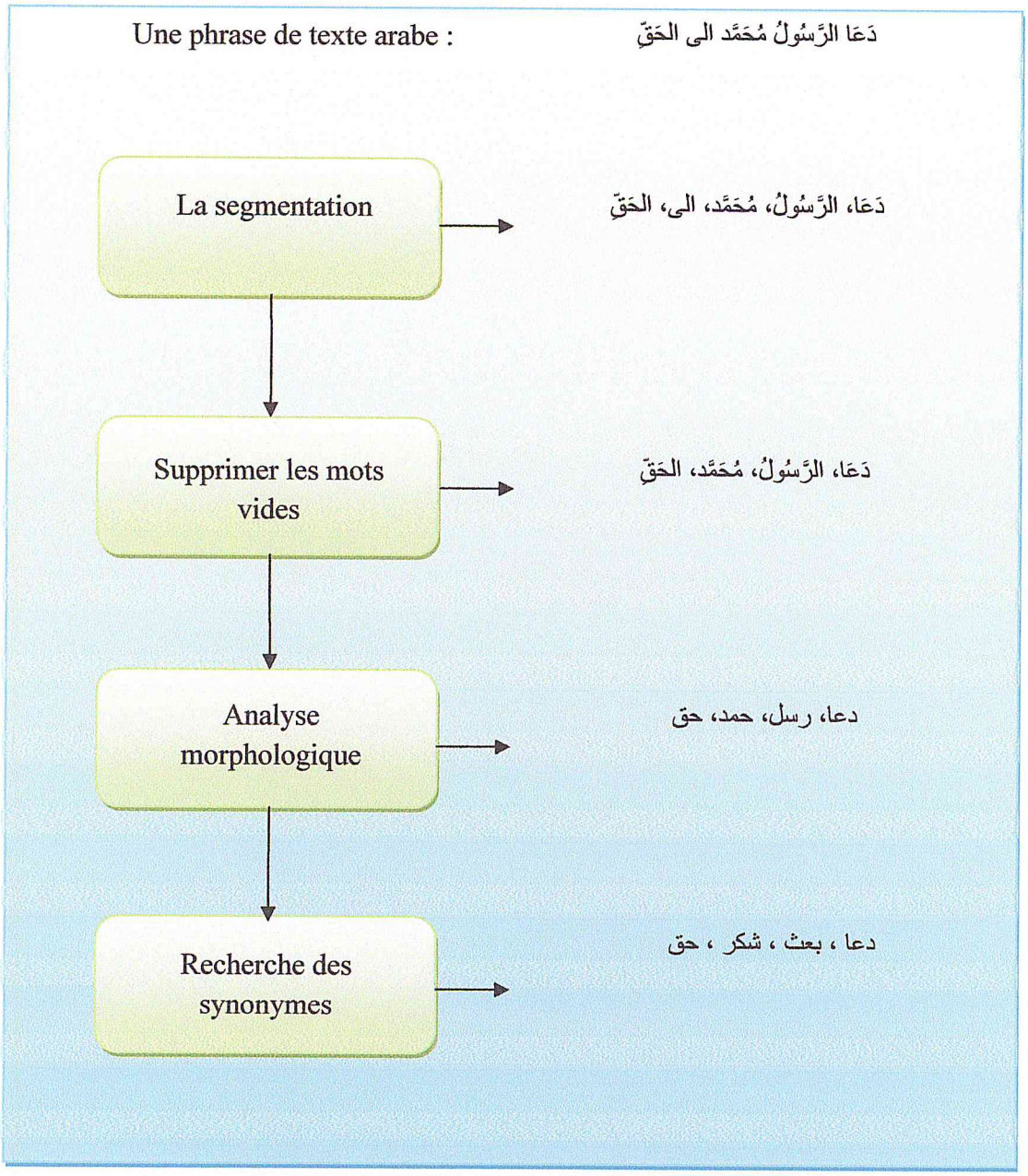


Schéma 04 : *un exemple de prétraitement.*

1-2-Le AWN (Arabic Word Net):

Pour la recherche des synonymes nous avons utilisées le Word Net Arabe qui est une base de données lexicale librement disponible pour l'arabe standard. Cette base de données suit la conception et la méthodologie du Princeton WordNet pour l'anglais et d'Euro-WordNet pour

les langues européennes. Sa structure est celle d'un thésaurus, il est organisé autour de la structure des ensembles de synonyme, c'est-à-dire des ensembles de synonymes et de pointeurs décrivant des relations vers d'autres ensembles de synonyme. [4]
Ces catégories sont au nombre de quatre : nom, verbe, adjectif et adverbe. WordNet arabe est donc un réseau lexical dont les ensembles de synonymes sont les noeuds et les relations entre ensemble de synonyme sont les arcs. Il faut noter toutefois que WordNet Arabe est une des rares ressources « libre » pour la langue générale arabe disponible en ligne. Chaque mot peut appartenir à un ou plusieurs ensembles des synonymes, et à une ou plusieurs catégories du discours. Actuellement (septembre 2015), WordNet Arabe est dans sa version « 2.0 », Il compte 11269 ensembles de synonymes(7960 noms, 2538 verbes, 661 adjectifs et 110 adverbes), et 23481 mots.
WordNet arabe est librement téléchargeable sur internet sous la forme d'une base de données relationnelle avec une interface d'accès en Java. Cette version est nommée AWNBrowser. [32]

1-3-Description de l'approche implémentée :

Notre système est basé sur l'utilisation de l'approche Analyse Sémantique Latent pour la détection de similarité dans les documents, quelques étapes ont été modifiées dans le but de permettre d'évaluer l'approche proposée.

Le Latent Sémantique Analysis :

L'Analyse Sémantique Latente est la procédure automatique proposée par Landauer et Dumais (1997) pour construire un espace vectoriel.
La méthode LSA est fondée sur le fait que des mots qui apparaissent dans un même contexte sont sémantiquement proches. Le corpus est représenté sous forme matricielle. Les lignes sont relatives aux mots et les colonnes représentent les différents contextes choisis (un document, un paragraphe, une phrase, etc.). Chaque cellule de la matrice représente le nombre d'occurrences des mots dans chacun des contextes du corpus. Deux mots proches au niveau sémantique sont représentés par des vecteurs proches. La mesure de proximité est généralement définie par le cosinus de l'angle entre les deux vecteurs. [34]

La première étape de la procédure consiste à représenter le corpus sous la forme d'une matrice de cooccurrences.

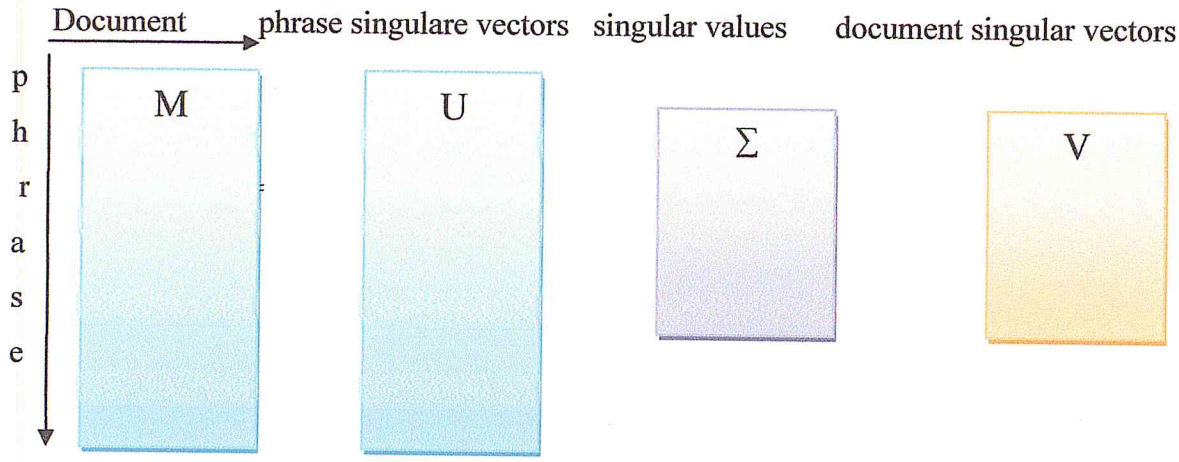


Schéma05 : *le Latent sementic analysis.*

1.3.1 -la matrice de cooccurrence

La création de la matrice relative aux mots du texte ou chaque mot est représentée par une ligne cette matrice sera créé à la base de pondération des termes.

Pondération des termes

La pondération permet d'attribuer un poids pour un terme d'indexation afin de représenter l'importance de ce terme dans le document. La plupart des techniques de pondération sont basées sur les facteurs TF et IDF :

- TF (*Term Frequency*) : mesure l'importance d'un terme dans un document. Elle est utilisée pour déterminer la pondération locale. Cette mesure est proportionnelle à la fréquence du terme dans le document. Elle peut être utilisée telle quelle ou selon plusieurs déclinaisons (log(TF), présence/absence,...) [40]

- IDF (*Inverse of Document Frequency*) : ce facteur mesure l'importance d'un terme dans toute la collection. Cette mesure est utilisée pour désigner la pondération globale.

Un terme qui apparait souvent dans la base documentaire ne doit pas avoir le même impact qu'un terme moins fréquent. Il est généralement exprimé comme suit : (Ndf), où df est le nombre de documents contenant le terme et N est le nombre total de documents de la base documentaire. [41]

La mesure TF * IDF est une bonne approximation de l'importance d'un terme dans un document. Cette mesure a eu un succès limité dans les corpus de tailles très variables.

Mais L'inconvénient de la mesure TF * IDF est quelle ne tient pas compte de la longueur de document, en effet, un terme dans un document long à plus de chances d'apparaître plusieurs fois que le même terme dans un document court. Plus le document est long, plus les termes

utilisés se répètent. La longueur des documents peut aussi induire l'utilisation d'un grand nombre de termes pour décrire un sujet. Pour pallier ce type de problèmes nous avons proposé un algorithme simple de calcul de fréquence d'apparition des mots dans le texte pour réduire la complexité de l'algorithme LSA par rapport à la taille des textes.

Algorithme1 : fréquence des termes du document

```
Def freq_word (word,doc=[{}]) :
    f=0
    for i in range (len(docs)):
    if docs[i].has_key(word):
        f + =docs[i][word]
return f


Def get_mot_pls_frequent (ls,doc s=[{}]):
    maxx= 0
    w = ls[0]
    for I in ls:
    print 'i=',i
    v = freq_word (i,docs)
        if maxx < v:
            maxx = v
            w = i
        print 'w=',w
return w
```

L'algorithme 02 : *la fréquence d'un terme.*

La deuxième étape, consiste à appliquer à cette matrice une analyse factorielle appelée décomposition en valeurs singulières pour obtenir un espace.

1.3.2- la fonction SVD :

La décomposition en valeurs singulières est une méthode générale de décomposition linéaire d'une matrice en composantes principales indépendantes. Comme une analyse en composantes principales, cette méthode permet de dégager d'un ensemble de données - ici des fréquences de cooccurrence - un nombre de facteurs sans corrélation entre eux et rendant chacun compte de la variance de l'ensemble des données.

Si n facteurs rendent compte de la totalité de la variance des fréquences de cooccurrence, alors les données peuvent être représentées dans un espace à n dimensions, chaque dimension correspondant à un facteur. [35]

Le tableau comportant les mots en lignes et les contextes en colonnes forme une matrice rectangulaire, Xm*c dans laquelle m est le nombre de lignes et c le nombre de colonnes.

Cette matrice rectangulaire Xm*c est décomposée en trois matrices dont elle est le produit, Um*n, Dn*n et Vc*n :

La matrice Dn*n est une matrice diagonale avec n colonnes et n lignes, dont les cellules de la diagonale contiennent "les valeurs singulières".

La matrice mot, U, est une juxtaposition de m lignes comportant n valeurs. Les n valeurs de chaque ligne sont les coordonnées d'un vecteur représenté dans un espace à n dimensions associé à un mot du corpus.

Chaque mot est donc représenté dans un espace à n dimensions.

À l'issue de cette étape, la similarité sémantique entre deux mots peut alors être
Calculée.

1.3.3-Le calcule de similarité (la mesure de cosinus) **:**

La similarité sémantique est estimée par le calcul du cosinus de l'angle que forment les vecteurs représentant ces mots dans l'espace à n dimensions.

Soient le vecteur u et le vecteur v, le cosinus de l'angle 0 par u et v est :

$$\cos\theta = \frac{u \cdot v}{\|u\| \times \|v\|}$$

Deux vecteurs identiques forment un angle nul dont le cosinus est égal à 1.

Deux vecteurs perpendiculaires forment un angle droit dont le cosinus est égal à 0 et deux vecteurs opposés forment un angle plat dont le cosinus est égal a –1. La similarité sémantique varie donc de –1 à 1.

La dernière étape consiste à éliminer, parmi les dimensions de l'espace résultant de la décomposition en valeurs singulières, un certain nombre de dimensions, considérées comme non pertinentes.

1.3.4- réduction du nombre de dimensions :

Toutes les dimensions dégagées de la décomposition en valeurs singulières ne sont pas pertinentes. Les dimensions associées aux valeurs singulières les plus faibles n'expliquent qu'une très faible part de la variance des données d'origine. Si ces dimensions n'étaient pas éliminées, le modèle ferait des erreurs d'estimation de la similarité sémantique. Comme les dimensions sont abstraites, il n'existe pas de critères d'élimination des dimensions non pertinentes. En conséquence, le nombre de dimensions éliminées doit être déterminé de manière empirique.

1.4-La détection de similarité dans les paragraphes :

L'algorithme de détection de similarité dans les paragraphes consiste à parcourir une liste de tous les paragraphes appartenant à chaque document de la collection des documents.

Cet algorithme exploite la notion de similarité entre les paragraphes dans un texte arabe. Le résultat de cet algorithme sont les paragraphes copiés a partir de collection de documents que nous avons choisi. Cet algorithme a été intégré dans notre processus de détection de plagiat afin de La figure ci-dessous :

```
Algorithme1 : Similarité.
Entrées : Matrice LSA, phrase, Doc j ;
Sorties :
Début
    S←0 ;
    Pour chaque mot ∈ phrase faire
        S← S+ Matrice LSA [Mot i, Doc j];
    Finpour ;
    Retourner (S) ;
Fin.
```

Algorithme 03 : *la similarité de LSA.*

```
Algorithme2 : détection de similarité dans les paragraphes.
Entrées : Matrice LSA, List {paragraphes}
Sorties :
Debut
    Pour chaque ph i ∈ List faire
        Pour chaque doc i ∈ corpus faire
            Si similarité (ph i, doc i, Matrice LSA) > 0.5 alors
                Écrire ('ph est plagia dans ', doc i);
            Finsi ;
        Finpour ;
    Finpour ;
Fin.
```

Algorithme 04 : *la similarité de LSA dans les paragraphes.*

Conclusion :

Nous avons exposé dans ce chapitre notre travail d'analyse et de conception de notre application, pour ce faire nous avons exploité la méthode LSA et d'autres outils et fonctions de traitement de texte.

Le prochain chapitre est réservé à l'implémentation de notre application.

CHAPITRE VII

IMPLEMENTATION ET REALISATION :

Introduction :

Dans ce chapitre, nous décrivons l'ensemble des outils permettant de prendre en charge notre système de détection de plagiat dans les textes arabes, nous proposons aussi une interface utilisateur et ensuite une évaluation de l'approche proposées pour une meilleure détection dans le système de détection plagiat.

1-Les langages de programmation

Pour réaliser notre application nous avons choisi deux langages de programmation le python et le java qui sont des environnements de développement intégré (EDI) :

1.1-Définition de Python

Python est un langage de script de haut niveau, structuré et open source. Il est multi-paradigme et multi-usage.

Développé à l'origine par Guido Van Rossum en 1993, il est, comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde.

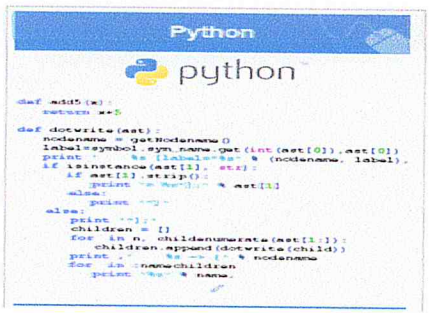


Figure 01 : *le python.*

Conçu pour être orienté objet, il n'en dispose pas moins d'outils permettant de se livrer à la programmation fonctionnelle ou impérative; c'est d'ailleurs une des raisons qui lui vaut son appellation de « langage agile ».

Parmi les autres raisons, citons la rapidité de développement (qualité propre aux langages interprétés), la grande quantité de modules fournis dans la distribution de base ainsi que le nombre d'interfaces disponibles avec des bibliothèques écrites en C,C++ ou Fortran. Il est également apprécié pour la clarté de sa syntaxe. [36]

Python est remarquable pour le nombre de bibliothèques accessibles via l'installation des modules appropriés. Que ce soit la connection avec une base de donnée, l'utilisation de

bibliothèques d'interface graphique (wxPython, PyQt, pyGTK), la manipulation avancée de XML (pyXML), le traitement d'image (Python Imaging Library), le développement de jeu vidéo (pygame), OpenGL, la grande majorité des technologies actuelles dispose de son extension python.

1.2-Les package utilisées

Numpy : NumPy (prononcé "Numb Pie" ou parfois "Numb pee ") la bibliothèque du calcul vectoriel est une extension du langage de programmation Python , l' ajout du support pour les grandes multi-dimensionnelles, des tableaux et des matrices , avec une grande bibliothèque de haut niveaumathématiques des fonctions de fonctionner sur ces tableaux. L'ancêtre de NumPy, numérique, a été créé à l' origine par Jim Hugunin avec des contributions de plusieurs autres développeurs. En 2005, Travis Oliphant créé NumPy en incorporant descaractéristiques de la numarray concurrence en numérique, avec de nombreuses modifications. NumPy est open source et a denombreux contributeurs. [37]

Scipy : SciPy (prononcé «Sigh Pie») la bibliothèque des algorithmes est une open source Python bibliothèque utilisée par les scientifiques, les analystes et les ingénieurs qui font le calcul scientifique et l' informatique technique.

SciPy contient des modules pour l' optimisation , algèbre linéaire , l' intégration , l' interpolation , des fonctions spéciales , FFT, le signal et le traitement d'image , ODE solveurs et d' autres tâches courantes en sciences et en génie.

SciPy construit sur le NumPy objet de tableau et fait partie de la pile NumPy qui comprend des outils comme Matplotlib ,pandas et sympy . Il y a un ensemble croissant de bibliothèques de calcul scientifique qui sont ajoutés à la pile des NumPy chaque jour. Cette pile NumPy a des utilisateurs similaires à d'autres applications telles que MATLAB , GNU Octave et Scilab. La pile NumPy est aussi parfois appelée la pile SciPy. [38]

1.3-Définition Java

Le langage **Java** est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld. [39]

Java est à la fois un langage de programmation et un environnement d'exécution. Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'Unix, Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications... C'est la plate-forme qui garantit la portabilité des applications développées en Java. [40]

1.4-NetBeans

Est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de *greffons*. Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

2-Fonctionnement de système

Le système se compose de deux parties, la première partie est le traitement du texte arabe qui est pour but de faciliter l'utilisation de l'approche de LSA pour les textes arabes, la seconde partie sert à utiliser notre approche pour faire la détection de plagiat dans les textes arabes, et il est schématiser comme suit :

La collection des documents

Convertir en fichier .TXT

Le texte entré

Les traitements texte

La segmentation « Tokenization »

(1)

La suppression des mots vides « stop-words »

(2)

Analyseur morphologique « stemming »

(3)

Recherche par le mot fréquent

Anti-dictionnaire

Recherche des synonymes « AWN »

(4)

AWN

Suffixe Préfixe

La détection de plagiat (l'approche de détection de similarité)

Latent sementic analysis

Créer la matrice X_0

(1)

SVD

(2)

Algo : fréquence du terme

Algo : similarité des paragraphes

Calcule de similarité cosinus

(3)

Calcule de similarité dans les paragraphes

(4)

Résultats

Dans un fichier.TXT

Schéma 06 : *Fonctionnement du système*

3-Présentation de l'application

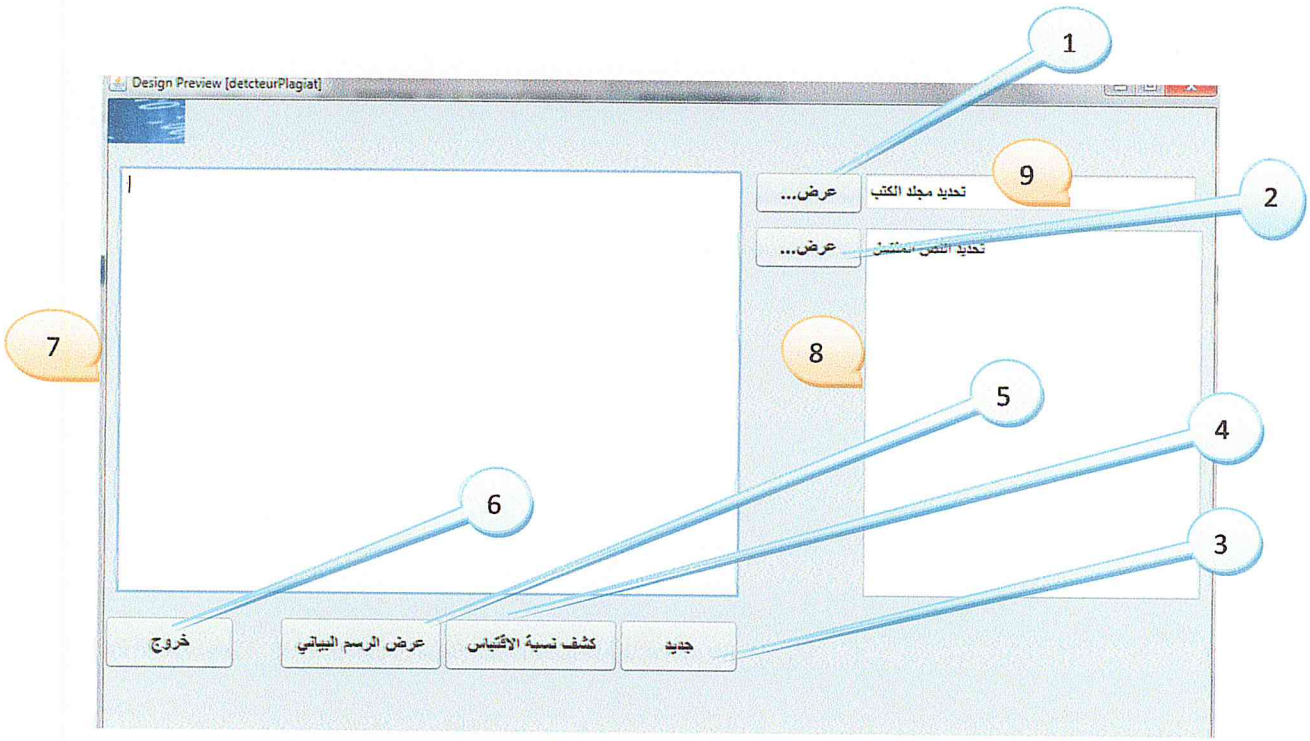
Pour faciliter la vue de notre système nous avons construit une interface utilisateur par l'IDE « netbeans » et pour notre système nous avons utilisé la version 8.0 et nous avons intégré une librairie pour relier notre code python avec notre interface java qui s'appelle « jython ».

3.1-Jython 2.7

jython est une bibliothèque implémente le langage de programmation Python sur le Plate-forme Java .elle était créée pour but de compiler un code source Python vers le codebyte Java qui peut fonctionner directement sur le bytecode Java compilées,le jython est un soutien supplémentaire pour faciliter l'utilisation des packages Java à partir de code python .

3.2-L'interface graphique :

L'interface utilisateur que nous avons développée est schématisée comme suit :



Figures 02 : *l'interface principale de notre application.*

Le menu comporte :

Le bouton (1) pour ouvrir l'emplacement.

Le bouton (2) pour afficher le texte entré.

Le bouton (3) pour ouvrir un nouveau fichier a traité.

Le bouton (4) pour afficher le pourcentage de plagiat.

Le bouton (5) pour afficher le résultat en courbe et graphe.

Le bouton (6) pour sortir de l'application.

L'espace (7) pour afficher les graphes et les courbes et le résultat finale.

L'espace (8) pour afficher le texte plagié.

L'espace (9) pour afficher l'emplacement de fichier.

Dans ce qui suit nous présentons notre expérimentation sur notre application.

4-Expérimentation et validation

4.1- corpus d'évaluation

Pour notre expérimentation nous avons utilisé une collection de 5 documents arabe, cette collection compte environ 300 000 mots différents. Cette collection est dans le domaine religieux.

Livre (1) : كتاب لا تحزن، عائض القرني

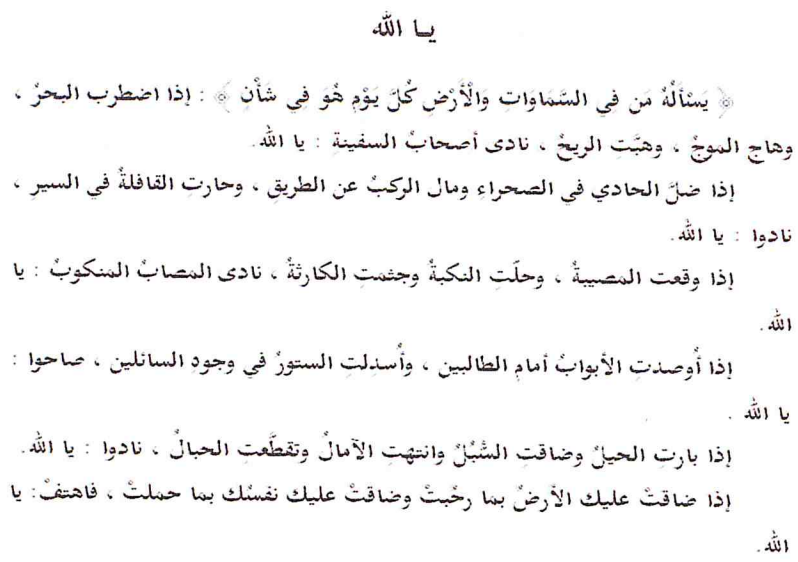
Livre (2) : كتاب جدد حياتك ، الامام محمد الغزالي

Livre (3) :كتاب دع القلق و ابدأ الحياة ، ديل كارنيدجي

Livre (4) :كتاب استمتع بحياتك ، محمد العريفي

Livre (5) :القرآن الكريم

Les textes de cette collection sont organisés sous la forme de fichier textuel écrit en arabe :



يـا الله

﴿ يَسْأَلُهُ مَن فِي السَّمَاوَاتِ وَالْأَرْضِ كُلَّ يَوْمٍ هُوَ فِي شَأْنٍ ﴾ : إذا اضطرب البحرُ ، وهاج الموجُ ، وهبّت الريحُ ، نادى أصحابُ السفينة : يا الله.

إذا ضلّ الحادي في الصحراء ومال الركبُ عن الطريق ، وحارت القافلةُ في السير ، نادوا : يا الله.

إذا وقعت المصيبةُ ، وحلّت النكبةُ وجثمت الكارثةُ ، نادى المصابُ المنكوبُ : يا الله.

إذا أُوصدت الأبوابُ أمام الطالبين ، وأُسدلت الستورُ في وجوه السائلين ، صاحوا : يا الله .

إذا بارت الحيلُ وضاقت السُّبُلُ وانتهت الآمالُ وتقطّعت الحبالُ ، نادوا : يا الله.

إذا ضاقتْ عليك الأرضُ بما رحُبتْ وضاقتْ عليك نفسُك بما حملتْ ، فاهتفْ: يا الله.

Figure 03: *Exemple d'un fichier de texte de la collection.*

Il faut noter que ces textes sont voyellés.

Nous avons utilisé un ensemble de fichier textes copiées que nous avons construit manuellement à partir de notre corpus. L'ensemble des résultats de ces testes ont été enregistré dans un fichier .Txt à part. Ce dernier va nous permettre de faire les calculs de précision et de rappel.

4.2- tests proposés

Afin d'évaluer notre système de détection de similitude entre les documents nous avons étudié séparément trois expérimentations pour améliorer le système alors ces testes ont été basées sur le remplacement des mots par leur synonymes dans les documents et par le changement du structure de la phrases afin d'atteindre le but de détection, les calcules de ces testes ont été fait manuellement.

4.2.1-Expérimentation 1 :

La première expérimentation est basé sur 2 testes essentiel nous avons remplacé les mots par leurs synonymes dans les phrases, puis changé structure générale de la phrase.

Par exemple :

ما أصابك لم يكنْ لِيُخطئِك ، وما أخطأكَ لم يكنْ لِيُصيبك

لم يخطئك ما أصابك ، و لم يكن ليصيبك ما أخطئك

4.2.2-Expérimentation 2 :

Pour mesurer l'impact de chacun des traitements nous avons appliqué chacun de ces traitements séparés :

1- appliquer seulement l'élimination des mots vides.

2- appliquer l'élimination des mots vides + l'analyse morphologique.

3- appliquer l'élimination des mots vides + l'analyse morphologique + la recherche des synonymes.

4.2.3-Expérimentation 3 :

Augmenter la taille de corpus et des textes entrée.

4.3-Évaluation et résultat

Une simple comparaison des résultats obtenus avant et après l'utilisation des traitements sur un texte sélectionné, en utilisant les métriques d'évaluation de PRECISION [35]:

$$\text{PRECISION} = \frac{\text{Nombre de phrases plagiées}}{\text{Nombre totale de phrases}} * 100$$

Nous permet de déduire que cette méthode améliore dans la plupart des cas, le nombre de documents traités et le résultat retournés. Avant de discuter les résultats obtenus, nous soulignons le fait que notre système ne consomme pas beaucoup du temps machine comparé aux autres systèmes dans le domaine de recherche de traitement automatique de langue.

4.3.1-Résultat de l'expérimentation 1

Le premier teste était le changement de structure de la phrases : ce changement n'affecte pas la détection de plagiat car l'analyseur morphologique rendre automatiquement les racines des mots et puis les tokens se considère comme des unités comparable alors le calcule sera même.
Le deuxième teste était le remplacement des mots d'origine par leur synonymes : puisque les synonymes de chaque mot sont dans la BD de AWN le remplacement n'affecte pas les calcules de détection de plagiat alors le résultat est toujours même.

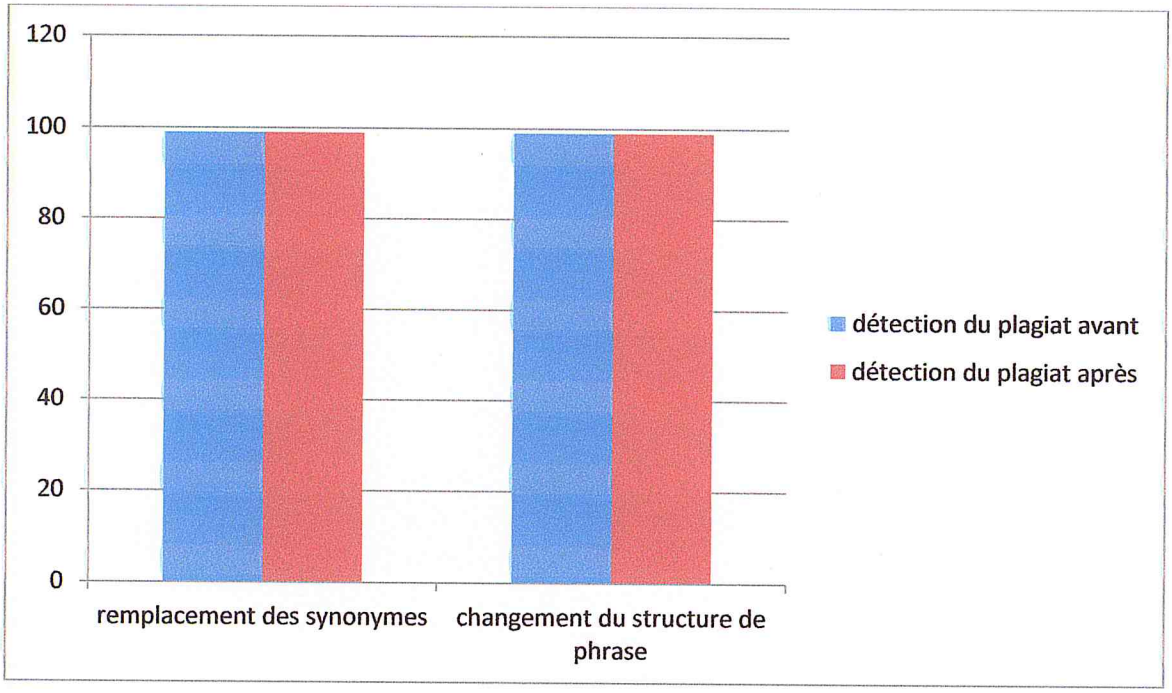


Schéma 07 : *le pourcentage de plagiat contre les tests proposés.*

4.3.2- Résultat de l'expérimentation 2

1-l'élimination des mots vides seulement : ne détecte pas les bonnes parties de plagiat par contre elle augmente la quantité de données se qui donne des phrases plagiées pleines des mots vides seulement.

2- l'élimination des mots vides seulement + l'analyseur morphologique : elle fait la bonne détection au début aussi lors du changement de la structure générale de la phrases mais lors de remplacement des mots par leur synonymes elle ne peut pas trouver le bon résultat.

3- l'élimination des mots vides seulement + l'analyseur morphologique + la recherche des synonymes : automatiquement l'application de tout les traitements est la meilleure façon de trouver le bon résultat au même temp de détecter plus d'informations plagiées.

Nous avons calculé la précision dans ces cas, et la précision moyenne.

Les résultats obtenus sont comme suit (voir tableau)

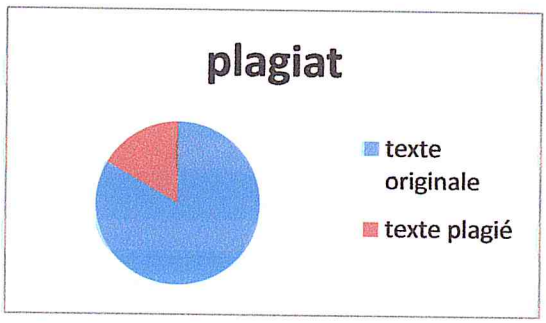
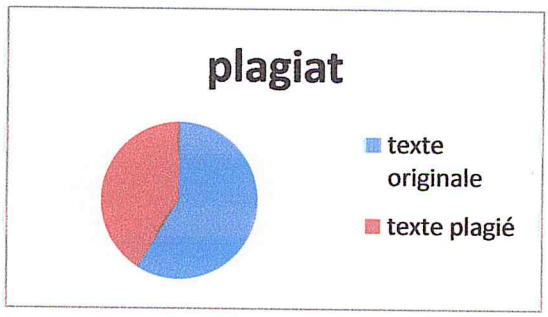
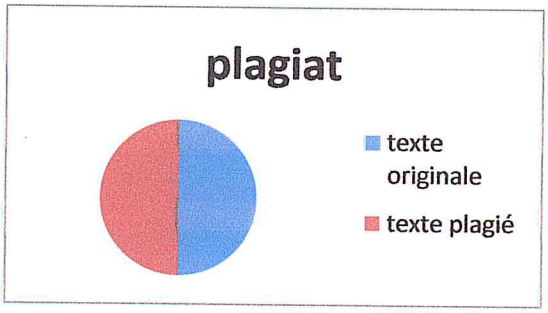
Résultats1 :
Teste1 : $PRECISION = \frac{\text{Nombre de phrases plagiées}}{\text{Nombre totale de phrases}} * 100$ 10/20 * 100 = 50% Valeur de plagiat = 10 %   
Teste2 : $PRECISION = \frac{\text{Nombre de phrases plagiées}}{\text{Nombre totale de phrases}} * 100$ 10/20 * 100 = 50% Valeur de plagiat = 30%   
Teste3 : $PRECISION = \frac{\text{Nombre de phrases plagiées}}{\text{Nombre totale de phrases}} * 100$ 10/20 * 100 = 50% Valeur de plagiat = 50%   

Tableau 01 : *l'évaluation des trois tests.*

Comparaison entre les 3testes :

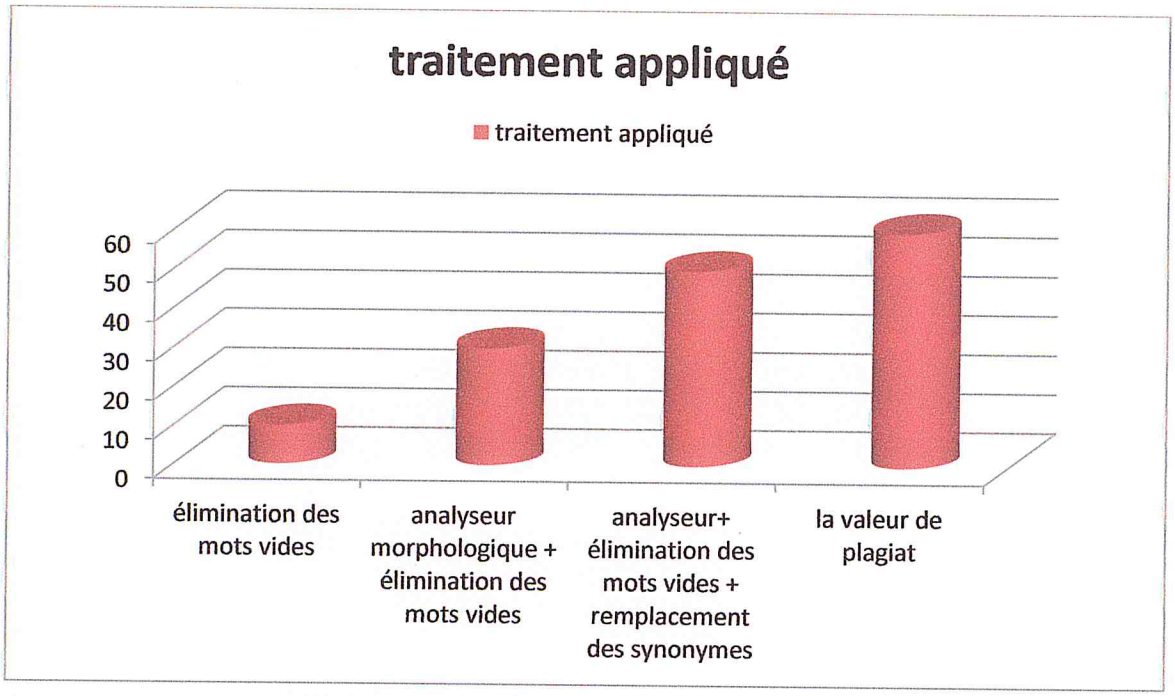
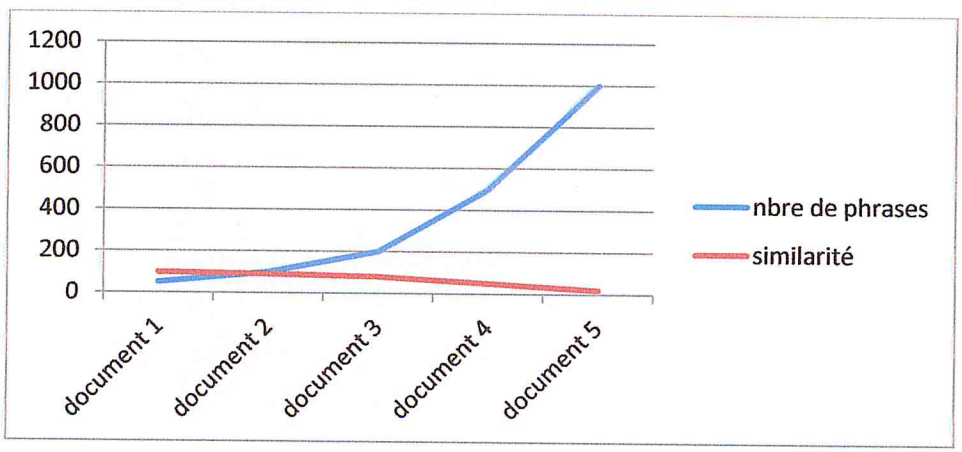


Schéma 07 : *la valeur de plagiat par rapport à nos testes.*

On remarque que a partir des testes proposées le prétraitement est une étape très nécessaire pour atteindre le but de détection de plagiat dans le texte.

4.3.3- Résultat de l'expérimentation 3 :

Lors l'augmentation de la taille pour la méthode de LSA la détection de similarité devient plus difficile a compter et elle se diminue. Le graphe au dessous montre la diminution de pourcentage de similarité contre le nombre de phrases par documents:



Graphe 08 : *Représentation de la similarité par rapport à la taille du document*

5-Discussion :

L'expérimentation réalisée avait pour but d'évaluer l'approche de l'analyse sémantique latente pour la détection du plagiat dans les textes arabes. Nous avons commencé par sélectionner la collection des documents, qui été considéré comme une étape de préparation pour les propositions, en utilisant une ressource sémantique (WordNet arabe dans notre cas). Puis, nous avons testé des différentes stratégies qui reposent sur la modification des documents et des phrases inclus. La comparaison, en termes de Précision, des différents tests faits reposant sur plusieurs éliminations des étapes de traitements nous a permis de conclure que notre système (par paragraphes) est bien meilleur qu'un système par mot. Les résultats obtenues nous ont permet aussi de déduire que nos algorithmes implémentés sont efficace pour faire le calcule de similarité. Une conséquence directe est que nous pouvons dire que l'approche de LSA peut améliorer les performances d'un système de détection plagiat pour les textes arabes.

Conclusion

Dans ce chapitre nous avons évalué l'approche de LSA pour les textes arabes. Pour ce faire nous avons construit une interface simple qui nous a aidé d'afficher nos résultats.
Nos expérimentations effectuées sur un corpus arabe de taille moyenne nous ont montré que les ressources sémantique (pour notre cas : WordNet arabe), les traitements appliqués sur les textes, les algorithmes implémentés de recherche des phrases améliorent considérablement la qualité d'un système de détection de plagiat arabe. En faisant abstraction des temps machine importants dédiés aux opérations de recherche dans les collections de documents, l'apport de ces algorithmes et de LSA aux systèmes de recherche de similitude en arabe est sans doute très intéressant et certes, mais comme il exige des documents de large taille, il est encore tôt de notre point de vue, d'estimer avec certitude les résultats finales de ce système.

Conclusion générale

Conclusion

La détection de similarité dans les documents a pour objectif d'éviter tout les types de plagiat, d'ignorance, de copier/coller, de tricher…etc, dans l'environnement d'information, et afin d'atteindre cet objectif, un système de détection plagiat doit représenter, stocker, traiter, organiser les termes de texte puis fournir à l'utilisateur les éléments correspondant au besoin de détection exprimé par sa demande. Notre mémoire s'inscrit dans le cadre de la détection de plagiat pour les textes en langue arabe.

Ainsi un système détection de plagiat pour les textes en langue arabe doit prendre en considération ses caractéristiques singulières et proposer des outils et des techniques automatiques afin de permettre son traitement informatique. L'objectif de notre travail a été d'une part, choisir les meilleurs traitements pour un texte en langue arabe, afin de passer a l'étape de détection de similarité par l'approche de LSA.

Les travaux présentés dans ce mémoire se situent dans le contexte général de l'utilisation de l'approche de l'analyse latente sémantique et d'autres approches pour la détection de plagiat dans les documents. Le but est alors d'exploiter les méthodes présentées, tout d'abord, pour une meilleure représentation de résultats, puis, pour améliorer la correspondance entre le besoin de l'utilisateur et l'information. Dans le cadre des travaux présentés dans ce mémoire nous avons exploité Wordnet Arabe, un dictionnaire pour la langue arabe librement téléchargeable sur le web. Nous nous sommes intéressés dans ce travail à proposer des solutions permettant de répondre à la question est-ce qu'il existe un moyen (méthode) pour améliorer les performances d'un système de détection du plagiat pour un texte arabe?

Dans ce cadre, nous avons présenté principalement une application traduisant le point de vue de l'utilisation de différents traitements et méthodes pour atteindre un meilleur résultat en utilisant LSA. Cette approche vise à représenter les termes de documents comme une matrice qui représente tout le document puis des fonctions qui calcule la similarité de ces matrices.
Nous avons évalué le système pour les textes arabe, cette évaluation se compose de trois partie, la première évaluation sert a testé l'efficacité de changement du structure de la phrase et remplacement de synonyme, la seconde partie a pour but de testé l'efficacité et de mesurer l'apport de l'utilisation de différents traitements dans un texte arabe, la dernière évaluation sert a testé des corpus et des textes arabe de large taille.

Nos expérimentations effectuées sur un corpus arabe de taille petite nous ont montré que les pré-traitements effectués améliorent considérablement la qualité de notre système pour ce texte arabe. L'apport de l'utilisation de LSA aux systèmes de détection plagiat en arabe est sans doute très intéressant, mais comme il exige une taille de matrice moyenne, il est encore tôt de notre point de vue, d'estimer avec certitude le taux de cet apport dans l'amélioration effective des résultats de la détection de similarité.

Perspectives

Notre travail ouvre plusieurs perspectives :

-Etudier l'effet des autres méthodes et approches sur les systèmes de détection plagiat en arabe.

-Evaluer un système performant sur tous les niveaux pour détection de plagiat des textes arabe de très large taille.

- Améliorer la méthode de détection des termes en se basant sur les autres relations existantes dans Wordnet Arabe.

[1] Robert, Paul. Micro Robert,France. Les Dictionnaires ROBERT,804, 1984.

[2] Lars R. Jones. Academic Integrity & Academic Dishonesty: A handbook about cheating and plagiarism. (12/03/2016).

[3] Ramesh R.Naik, Maheshkumar B.Landge, C.Namarata Mahender. "A review on Plagiarism detection tools", International journal of computer applications, Vol 125, N° 11, pp 125-367, (2015).

[4] Ranjeet.Kumar, R.C.Tripathi. "An analysis of automated techniques for textual similarity in research documents", International journal of advanced science and technology, Vol 56, (2013).

[5] B. Gipp, N. Meuschke. Citation Pattern Matching Algorithms for Citation-based Plagiarism Detection: Greedy Citation Tiling, Citation Chunking and Longest Common Citation Sequence. (15/03/2016).

[6] Tracey Bretag, Saadia Mahmud. Self-Plagiarism or Appropriate Textual Re-use?. (15/03/2016).

[7] Didier Duguest. Etude comparative des logiciels anti plagiat. (27/03/2016).

[8] Ahmed Jabr Ahmed Muftah. Document plagiarism detection algorithm using semantic networks,(Mémoire de Master), Université de technologie, Malaisie. (2009).

[9] Abbassi Meftah. Les systèmes de recherche d'information, (Mémoire de Master), Université de Ouargla. Algérie. (2010).

[10] Man Yan Miranda Chong. A study on plagiarism detection and plagiarism direction identification using natural language processing techniques, (Mémoire de doctorat), Université de Wolverhampton, Angleterre. (2013).

[11] Bouyakoub Soumia, Kaouadji Sarra. Enrichissement de la représentation conceptuelle dans la catégorisation de texte en utilisant les mesures de similarité sémantique, (Mémoire de Master), Université Abou Bakr Belgaid, Tlemcen. (2013) .

[12] Jean Martinet, Yves Chiaramella, Philippe Mulhem. Un modèle vectoriel étendu de recherche d'information adapté aux images. (01/05/2016).

[24] F.Douzidia, Résumé automatique de texte arabe, Mémoire de M.Sc en informatique Université de Montréal, Québec, 2004.

[25] Soraya Zaidi–Ayad, Une plateforme pour la construction d'ontologie en arabe : Extraction des termes et des relations à partir de textes (Application sur le Saint Coran), Thèse de doctorat, université, Badji Mokhtar, Annaba, 2013.

[26] Mohamed EL-BACHIR, Detection of plagiarism in arabic documents , thèse de doctorat, université du king saud, arabie saudite, riyadh, Septembre 2012.

[27] Ameera JADALLAH et Ashraf ALNAGAR, detection of arabic plagiarism. (22/04/2016)

[28] Mohamed AL ZAHRANI et NAOMI Salmi, plagiarism detection in arabic scripts using fuzzy information retrival, université du Taif et du Malaysia Johor, arabie Saudite et Malaysa, Novembre 2008.

[29] Salha Alzahrani, Arabic Plagiarism Detection Using Word Correlation in N-Grams with K-overlapping Approach 2015, college of comuting ana information thecnolgy,université du Taif, Arabe saudite, Septembre 2015.

[30] Yahya A.Abdlrahman PHD et Ahmed KHALID Assistant et professeur, these doctorat, université du Sudan department de computer science and thecnology et université du Najran du KSA, Khartoum Sudan et Arabe Saudite.

[31] K.Darwish,builiding a shallow Arabic Morphological analyzer in one day, réunion annuelle de l'association Computational Linguistics, Philadelfia USA, PP. 47-54.

[32] M.ATTIA, these doctorat, A large scale computational processor of the arabic morphology, université du caire, la caire Egypt, 2000.

[33] K.DRWISH, dissertation doctorale, Probalistic methods for searching OCR-Degraded Arabic text, université du Maryland, Washington USA.

[34] Abderrahim Mohamed ALAA EDDINE, these doctorat, Exploitations des ontologies dans les systemes d'informations Arabes, université de tlemcen, Tlemcen Algérie, 25/02/2016.

[13] Thabet Slimani, Boutheina Ben Yaghlane, Khaled Mellouli. Une extension de mesure de similarité entre les concepts d'une ontologie. (17/05/2016).

[14] Mathieu Roche, Jacques Chauché. LSA : Les Limites d'une Approche Statistique. . (20/06/2016) .

[15] D.DUGUEST, Etude comparative des logiciels antiplagiat, Mai 2008, pp 12-15.

[16] Tahar DILEKH. Implémentation d'un outil d'indexation et de recherche des textes en arabe. Mémoire de magister, Université Hadj Lakhdar , Algérie Batna , 2011.

[17] Soraya Zaidi–Ayad, Une plateforme pour la construction d'ontologie en arabe : Extraction des termes et des relations à partir de textes (Application sur le Saint Coran), Thèse de doctorat, université, Badji Mokhtar, Annaba, 2013.

[18] Tahar DILEKH. Implémentation d'un outil d'indexation et de recherche des textes en arabe. Mémoire de magister, Université Hadj Lakhdar , Algérie Batna , 2011.

[19] Djamel Kouloughli, Grammaire de l'arabe d'aujourd'hui et Pocket-Langues pour tous, 1994.

[20] Abderrahim Med El-Amine, Reconnaissance des unités linguistiques signifiantes, thèse de doctorat en informatique, université de tlemcen, 2008.

[21] F.Debilin, Voyellation automatique de l'Arabe, Proceeding Semitic '98 Proceedings of the Workshop on Computational Approaches to Semitic Languages, 1998.

[22] F.Douzidia, Résumé automatique de texte arabe, Mémoire de M.Sc en informatique Université de Montréal, Québec, 2004.

[23] Baloul, S. Développement d'un système automatique de synthèse de la parole à partir du texte arabe standard voyellé, Thèse de doctorat, Université du Maine, Académie de Nantes, France. 2003.

[35] Y. Kompaoré, Fusion de systèmes et analyse des caractéristiques linguistiques des requêtes : vers un processus de RI adaptatif. Thèse de doctorat, université Paul Sabatier, 2005.

[36] S. Karbasi : Pondération des termes en Recherche d'Information: Modèle de pondération basé sur le rang des termes dans les documents. Thèse de doctorat, université Paul Sabatier, 2007.

[37] C.BELLISSENS et P.THERAOUANNE et G.DENHIERE, deux modeles vectoriels de la mémoire sémentique, université de paris VIII, Paris France, Décembre 2004.

[38] G.VAROQUAUX, Python comme langage scientifique, Linux magazine, explorer les richesses du monde python, p40, 2009.

[39] Danny B. Lange et Oshima Mitsuru, Programming and Deploying Java Mobile Agents Aglets, Boston USA, 1998.

[40] S.Antipolis, Java de base 1, université de Nice, France.