République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj

Faculté des Mathématiques et de l'Informatique

Département de l'Informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme

Master en Informatique

Spécialité : Ingénierie de l'informatique décisionnelle

Thème :

Une nouvelle méthode de racinisation hybride et statistique pour la langue arabe

Représenté par :

- BOUBAAYA Hocine

Devant le jury composé de :

Président : BENDIAF Messaoud

Examinateur : SAIFI Abdelhamid

Encadrant : BELAZZOUG Mouhoub

Année Universitaire : 2024-2025

Remerci ements

A l'issue de ce travail nous remercions, en premier lieu, le bon Dieu de nous avoir donné la force et le courage de le mener à terme.

Nous tenons, également, à exprimer notre sincère reconnaissance et notre profonde gratitude à tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire, notamment nos chers enseignants ainsi que notre Encadreur **BELAZZOUG MOUHOUB** dont les conseils et orientations nous ont été précieusement utiles.

Dédicac

es

Nous dédions ce travail à nos très chers parents

A nos familles et amis

Et à toutes les personnes qui, de près ou de loin, nous ont aidés à mener à bien ce travail.

Qu’elles trouvent ici l’expression de notre profonde reconnaissance.

Résumé :

Ce mémoire traite de la racinisation des textes arabes, une étape clé dans le traitement automatique de la langue arabe. L'objectif est de proposer une méthode hybride combinant des techniques statistiques, des ressources sémantiques et l'apprentissage automatique, afin d'améliorer la précision de l'extraction des racines.

Le travail comprend une analyse des méthodes existantes, une évaluation comparative, le développement d'un modèle statistique souple basé sur des règles morphologiques, ainsi qu'une expérimentation sur un corpus arabe. Les résultats montrent une nette amélioration par rapport aux stemmers traditionnels, en termes de justesse et de couverture linguistique.

Mots-clés : TALN, Méthodes statistiques, Stemming, Morphologie, Racinisation, Langue arabe, Corpus arabes, Ressources lexicales.

تلخيص

تندر يتناول هذا البحث عملية **الرَّكْن (الرّصن)**في النصوص العربية، باعتبارها خطوة أساسية في مجال المعالجة الآلية للغة العربية. يهدف إلى اقتراح طريقة هجينة جديدة، تجمع بين التقنيات الإحصائية والموارد الدلالية ونماذج التعلم الآلي، لتحسين دقة استخراج الجذور.

يشمل العمل دراسة نقدية للمقاربات السابقة، وتقييماً مقارناً، ثم تطوير نموذج إحصائي مرن يستند إلى قواعد صرفية، مع اختبار الطريقة المقترحة على مجموعة من النصوص العربية. أظهرت النتائج تفوق النموذج من حيث الدقة وتغطية البنية اللغوية مقارنة بالمركّنات التقليدية.

الكلمات المفتاحية: المعالجة الآلية للغة الطبيعية، الأساليب الإحصائية، التجذيع، الصرف، استخراج الجذور، اللغة العربية، المدونات العربية، الموارد المعجمية.

Abstract

This research focuses on the process of **stemming** in Arabic texts, a fundamental step in Arabic Natural Language Processing (NLP). It aims to propose a novel hybrid stemming method that combines statistical techniques, semantic resources, and machine learning models to enhance the accuracy of root extraction.

The work includes a critical review of existing Arabic stemming approaches, a comparative evaluation of statistical methods, and the development of a flexible statistical model based on morphological rules. The proposed method is tested on a corpus of Arabic texts, and the results demonstrate its superiority in terms of precision and linguistic coverage compared to traditional stemmers.

Keywords:
Natural Language Processing, Statistical Methods, Stemming, Morphology, Root Extraction, Arabic Language, Arabic Corpora, Lexical Resources.

Table des matières

Résumé : أ

Liste des Figures خ

Liste des tableaux خ

Liste Des Abréviations د

Structure du mémoire iii

- Chapitre 1 : Le traitement du langage naturel iii
- Chapitre 2 : Étude de la langue arabe iii
- Chapitre 3 : La racinisation iii
- Chapitre 4 : iii

Chapitre 01 i

Traitement Du Langage Naturel i

1-Le Traitement du Langage Naturel (TLN) 1

1-Introduction 1

- Traitement en compréhension (analyse) 2
- Traitement en génération (synthèse) 2

2-Composantes clés du TLN 2

1. Prétraitement linguistique 2
2. Analyse morpho-syntaxique 2
3. Reconnaissance d'entités nommées (NER) 2
4. Traduction automatique 2
5. Analyse sémantique 2

3-Applications concrètes du TLN 3

4-Particularité du TLN pour la langue arabe 3

5-Apprentissage automatique (Machine Learning) 3

5.1 Méthodes d'apprentissage 4

- Apprentissage supervisé 4
- Apprentissage non supervisé 4
- Apprentissage semi-supervisé 4
- Apprentissage actif 4
- Apprentissage par renforcement 4
- Apprentissage par transfert 4

5.2 Modèles d'apprentissage 5

- SVM (Support Vector Machine) 5

- Arbres de décision 5
- Forêt d'arbres décisionnels 5
- KNN (K plus proches voisins) 5

6. Apprentissage profond (Deep Learning) 5

6.1 Fonctionnement du deep learning 5

7. Apprentissage automatique et traitement du langage naturel (TLN) 6

7.1 Le rôle du ML dans le TLN 6

7.2 Approches courantes dans le TLN 6

- Modèles basés sur des règles 6
- Approches statistiques 6
- Apprentissage profond 6

7.3 Applications du ML dans le TLN 7

7.4 Défis et perspectives 7

- Ambiguïté du langage 7
- Diversité linguistique 7
- Biais dans les données 7

8-Traitement Automatique De La Langue Arabe (Tala) 7

8-1 Introduction 7

8-2Le traitement automatique de la langue arabe 8

- Riche morphologie 8
- Absence fréquente de voyelles brèves (diacritiques) 8
- Variété dialectale 8
- Structures syntaxiques complexes 8

8-3 Racinisation (Stemming) 8

1- Étiquetage morphosyntaxique (POS Tagging) 9

- Farasa 9
- CAMeL Tools 9

2- Reconnaissance des entités nommées (NER) 9

- NER basé sur le Deep Learning 9

3- Applications du TALA 10

1. Recherche d'information (IR) 10
2. Traduction automatique (MT) 10
3. Analyse de sentiment 10
4. Applications d'assistants virtuels 10

9-Conclusion du Chapitre 1 11

Chapitre 02 i
Étude De La Langue Arabe i
1-Introduction : 12
1-1-L'écriture arabe 12
1-2-L'alphabet 13
1-3-Signes diacritiques 15
1-4-Points diacritiques 15
1-5-Les voyelles 15
1-6-Autres signes diacritiques 16
1-7-Ascendants et descendants 16
2-Morphologie arabe 17
2-1 Structure d'un mot 18
2-2 Les antéfixes : 19
2-3 Les préfixes : 19
2-4 Les suffixes : 20
2-5 Les post fixes : 21
3- Les catégories des mots L'arabe 22
1- Verbe 22
2- Nom 23
3- Particule 24
4- Spécificités de la langue arabe 25
4-1 Complexité morphologique 25
4- 2 Absence de voyelles courtes 25
4- 3 Variabilité dialectale 25
4-4 Ambiguïté lexicale et syntaxique 26
5-Conclusion du Chapitre 2 26
Chapitre 03 i
Racinisation (Stemming) i
1.Introduction : 27
2. Spécificités de la racinisation en arabe 28
- Morphologie non concaténative : 28
- Affixation complexe : 28
- Agglutination : 28
- Ambiguïté sans voyelles 28
- Pluriels irréguliers (pluriels cassés) : 28

3. Types de racinisation en arabe 28
A. STEMMING LÉGER (LIGHT STEMMING) 28
Contexte linguistique en arabe 29
Principe du stemming léger 29
Outils populaires de stemming léger pour l'arabe 30
B. Racinisation complète (root-based stemming) 31
Importance de la racine en arabe 31
Méthodologie 32
C. APPROCHES STATISTIQUES OU HYBRIDES(N-GRAM) 33
Les approches statistiques 34
A) Applications typiques : 34
b) Limites : 34
4-Conclusion de chapitre 3 35
Chapitre 04 27
Proposition d'un système hybride de racinisation 27
1-Introduction : 36
1. Les algorithmes légers 36
2. Les approches fondées sur les règles morphologiques 36
2. Prétraitement du texte 37
2.1 Normalisation du texte : 38
2.2. Suppression des mots vides et des mots arabisés 38
3. Développement du Stemmer Arabe (Blight) 39
3.1 Suppression des articles définis 40
4- Sources et Données Utilisées 50
4-1 Répartition des racines par longueur 51
4-2 Fusion de ressources morphologiques arabes 51
5 CORPUS UTILISER 53
5-1 Extraits de Wikipédia en arabe : 53
5-2 Extraits du corpus du 20 septembre 2019 54
6 Outils et langages utilisés : 56
7 Évaluation expérimentale et comparaison avec l'algorithme de Khoja 57
7.1. Protocole expérimental 57
7.2. Méthodologie d'évaluation 58
7-3 Comparaison des performances : Blight et Khoja stemmer 59
7-4 Résultats obtenus 59

7-5 Analyse comparative 60
8-Conclusion du Chapitre 4 62
Conclusion 40
Générale 40
Conclusion Générale 65
BIBLIOGRAPHIE: 68

Liste des Figures

Figure 1 : Caractères qui ont le même corps 13
Figure 2 Signes diacritiques dans l'écriture arabe 16
Figure 3 Ascendants et descendants dans la langue arabe 17
Figure 4 Segmentation du mot en arabe« أَسَتَذْكِرُونَهُ». 22
Figure 5 présentations graphiques des résultats de TP ,FP et FN 60
Figure 6 Comparaison des performances entre les stemmers Blight et Khoja 60

Liste des tableaux

Tableau 1 Différents formes d'un caractère arabe 14
Tableau 2 Classification des consonnes arabes 14
Tableau 3 Structure d'un mot arabe. 18
Tableau 4 listes des suffixes arabes. 20
Tableau 5 listes des post fixes arabes. 21
Tableau 6 Listes d'affixes 41
Tableau 7 Classification des poids morphologiques arabes selon le nombre de lettres 50
Tableau 8 : Distribution des racines par longueur selon les ressources 51
Tableau 9: Distribution finale des racines correctes 52
Tableau 10 : Statistiques générales du corpus Arabic Wikipedia 54
Tableau 11 : les mots les plus fréquents dans le corpus Arabic Wikipedia. 55
Tableau 12 Résultats obtenus entre Blight et Khoja stemmer 59

Liste Des Abréviations

TALA : TRAITEMENT AUTOMATIQUE DE LA LANGUE ARABE

RNN : Récurrent Neural Network

KNN : K plus proches voisins

SVM : Support Vector Machine

ML : Machine Learning

NER : Reconnaissance d'entités nommées

IA : l'intelligence artificielle

NLP : Natural Language Processing

TLN : Traitement du Langage Naturel

TALN : Le traitement automatique de la langue naturelle

POS Tagging : Part-of-Speech Tagging

IR : Recherche d'information

NMT : La traduction neuronale automatique

TLN : Traitement du Langage Naturel

UMass : University of Massachusetts

NLTK : Natural Language Toolkit

Def : affixes

S : suffixes

P : préfixes

Introduction Générale

Le Traitement Automatique de la Langue Naturelle (TALN) constitue un champ de recherche en pleine évolution, soutenu par les avancées en intelligence artificielle, apprentissage automatique et la disponibilité croissante de corpus textuels massifs. Ce domaine vise à doter les systèmes informatiques de la capacité à comprendre, analyser et générer du langage humain, ouvrant la voie à de nombreuses applications, telles que les moteurs de recherche, les assistants conversationnels, la traduction automatique ou l'analyse de sentiments.

Dans ce contexte, la langue arabe représente un défi linguistique particulier. Sa morphologie dérivationnelle non concaténative, l'absence fréquente de diacritiques, la richesse de son vocabulaire ainsi que la coexistence entre arabe standard et dialectes rendent le traitement automatique de cette langue extrêmement complexe. Parmi les étapes fondamentales du TALN figure la racinisation (ou stemming), qui consiste à réduire un mot à sa racine ou à une forme de base afin de limiter la variabilité morphologique. Cette étape est cruciale pour la recherche d'information, la classification automatique ou encore la détection sémantique.

Les premières approches arabes de racinisation, telles que les stemmers Light10, Khoja ou ISRI, reposent principalement sur des règles linguistiques déterministes (suppression d'affixes, correspondance avec des schèmes morphologiques). Bien qu'efficaces dans certains contextes, ces méthodes montrent leurs limites face à des corpus hétérogènes, peu normalisés ou volumineux, en raison d'une couverture lexicale limitée et d'un manque de flexibilité morphologique.

C’est dans cette optique que s’inscrit le présent travail, qui propose une nouvelle méthode de racinisation hybride et statistique, dénommée D1ight. Cette méthode combine à la fois :

- des règles linguistiques allégées (préfixes, suffixes, articles définis),
- une analyse morphologique fondée sur les schèmes (awzān),
- des critères statistiques basés sur la fréquence des formes lexicales dans de grands corpus,
- et l’exploitation de listes de racines de référence.

L’approche proposée vise à réduire les erreurs de racinisation, tout en conservant la capacité à extraire des racines pertinentes dans des contextes variés. Une expérimentation comparative est menée sur des corpus standards, notamment Arabic Wikipedia, en confrontation avec des méthodes reconnues (khoja, Light10, ARLS). Les performances sont évaluées selon des métriques standards telles que la précision, le rappel et le F1-score.

L'objectif final est de démontrer que cette méthode hybride offre une meilleure robustesse morphologique et une amélioration significative des performances en traitement automatique de la langue arabe, notamment dans des applications telles que l’indexation, le résumé automatique ou la recherche d’information.

Structure du mémoire

Le mémoire est structuré comme suit :

- Chapitre 1 : Le traitement du langage naturel
  Présentation générale du TALN, des étapes clés du traitement linguistique automatique, et des approches classiques et modernes utilisées.
- Chapitre 2 : Étude de la langue arabe
  Analyse des spécificités linguistiques de l'arabe : son système d'écriture, sa morphologie, ses affixes, ses racines et les défis posés au traitement automatique.
- Chapitre 3 : La racinisation
  Présentation des méthodes de stemming, comparaison entre approches règle-based et statistiques, et étude des principaux algorithmes existants.
- Chapitre 4 : Proposition d'un système hybride de racinisation
  Détail de la méthode développée, choix méthodologiques, implémentation, expérimentation, et analyse des résultats.

Chapitre 01

Traitement Du Langage Naturel

1-Le Traitement du Langage Naturel (TLN)

1-Introduction

Le langage est l'un des fondements de la communication humaine. Reproduire cette capacité de compréhension et de génération du langage dans des systèmes informatiques représente un défi majeur. Le Traitement du Langage Naturel (TLN)**,** ou Natural Language Processing (NLP) en anglais, est la branche de l'intelligence artificielle (IA) qui cherche à permettre aux ordinateurs d'interpréter, d'analyser, de générer et de manipuler le langage humain. Le TLN se trouve aujourd'hui au cœur de nombreuses applications pratiques telles que la traduction automatique, les assistants vocaux, les systèmes de dialogue, ou encore l'analyse de sentiments sur les réseaux sociaux. Définition et portée du TLN

Le TLN est défini comme :

« un ensemble de méthodes et d'algorithmes permettant à une machine d'analyser, de comprendre et de produire du langage humain, que ce soit sous forme écrite ou parlée. »

Le TLN permet de transformer des textes non structurés en données exploitables par des systèmes informatiques, grâce à des techniques issues de la linguistique computationnelle et de l'apprentissage automatique (machine learning). Il comprend deux dimensions majeures :

- **Traitement en compréhension (analyse)** : où la machine tente de comprendre un texte ou un discours (ex: classification, résumé, traduction).
- **Traitement en génération (synthèse)**: où la machine génère du texte ou de la parole à partir de données (ex. : chatbots, assistants vocaux).

2-Composantes clés du TLN

1. **Prétraitement linguistique**
   Cela inclut la tokenisation, la normalisation, la suppression des stopwords, et la racinisation (stemming) ou lemmatisation.
   Exemple pour l'arabe : la racinisation permet de ramener les mots à leur racine trilittérale (comme "كتب" pour "مكتوب" ,"كاتب", etc.).
2. **Analyse morpho-syntaxique**
   Analyse de la structure des phrases : étiquetage grammatical (Part-of-Speech tagging), détection des dépendances syntaxiques, etc.
3. **Reconnaissance d'entités nommées (NER)**
   Identifier les noms de personnes, d'organisations, de lieux, de dates dans un texte.
4. **Traduction automatique**
   Convertir un texte d'une langue à une autre grâce à des modèles statistiques ou neuronaux.
5. **Analyse sémantique**
   Vise à comprendre le sens global d'un texte, y compris les relations entre concepts et la détection d'émotions, d'opinions, etc.

3-Applications concrètes du TLN

- Moteurs de recherche intelligents
- Correcteurs orthographiques et grammaticaux
- Traduction automatique (ex : Google Translate)
- Assistants virtuels (ex : Siri, Google Assistant)
- Analyse des sentiments dans les réseaux sociaux
- Systèmes de résumé automatique de documents

4-Particularité du TLN pour la langue arabe

Le traitement automatique de la langue arabe présente des défis spécifiques, liés à :

- **La richesse morphologique** (préfixes, suffixes, flexions),
- **L'ambiguïté scripturale** (absence de voyelles courtes dans les textes non vocalisés),
- **La diversité dialectale** (MSA vs. Dialectes locaux),
- **La racinisation complexe**, nécessitant des algorithmes spécialisés (ex. : Light10, Khoja, Blight).

Ces spécificités nécessitent le développement de techniques adaptées et de ressources linguistiques spécifiques comme Arabic WordNet, Farasa, ou encore des corpus annotés pour l'arabe.

5-Apprentissage automatique (Machine Learning)

L'apprentissage automatique (ML) est une branche de l'intelligence artificielle (IA) qui permet aux machines de réaliser des tâches humaines complexes comme la reconnaissance d'images ou la prise de décisions.

Les algorithmes de ML s'entraînent sur des données existantes pour prédire des résultats sur de nouvelles données. Ce processus est expliqué par les schémas du fonctionnement du machine learning.

5.1 Méthodes d'apprentissage

Les principales méthodes incluent :

- **Apprentissage supervisé** : L'algorithme apprend sur des données étiquetées pour prédire des sorties sur des données non étiquetées.
- **Apprentissage non supervisé** : Utilise des données non étiquetées pour identifier des patterns ou des structures cachées.
- **Apprentissage semi-supervisé** : Combine des données étiquetées et non étiquetées.
- **Apprentissage actif** : L'algorithme cherche activement les données les plus informatives pour accélérer l'apprentissage.
- **Apprentissage par renforcement** : Un agent apprend à maximiser ses récompenses en interagissant avec son environnement (par exemple, dans les jeux vidéo).
- **Apprentissage par transfert** : Utilise des connaissances acquises dans une tâche pour améliorer les performances dans une tâche différente mais similaire.

5.2 Modèles d'apprentissage

Les modèles d'apprentissage courants comprennent :

- **SVM (Support Vector Machine)** : Sépare les données en classes via une frontière maximale.
- **Arbres de décision** : Représentent visuellement des décisions basées sur des observations de données.
- **Forêt d'arbres décisionnels** : Utilise plusieurs arbres pour la classification.
- **KNN (K plus proches voisins)** : Classifie un élément en fonction des voisins les plus proches.

6. Apprentissage profond (Deep Learning)

Le deep learning est une sous-branche de l'apprentissage automatique qui utilise des réseaux de neurones artificiels inspirés du cerveau humain pour apprendre des patterns dans de grandes quantités de données. Il est utilisé dans des domaines tels que la reconnaissance faciale et vocale, avec des applications notables comme IBM Watson.

6.1 Fonctionnement du deep learning

Le deep learning repose sur des réseaux de neurones organisés en couches, où chaque couche apprend des caractéristiques spécifiques. Plus le réseau est profond, plus il peut extraire de caractéristiques complexes.

7. Apprentissage automatique et traitement du langage naturel (TLN)

L'application de l'apprentissage automatique au traitement du langage naturel a permis des avancées majeures dans la compréhension, l’analyse et la génération du langage humain.

7.1 Le rôle du ML dans le TLN

L'apprentissage automatique permet de résoudre des problèmes du TLN comme la classification de texte, l'analyse de sentiment, la traduction automatique et la reconnaissance d'entités nommées (NER).

7.2 Approches courantes dans le TLN

Les approches incluent :

- **Modèles basés sur des règles** : Basés sur des règles linguistiques pour analyser les textes.
- **Approches statistiques** : Utilisent des données massives pour apprendre des modèles de langage.
- Apprentissage profond : Les réseaux de neurones profonds, comme les RNN et les Transformers, traitent efficacement les séquences de texte.

7.3 Applications du ML dans le TLN

Les applications pratiques incluent les assistants virtuels (Siri, Alexa), l’analyse des avis clients, la recherche d’information et les chatbots.

7.4 Défis et perspectives

Bien que l'IA ait réalisé des progrès, des défis subsistent, notamment :

- **Ambiguïté du langage** : Les machines ont du mal à saisir les subtilités du langage humain.
- **Diversité linguistique** : Les langues moins répandues ou les dialectes sont difficiles à traiter.
- **Biais dans les données** : Les modèles peuvent refléter des biais présents dans les données d'entraînement, entraînant des décisions discriminatoires.

8-Traitement Automatique De La Langue Arabe (Tala)

8-1 Introduction

Le traitement automatique de la langue arabe (TALA) est un domaine de recherche interdisciplinaire qui combine la linguistique, l'informatique et l'intelligence artificielle (IA). Il vise à automatiser des tâches telles que la reconnaissance des entités nommées, l'étiquetage morphosyntaxique, et la traduction automatique, qui sont particulièrement complexes en raison des spécificités de la langue arabe.

Avec l'essor de l'apprentissage automatique, en particulier de l'apprentissage profond, des progrès considérables ont été réalisés dans ces domaines. Toutefois, la complexité de la langue arabe, notamment sa morphologie agglutinante, l'ambiguïté lexicale et la diversité des dialectes, reste un défi majeur pour les chercheurs.

8-2Le traitement automatique de la langue arabe

La langue arabe présente des caractéristiques linguistiques particulières qui posent des défis spécifiques au traitement automatique :

- Riche morphologie : un même mot arabe peut présenter de nombreuses formes différentes en fonction de la conjugaison, du genre, du nombre, ou de l'état d'annexion.
- Absence fréquente de voyelles brèves (diacritiques) : ce qui peut entraîner des ambiguïtés dans l'interprétation des textes.
- Variété dialectale : outre l'arabe standard moderne, il existe de nombreux dialectes régionaux très différents les uns des autres.
- Structures syntaxiques complexes : telles que l'usage fréquent de l'ordre Verbe-Sujet-Objet.

8-3 Racinisation (Stemming)

La racinisation est un processus clé dans le TALA, consistant à réduire un mot à sa racine pour en faciliter l'analyse. Des algorithmes comme le **Khoja Stemmer** et le **Light10 Stemmer** sont couramment utilisés pour traiter les mots arabes.

1- Étiquetage morphosyntaxique (POS Tagging)

L'étiquetage morphosyntaxique consiste à assigner une catégorie grammaticale à chaque mot dans une phrase. Cette tâche est rendue plus complexe en arabe à cause de la flexibilité syntaxique. Des outils comme Farasa et CAMeL Tools ont été développés pour accomplir cette tâche.

- **Farasa** : Il est basé sur une approche hybride qui utilise des modèles statistiques et des règles linguistiques pour l'étiquetage morphosyntaxique.
- **CAMeL Tools** : Ce système utilise des modèles d'apprentissage profond pour l'analyse morphosyntaxique et offre des résultats précis pour des textes arabes divers.

2- Reconnaissance des entités nommées (NER)

La reconnaissance des entités nommées (NER) est une tâche essentielle du TALA, permettant d'identifier des entités spécifiques comme des noms de personnes, de lieux ou d'organisations. Les modèles de Deep Learning ont été utilisés pour améliorer la précision de la NER en arabe.

- NER basé sur le Deep Learning : Des architectures comme les réseaux neuronaux récurrents (RNN) et les long short-term memory networks (LSTM) ont montré des résultats prometteurs dans la NER pour l'arabe.

3- Applications du TALA

Le traitement automatique de la langue arabe a des applications dans plusieurs domaines, tels que la recherche d'information, la traduction automatique, l'analyse de sentiment, et les assistants virtuels.

1. Recherche d'information (IR) :

   Le TALA est crucial pour améliorer la recherche d'information sur

le web en arabe. Les systèmes modernes utilisent des modèles de racinisation et de désambiguïsation contextuelle pour mieux comprendre les requêtes des utilisateurs.

2. Traduction automatique (MT) :

   La traduction neuronale automatique (NMT), qui repose sur des architectures comme Transformer, est de plus en plus utilisée pour traduire l'arabe vers d'autres langues. Des modèles comme BERT et AraBERT ont montré des améliorations significatives dans ce domaine.

3. Analyse de sentiment :

   L'analyse de sentiment, utilisée pour évaluer les opinions exprimées dans des textes comme les critiques ou les discussions sur les réseaux sociaux, bénéficie de l'application des réseaux neuronaux convolutifs (CNN) et des LSTM pour traiter le texte arabe.

4. Applications d'assistants virtuels :

   Des systèmes comme Google Assistant et Siri commencent à intégrer des capacités de traitement du langage arabe, nécessitant des modèles avancés pour comprendre et répondre de manière appropriée aux requêtes en arabe.

9-Conclusion du Chapitre 1

Le Traitement du Langage Naturel (TLN) constitue une discipline essentielle au croisement de l'intelligence artificielle, de la linguistique et de l'informatique, visant à permettre aux machines de comprendre, analyser et produire du langage humain. Grâce aux avancées de l'apprentissage automatique, et plus particulièrement de l'apprentissage

profond, le TLN a connu des progrès significatifs dans divers domaines tels que la traduction automatique, les systèmes de dialogue, ou encore l'analyse de sentiment.

Cependant, lorsqu'il s'agit de la langue arabe, des défis supplémentaires se posent en raison de sa morphologie riche, de son écriture complexe, de l'absence fréquente de voyelles brèves et de la diversité de ses dialectes. Ces particularités exigent le développement de techniques spécifiques, telles que des algorithmes de racinisation adaptés (Khoja, Light10, Blight), des outils de désambiguïsation morphosyntaxique (Farasa, CAMeL Tools), et des modèles de traitement du langage basés sur le deep learning (RNN, LSTM, Transformers).

Ce chapitre a permis de poser les fondements théoriques nécessaires à la compréhension du TLN et du TALA, tout en illustrant les enjeux technologiques, linguistiques et applicatifs. Ces connaissances seront fondamentales pour aborder les chapitres suivants, notamment ceux portant sur l'amélioration des techniques de racinisation arabe et sur la conception de systèmes intelligents adaptés aux spécificités de cette langue.

Chapitre 02

Étude De La Langue Arabe

1-Introduction :

L'arabe est la langue parlée à l'origine par les Arabes. C'est une langue sémitique (comme l'akkadien et l'hébreu). Au sein de cet ensemble, elle appartient au Sous-groupe du sémitique méridional. Du fait de l'expansion territoriale au Moyen Âge et par la diffusion du Coran, cette langue s'est répandue dans toute l'Afrique du nord et en Asie mineure. Dire langue arabe, c'est donc parler d'un ensemble complexe dans lequel se déploient des variétés Ecrites et orales répondant à un spectre très diversités d'usages sociaux, des plus savants aux plus populaires. Mais au-delà de cette diversité, les sociétés arabes sont une conscience aiguë d'appartenir à une Communauté linguistique homogène.

1-1-L'écriture arabe

La langue arabe est parlée par plus de 400 millions de personnes et utilisée dans plus de 22 pays. Elle est également employée dans la plupart des écrits, à l'oral, dans les situations officielles ou formelles (discours religieux, politiques, journaux télévisés,). L'arabe littéral se distingue ainsi de l'arabe dialectal, qui est la langue vernaculaire parlée au quotidien et ce depuis l'expansion de l'islam. Cette variété de la langue recouvre plusieurs dialectes locaux pouvant varier assez fortement d'un pays à l'autre. Dans tous les pays arabes, un dialecte national composé par plusieurs dialectes locaux est parlé. Il existe 29 langues utilisant l'alphabet arabe : comme pour l'hébreu et le syriaque.

1-2-L'alphabet

L’alphabet arabe comporte 28 lettres pas de notion de majuscules ou de minuscules. L’écriture arabe est cursive que ce soit en imprimé ou en manuscrite. Elle s’écrit de droite à gauche. La forme des lettres dépend de leur position dans le mot. Certaines lettres prennent jusqu’à 4 formes différentes : par exemple le (ح ـحـ ـجـ) ou le (ه ـهـ ـهـ). La plupart des lettres, les formes début/milieu et fin/isolé sont identiques à la ligature près. La présence d’une ligature avec la lettre précédente ou avec la lettre suivante ne modifie pas la forme de la lettre de manière significative. En arabe, les ligatures se situent toujours au niveau de la ligne d’écriture, c'est-à-dire qu’il n’existe pas de lettre à liaison haute comme le ’o’ ou le ’v’ en alphabet latin. En écriture arabe, il existe toutefois des ligatures verticales. Certains caractères ont le même corps, mais la présence ou la position d’un point ou d’un groupe de points, est un trait déterminant pour distinguer ces caractères. La figure 1 montre quelques caractères qui ont le même corps qui se différencient seulement par la présence et le nombre de points au-dessus ou en dessous de leurs corps.

س ش ـث ت ب ـر ز ـص ض

Figure 1 *: Caractères qui ont le même corps*

N°	Lettre	Isolée	Forme		
			Initiale	Médiane	Finale
1	Alif	ا	ـا	ـا ـا	ـا
2	Baa	ب	بـ	ـبـ	ـب
3	Taa	ت	تـ	ـتـ	ـت
4	Thaa	ث	ثـ	ـثـ	ـث
5	Jeem	ج	جـ	ـجـ	ـج
6	Haa	ح	حـ	ـحـ	ـح
7	Khaa	خ	خـ	ـخـ	ـخ
8	Daal	د	د	ـد	ـد
9	Thaal	ذ	ذ	ـذ	ـذ
10	Raa	ر	ر	ـر	ـر
11	Zaay	ز	ز	ـز	ـز
12	Seen	س	سـ	ـسـ	ـس
13	Sheen	ش	شـ	ـشـ	ـش
14	Saad	ص	صـ	ـصـ	ـص
15	Shaad	ض	ضـ	ـضـ	ـض
16	Ttaa	ط	طـ	ـطـ	ـط
17	Dthaa	ظ	ظـ	ـظـ	ـظ
18	Ain	ع	عـ	ـعـ	ـع
19	Ghen	غ	غـ	ـغـ	ـغ
20	Faa	ف	فـ	ـفـ	ـف
21	Qaf	ق	قـ	ـقـ	ـق
22	Kaf	ك	كـ	ـكـ	ـك
23	Lam	ل	لـ	ـلـ	ـل
24	Mem	م	مـ	ـمـ	ـم
25	Noon	ن	نـ	ـنـ	ـن
26	Haa	ه	هـ	ـهـ	ـه
27	Wow	و	و	ـو	ـو
28	Yaa	ي	يـ	ـيـ	ـي

Tableau 1 Différents formes d'un caractère arabe

Ses alphabets changent de forme en fonction de leur position dans le mot (début, milieu, fin, isolé) et se compose de deux familles contenant le même nombre de consonnes:

- Familles Solaires : contient 14 consonnes.
- Familles Lunaires : contient 14 consonnes.

Familles Solaires														Familles Lunaires												
ن	ل	ظ	ط	ض	ص	ش	س	ز	ر	د	ذ	ث	ت	ي	و	م	ه	آ	ق	ف	غ	ع	خ	ح	ج	ب	ا

Tableau 2 Classification des consonnes arabes

1-3-Signes diacritiques

Le signe diacritique est une composante secondaire d'une lettre, qui vient la compléter ou en modifier le sens. Elles désigneront à la fois points, voyelles et autres signes secondaires (chadda, madda, hamza, ...). Cependant, dans certains travaux, seules les voyelles arabes sont appelées diacritiques.

1-4-Points diacritiques

Dans l'alphabet arabe, 15 lettres parmi les 28 possèdent un ou plusieurs points. Ces signes diacritiques sont situés soit au-dessus, soit en dessous de la forme à laquelle ils sont associés, mais jamais les deux à la fois. Un groupe de deux points peut ainsi s'écrire sous forme d'une seule, ou de deux composantes connexes. On remarque la très forte similarité entre deux points reliés par un trait. Un groupe de trois points peut donner lieu à une, deux ou trois composantes connexes, en fonction du style d'écriture.

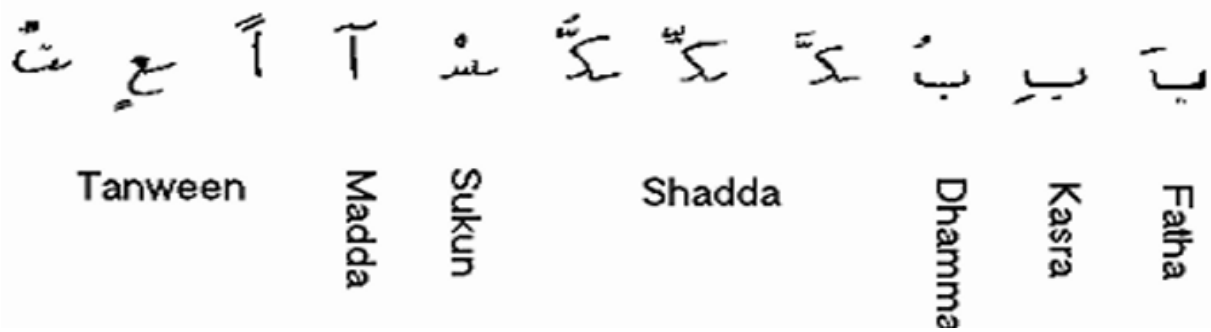
1-5-Les voyelles

En arabe, les voyelles ne sont pas des lettres, mais des signes diacritiques associés aux lettres sur lesquelles ils s'appliquent Les voyelles peuvent parfois être mentionnées sur certaines lettres pour lever l'ambiguïté et faciliter la lecture. Mais en général, les scripteurs les omettent purement et simplement, et c'est au lecteur qu'est réservé le soin d'interpréter correctement le sens de la phrase en fonction du contexte. En général on ne représente pas les voyelles, sauf dans les manuels scolaires. L'absence de voyelles peut toutefois être source de confusions. Un mot peut avoir plusieurs voyelles et par conséquent, plusieurs catégories grammaticales.

1-6-Autres signes diacritiques

Les autres signes diacritiques sont la hamza, la chadda et la madda. La chadda est une accentuation de la lettre (c'est l'équivalent d'une consonne doublée). Hamza et madda suivent des contraintes morphosyntaxiques plus complexes.

Figure 2 Signes diacritiques dans l'écriture arabe



1-7-Ascendants et descendants

Comme dans l'écriture latine, l'écriture arabe contient des ascendants et des descendants. En arabe, les descendants peuvent se prolonger horizontalement sous la bande de base, ce qui introduit une superposition verticale entre la lettre qui comprend le descendant et la lettre suivante

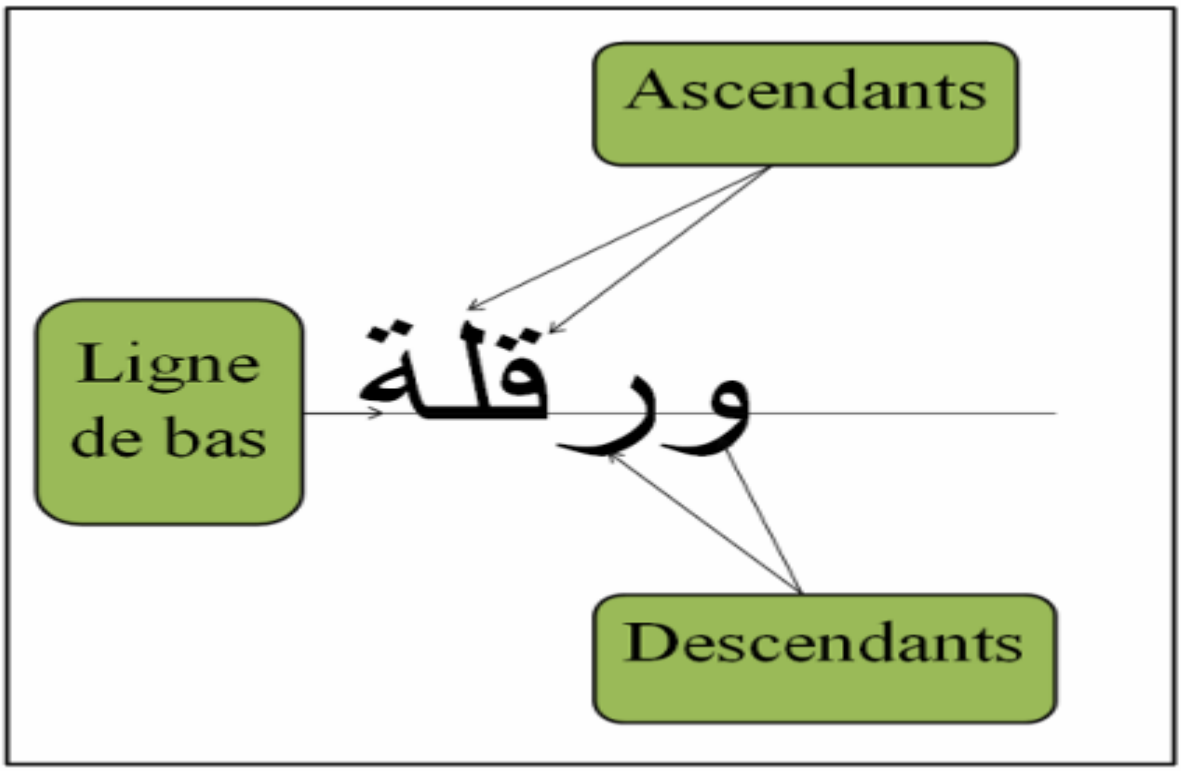


Figure 3 Ascendants et descendants dans la langue arabe

2-Morphologie arabe

Le lexique arabe comprend trois catégories de mots : verbes, noms et particules. La plupart de noms et de verbes sont dérivés d'un nombre réduit (approximativement 10000) de racines. Ces racines sont les unités linguistiques portant une signification sémantique, et la plupart d'entre elles se composent seulement de 3 consonnes, et rarement de 4 ou de 5 consonnes. De ces racines, nous pouvons produire des dérivés nominaux et verbaux par l'application des modèles (règles morphologiques). On peut produire jusqu'à 30 mots d'une racine de 3 consonnes.

L'Arabe comprend environ 150 modèles (schémas ou patrons) dont certains plus complexes, comme le redoublement d'une consonne ou l'allongement d'une voyelle de la racine, l'adjonction d'un ou de plusieurs éléments ou encore la combinaison des deux.

2-1 Structure d'un mot

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. Le problème posé par de telles techniques pour des applications de traitements de langue est malheureusement l'absence de correspondance biunivoque entre les segments ainsi identifiés et les unités textuelles élémentaires (les mots) manipulées dans le traitement linguistique. En arabe cette séquence de lettres est appelée le mot graphique.

Les mots sont séparés par des espaces et d'autres signes de ponctuation. Néanmoins, des prépositions sont agglutinées au mot (apparaissant après eux), faisant des limites invisibles entre le mot et la préposition.

Plusieurs types d'affixes sont agglutinés au début et à la fin des mots :

Antéfixe	Préfixe	Noyau	Suffixe	Post fixe

Tableau 3 Structure d'un mot arabe.

Ainsi nous pouvons les classer par catégorie selon leur rôle syntaxique.

2-2 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 «فَ » fa et «وَ » wa.

♣ L'interrogation par le morphème « أ » a.

♣ La marque du futur «سَ» sa.

♣ L'article « ال » al.

♣ Les prépositions par les lettres «بِ » bi et «لِ » li.

♣ Les particules du subjonctifs النصب «فَ » fa, «لِ » li, et «وَ » wa.

♣ Le marqueur de comparaison par les lettres «كَ » ka.

♣ Le marqueur de corroboration «لَ » la.

♣ La particule du jussif (الجزم) par la lettre « لْ» li.

2-3 Les préfixes :

Les préfixes, habituellement représentés par une seule lettre, indiquent la personne de conjugaison des verbes au présent. Ils ne se combinent pas entre eux.

2-4 Les suffixes :

N°Suff	Suffixe
1	ً
2	ٔ
3	ٌ
4	ٍ
5	َ
6	َا
7	َاتٌ
8	َاتٍ
9	َاتُ
10	َاتِ
11	انِ
12	َانٌ
13	َةٌ
14	َةٌ
15	َةٍ
16	َةَ
17	َةُ

N°Suff	Suffixe
18	َةِ
19	َتَا
20	تَانِ
21	َتَيْ
22	تَيْنِ
23	َتْ
24	َنَّ
25	َنْ
26	َوْا
27	َوْنَ
28	َيْ
29	َيْنَ
30	يْنِ
31	ُ
32	ُنَّ
33	ُنْ
34	ُو

N°Suff	Suffixe
35	ُوا
36	ُونَ
37	ِ
38	ِنَّ
39	ِنْ
40	ِي
41	ِينَ
42	ْ
43	ْتَ
44	ْتُ
45	ْتُمَا
46	ْتُمْ
47	ْتُنَّ
48	ْتِ
49	ْنَ
50	ْنَا
51	ْنَانِ

N°Suff	Suffixe
52	تَ
53	تُ
54	مَا
55	تُمْ
56	تُنَّ
57	تِ
58	نَ
59	نَا
60	نَانِ
61	يَّا
62	ِيٌّ
63	ِيٌّ
64	ِيَّ
65	ِيُّ
66	ِيٍّ

Tableau 4 listes des suffixes arabes.

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 exhaustive de tous les suffixes :

2-5 Les post fixes :

Finalement, les post fixes 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° post fixe	post fixe	Description
1	ـِي	1er Personne, Masculin/Féminin, Singulier
2	ـِيَّ	1er Personne, Masculin/Féminin, Singulier
3	نَا	1er Personne, Masculin/Féminin, Duel/Pluriel
4	كَ	2eme Personne, Masculin, Singulier
5	كِ	2eme Personne, Féminin, Singulier
6	كُمَا	2eme Personne, Masculin/Féminin, Duel
7	كُمْ	2eme Personne, Masculin, Pluriel
8	كُنَّ	2eme Personne, Féminin, Pluriel
9	هُ	3eme Personne, Masculin, Singulier
10	هَا	3eme Personne, Féminin, Singulier
11	هُمَا	3eme Personne, Masculin/Féminin, Duel
12	هُمْ	3eme Personne, Masculin, Pluriel
13	هُنَّ	3eme Personne, Féminin, Pluriel
14	ـِهِ	3eme Personne, Masculin, Singulier
15	ـِهِمَا	3eme Personne, Masculin/Féminin, Duel
16	ـِهِمْ	3eme Personne, Masculin, Pluriel
17	ـِهِنَّ	3eme Personne, Féminin, Pluriel

Tableau 5 listes des post fixes arabes.

Tous ces affixes devraient être traités correctement pendant la lemmatisation de mots.

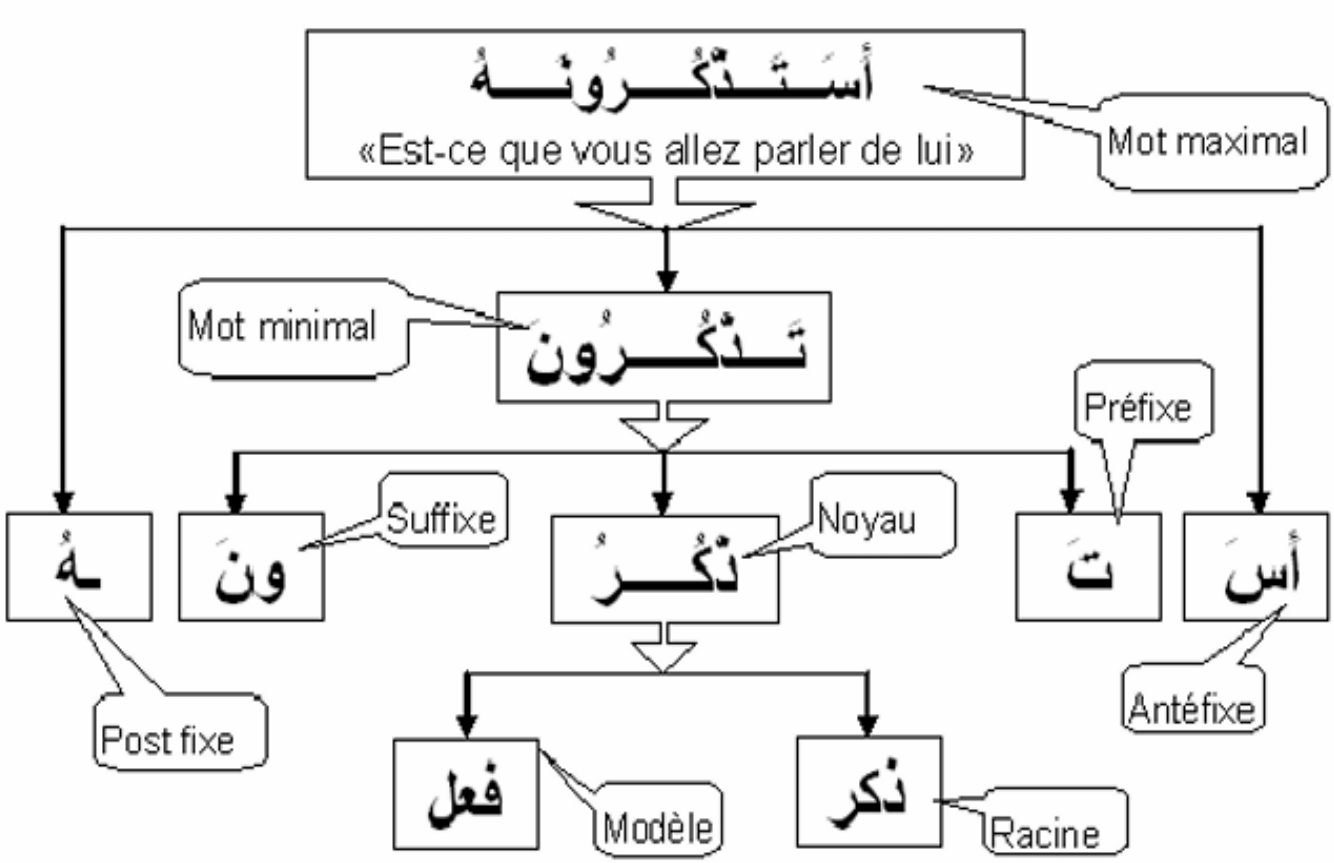


Figure 4 Segmentation du mot en arabe« أَسَتَذَكَّرُونَهُ».

3- Les catégories des mots L'arabe

Comprend trois catégories de mots : verbe, nom et particule.

1- Verbe

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, peux 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).

2- Nom

Les noms arabes regroupent les substantifs, les adjectifs et les pronoms, ainsi que d'autres noms invariables. Les substantifs et les adjectifs sont créés en prenant pour origine tantôt un type verbal, tantôt un type nominal. Nous pouvons distinguer dans deux classes de noms : la première regroupe les noms conjugables ou semi-conjugables qui peuvent avoir la forme duelle, plurielle, etc. la deuxième classe regroupe les noms non-conjugables qui gardent la même forme quel que soit le contexte. Les noms conjugables sont soit des noms primitifs, qui échappent à toute dérivation comme « غَزَالْ» (gazelle), soit des noms dérivationnels qui sont formés à partir d'une racine comme « مَكْتَبَة» (bibliothèque) de la racine «كتب».

3- Particule

Les particules sont des lemmes invariables et en nombre limité. Ils indiquent l'articulation de la phrase. Elles sont classées selon leur champ sémantique et leur fonction dans la phrase; on en distingue plusieurs types

♣ Préposition : exemple (ب ،ك ،ل ،عن ،حتى)

♣ Particules de coordination : exemple (و ،ف ،ثم ،أو)

♣ Particules interrogatives : exemple (أ ،هل ،أم)

♣ Particules d'affirmation : exemple (أجل، بلى ، نعم)

♣ Particules de négation : exemple (لا، لن ،لم)

♣ Particules distinctives : exemple (أي)

♣ Particules relatives : exemple (ما)

♣ Particules de futur : exemple (سَوف ، سَ)

♣ 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.

4- Spécificités de la langue arabe

4-1 Complexité morphologique

L’arabe est une langue agglutinante et flexionnelle, où les mots peuvent être modifiés par des préfixes, suffixes et infixes. Un mot peut avoir plusieurs formes en fonction de son contexte grammatical. Par exemple, **"كتب"** (ktb) est la racine de plusieurs formes comme **"مكتبة"** (maktaba, bibliothèque) et **"مكتبات"** (maktabāt, bibliothèques). Ces variations rendent la racinisation nécessaire pour réduire les mots à leurs racines.

4- 2 Absence de voyelles courtes

L’arabe écrit omet souvent les voyelles courtes, ce qui peut conduire à plusieurs interprétations d’un même mot. Par exemple, **"كتبت"** peut signifier "il a écrit" ou "il a été écrit" selon le contexte. Cela nécessite des modèles qui peuvent désambiguïser les mots en fonction du contexte.

4- 3 Variabilité dialectale

L’arabe présente une grande diversité dialectale, avec des différences significatives entre l’arabe standard moderne (Fusha) et les dialectes régionaux. Les systèmes TALA doivent être capables de traiter ces variations. Par exemple, l’arabe égyptien et l’arabe levantin diffèrent considérablement dans leur vocabulaire et leur syntaxe.

4-4 Ambiguïté lexicale et syntaxique

L’arabe présente également une forte ambiguïté lexicale et syntaxique. Un même mot peut avoir plusieurs significations, ce qui

nécessite des modèles contextuels capables de résoudre cette ambiguïté en fonction du texte.

5-Conclusion du Chapitre 2

Ce chapitre a mis en lumière les particularités linguistiques et morphologiques de la langue arabe, telles que sa structure dérivationnelle, l'utilisation d'affixes complexes, l'absence fréquente de voyelles brèves, ainsi que la diversité de ses dialectes. Ces spécificités rendent le traitement automatique de l'arabe particulièrement complexe.

Comprendre ces éléments est essentiel pour concevoir des outils adaptés au TALN, notamment pour la racinisation, l'analyse morphosyntaxique et la compréhension des textes. Le prochain chapitre abordera les méthodes existantes pour le traitement morphologique de l'arabe, en particulier les techniques de racinisation.

Chapitre 03

Racinisation (Stemming)

1.Introduction :

La racinisation (ou stemming) est une étape essentielle dans le traitement automatique de la langue naturelle (TALN), notamment pour les systèmes de recherche d'information, de classification de texte, ou encore pour la traduction automatique. Elle consiste à réduire un mot fléchi, dérivé ou conjugué à une forme de base, appelée radical ou racine. Dans le contexte de la langue arabe, cette tâche est particulièrement complexe en raison de sa morphologie riche et non concaténative, basée sur des racines trilitères et des modèles dérivationnels.

Contrairement aux langues indo-européennes, où les suffixes sont généralement ajoutés à une base fixe, l'arabe utilise une structure interne de racines (souvent de trois consonnes), sur laquelle sont appliqués des schèmes (modèles morphologiques) pour produire différentes formes grammaticales et lexicales. Par exemple, la racine « ك-ت-ب » peut générer des mots tels que « كتاب » (livre), « كاتب » (écrivain), ou « مكتبة » (bibliothèque), en fonction du schème utilisé.

La racinisation permet de regrouper les variantes morphologiques d'un mot sous une même forme représentative, ce qui améliore la récupération d'information, l'analyse sémantique,

Dans les sections suivantes, nous présentons les différentes méthodes de racinisation appliquées à la langue arabe, leurs principes, leurs avantages et inconvénients, ainsi que les principaux outils existants dans le domaine.

2. Spécificités de la racinisation en arabe

La langue arabe présente des défis uniques pour la racinisation :

- Morphologie non concaténative : la racine arabe est généralement trilittérale (trois lettres), insérée dans un schème (motif vocalique) pour produire des dérivés.
- Affixation complexe **:** les mots peuvent contenir plusieurs préfixes, suffixes et infixes.
- Agglutination **:** les prépositions, conjonctions et articles peuvent être attachés au mot.
- Ambiguïté sans voyelles **:** la vocalisation est souvent absente dans les textes modernes, rendant l'analyse plus difficile.
- Pluriels irréguliers (pluriels cassés) **:** ils changent la forme interne du mot de façon imprévisible (ex. : نساء → امرأة).

3. Types de racinisation en arabe

A. STEMMING LÉGER (LIGHT STEMMING)

Le stemming léger, ou racinisation légère, est une méthode de réduction morphologique visant à supprimer uniquement les affixes fréquents (préfixes et suffixes) dans les mots arabes, sans essayer d'en extraire la racine trilittérale, Contrairement aux approches basées sur la racine, cette méthode cherche à préserver le sens lexical du mot pour maintenir une pertinence sémantique plus forte dans des tâches comme la recherche d'information (RI), la classification automatique, et l'analyse de sentiments.

Contexte linguistique en arabe

La langue arabe présente une morphologie non concaténative, avec des racines trilitères et de nombreux affixes (préfixes, infixes, suffixes), souvent agglutinés. On y retrouve :

- Des préfixes fréquents : و…، ف، ال، ب، ك
- Des suffixes répandus : ون…، ين، ات، ها، هم
- Une structure morphologique dérivationnelle riche,
- Une flexion qui génère de nombreuses formes grammaticales.

Ces caractéristiques rendent les approches classiques (e.g. le suffix-stripping de Porter pour l'anglais) inefficaces pour l'arabe. Le stemming léger offre donc une alternative pragmatique, consistant à opérer une réduction minimale pour maximiser la précision tout en limitant les erreurs de sous-racinisation.

Principe du stemming léger

Le stemming léger se déroule généralement en trois étapes principales:

1. **Normalisation** :
    - Suppression des diacritiques,
    - Unification de lettres variantes (e.g. « أ » → « ا », « ي » → « ى », « ة » → « ه »).
2. **Suppression des affixes fréquents** :
    - **Préfixes** : و…، ال، لل، ف، بـ، كـ
    - **Suffixes** : ون…، ين، ات، ها، هم، ان، كما
3. **Extraction du radical résiduel** :

- Le mot résultant est un radical simplifié, conservant souvent une signification claire, mais pas nécessairement la racine trilittérale.

Exemple pratique

Mot original	Préfixe supprimé	Suffixe supprimé	Radical obtenu
والكتابات	وال ← الكتابات	ات ← الكتاب	كتاب
بالمدارس	بال ← المدارس	س ← المدر	مدرس
فالمعلمين	فال ← المعلمين	ين ← المعلم	معلم

Limites et critiques

Malgré son efficacité en prétraitement, le stemming léger présente plusieurs limitations :

- Il ne regroupe pas toutes les formes dérivées sous une racine unique.
- Il ignore les pluriels irréguliers (« pluriel cassé »), fréquents en arabe.
- Il peut laisser subsister des ambiguïtés lexicales, notamment en l'absence de contexte.
- Il est moins performant pour les tâches nécessitant une compréhension fine, comme la traduction automatique.

Outils populaires de stemming léger pour l'arabe

1. **Light10 Stemmer** – Développé par Lucene pour le projet UMass Arabic Information Retrieval, il applique une série de règles simples pour supprimer les affixes les plus courants.

2. **ISRI Stemmer** – Conçu à l'Institute for Scientific Research in Iraq, il s'appuie sur des listes d'affixes et une stratégie de suppression conservatrice.
3. **Tashaphyne** – Une bibliothèque Python open source développée par Zerrouki, adaptée aux tâches de prétraitement léger pour le TALN arabe.
4. **NLTK Arabic Stemmer** – Intégré dans la bibliothèque NLTK, il fournit une racinisation simplifiée, utile pour les tâches basiques de traitement textuel.

B. Racinisation complète (root-based stemming)

La racinisation complète, ou stemming basé sur les racines, est une méthode de traitement morphologique qui vise à extraire la racine trilittérale (ou parfois quadrilitérale) d'un mot arabe. Cette racine, généralement composée de trois lettres, constitue le noyau sémantique à partir duquel plusieurs mots dérivés sont formés.

Ce type de stemming est particulièrement utile pour l'analyse linguistique avancée, l'extraction d'information, la classification thématique, et dans certains cas, la recherche d'information sémantique.

Importance de la racine en arabe

La langue arabe repose sur un système morphologique dérivationnel très structuré. À partir d'une racine (comme ك ت ب), on peut générer des mots de sens lié :

- كَتَبَ (il a écrit)

- كِتَاب (livre)
- كَاتِب (écrivain)
- مَكْتُوب (écrit)
- مَكْتَبَة (bibliothèque)

Ainsi, extraire la racine permet de regrouper toutes ces formes, même si elles diffèrent considérablement en apparence.

Méthodologie

L'algorithme de racinisation complète repose souvent sur les étapes suivantes :

1. **Normalisation orthographique** : unification des lettres et suppression des diacritiques.
2. **Suppression des affixes** : retrait de préfixes et suffixes fréquents (ex. ات, ون, يـ, تـ, ال…).
3. **Détection du schème morphologique** : identification du modèle verbal/nominal sous-jacent.
4. **Extraction de la racine trilittérale** : en utilisant des règles linguistiques ou un dictionnaire morphologique.

- **Khoja Stemmer (1999)**
  L'un des premiers stemmers basés sur les racines, utilisant des listes d'affixes et un dictionnaire de racines arabes.
- **MADAMIRA & Buckwalter Analyzers**
  Systèmes avancés de racinisation et d'analyse morphologique utilisés dans les applications de TALN.

- **Farasa Stemmer**

  Un stemmer efficace basé sur des modèles statistiques, combinant précision et rapidité.

Avantages

- Permet de grouper efficacement des mots de même origine sémantique.
- Améliore les performances dans des applications nécessitant une analyse profonde du sens.
- Particulièrement utile pour les dictionnaires électroniques et les systèmes de traduction automatique.

Inconvénients

- Complexité linguistique élevée : nécessite la reconnaissance de nombreux schèmes morphologiques.
- Erreur fréquente de sur-racinisation (overstemming) : plusieurs mots peuvent être réduits à une même racine incorrecte.
- Perte de nuances morphologiques importantes pour certaines applications (ex. sentiment analysis).
- Moins performant que le stemming léger dans les systèmes de recherche d'information à grande échelle.

C. APPROCHES STATISTIQUES OU HYBRIDES(N-GRAM)

Face à la complexité morphologique et à l’ambiguïté sémantique de la langue arabe, les approches purement basées sur les règles ont montré leurs limites. Pour surmonter ces obstacles, les chercheurs ont exploré des approches statistiques puis hybrides, qui combinent règles

linguistiques et apprentissage automatique, Ces méthodes exploitent les grandes quantités de textes annotés pour construire des modèles probabilistes ou neuronaux capables de généraliser le comportement linguistique de l'arabe.

Les approches statistiques

Les approches statistiques utilisent des techniques issues de la modélisation probabiliste, telles que les modèles de Markov cachés (HMM) ou les classifieurs bayésiens, pour analyser les mots, identifier leur catégorie grammaticale ou leur racine.

A) Applications typiques :

- Étiquetage morphosyntaxique : détermination automatique de la catégorie grammaticale des mots dans un texte
- Segmentation et racinisation : prédiction de l'emplacement des préfixes/suffixes en se basant sur des corpus annotés.
- Désambiguïsation lexicale : choix du sens correct d'un mot en fonction de son contexte statistique.

b) Limites :

- Nécessité de corpus volumineux et bien annotés.
- Difficulté à gérer les cas rares ou les dialectes peu représentés.
- Sensibilité à la qualité des données d'entraînement.

4-Conclusion de chapitre 3

La racinisation est une étape essentielle du traitement automatique de la langue arabe. Malgré sa complexité morphologique, des progrès importants ont été réalisés grâce aux combinaisons d'approches règles/statistiques. L'avenir de la racinisation arabe dépendra de l'intégration d'apprentissages contextuels intelligents, de corpus bien annotés, et de ressources lexicales enrichies.

Chapitre 04

Proposition d’un système hybride de racinisation

1-Introduction :

Le stemming ou racination consiste à réduire les formes fléchies et dérivées des mots à leur forme de base, appelée racine ou forme canonique. Plus précisément, il s'agit d'un processus automatique visant à supprimer les préfixes et suffixes d'un mot afin d'en extraire le radical. La racine peut être définie comme un morphème de base, ou un groupe de morphèmes liés, susceptibles d'accepter des affixes pour former de nouveaux mots.

Cependant, dans le cas de la langue arabe, la suppression des affixes peut souvent altérer le sens sémantique du mot. Cela s'explique par le fait que l'arabe repose essentiellement sur un système de racines trilitères ou quadrilitères et sur des modèles morphologiques (awzān) pour construire ses vocabulaires. Ainsi, pour qu'un stemmer soit efficace — qu'il soit léger ou fondé sur des règles —, il est impératif qu'il tienne compte de l'interaction complexe entre les schèmes morphologiques et les racines de base.

Les méthodes de racination en arabe peuvent être classées en deux grandes catégories selon le niveau d'analyse linguistique :

1. Les algorithmes légers (light stemmers) : ils se contentent de supprimer les préfixes et les suffixes sans chercher à identifier la racine exacte du mot.
2. Les approches fondées sur les règles morphologiques (root-based ou pattern-based) : elles visent à retrouver la racine originelle du

mot en s'appuyant sur l'analyse des modèles morphologiques et de la structure racinaire, offrant ainsi une plus grande précision.

La présente étude propose un nouveau système de racinisation innovant pour la langue arabe, baptisé **Blight**. Ce système adopte une approche hybride combinant les avantages des techniques de racinisation légère avec une analyse morphologique fondée sur les schèmes (awzān). L'objectif principal de cette approche est d'améliorer la précision de l'extraction des racines, tout en maintenant une complexité algorithmique raisonnable.

Ce chapitre présente en détail l'architecture du système Blight, en exposant les différentes étapes du processus de racinisation, de la normalisation des mots à l'analyse morphologique, en passant par la suppression conditionnelle des affixes et la détection des racines à l'aide de patrons. Une comparaison expérimentale avec les méthodes existantes (Khoja) est également proposée afin d'évaluer les performances du système sur un corpus représentatif.

2. Prétraitement du texte

Le prétraitement du texte constitue une étape essentielle pour améliorer l'efficacité des algorithmes de racinisation en arabe. L'objectif principal de cette phase est d'éliminer les mots bruités ou dépourvus de signification dans le corpus. En outre, le prétraitement permet de réduire les erreurs et d'augmenter la précision du processus de racinisation. Ainsi, l'ensemble des fichiers composant le corpus a été soumis aux opérations suivantes :

2.1 Normalisation du texte :

Le prétraitement du texte constitue une étape importante dans le processus de classification de textes. La phase de normalisation est l'une des principales étapes de prétraitement couramment effectuées dans les expériences liées au traitement de texte, parmi lesquelles figure le processus de racinisation (stemming). Dans notre processus de normalisation, nous utilisons la méthode proposée par Larkey et al. Ainsi, avant que la racinisation ne soit appliquée, le corpus ainsi que les requêtes de notre travail sont normalisés.

La tokenisation en langue arabe a été appliquée dans diverses solutions pour traiter l'ambiguïté des mots. Par exemple, certains caractères peuvent être écrits de différentes manières, comme le caractère (ء) appelé hamza, qui peut être composé de plusieurs formes différentes (أ ،إ ،آ). Cette variété dans l'écriture engendre une ambiguïté supplémentaire quant à la présence ou non de la hamza.

Par conséquent, en général, un seul token est attribué à chaque lettre à un moment donné, Par exemple, pour le mot (ألمستبعدون), le caractère (أ) est remplacé par (ا), ce qui transforme le mot en (المستبعدون).

2.2. Suppression des mots vides et des mots arabisés

La liste des mots vides comprend les mots d'un texte qui portent peu de sens. De plus, ces mots ne remplissent qu'une fonction syntaxique sans référer au contenu thématique du texte. Ces mots vides ont deux effets distincts sur le traitement automatique des langues (TAL), comme le soulignent Alshalabi et al 2013.

Ils peuvent influencer le processus de récupération de l'information en raison de leur fréquence relativement élevée, ce qui tend à réduire l'impact des variations de fréquence entre les mots moins courants, affectant ainsi le processus de pondération. La suppression des mots vides modifie également la longueur du document, ce qui influe par conséquent sur le calcul de la pondération.

Par ailleurs, ils affectent l'efficacité du traitement des textes en raison de leur nature et du fait qu'ils ne véhiculent pas de signification, ce qui peut entraîner un traitement important et peu productif.

En outre, les mots arabisés sont des emprunts étrangers provenant d'autres langues telles que le turc, le persan, l'anglais ou le français, entre autres. Des exemples typiques incluent les noms de voitures et de marques, les termes de mode contemporaine et les appareils électroniques, ainsi que les noms des mois du calendrier grégorien.

3. Développement du Stemmer Arabe (Blight)

La langue arabe repose sur un système de racines, à partir desquelles sont dérivés la majorité des noms et des verbes. Les stemmers arabes légers échouent fréquemment à extraire correctement la racine ou le radical d'un mot, en particulier lorsqu'il s'agit de néologismes ou de mots absents du lexique arabe standard. Ils rencontrent également des difficultés notables dans le traitement des pluriels brisés, ce qui entraîne souvent une racinisation incorrecte, altérant ainsi le sens original du mot.

Pour pallier ces limitations, notre objectif est d'améliorer le stemmer léger en arabe afin qu'il soit capable d'extraire les racines attendues avec une meilleure précision. Cela passe par une suppression

plus efficace des affixes, tout en préservant ceux qui constituent une partie intégrante du mot de base.

Les sections suivantes détaillent chacune de ces étapes.

- Remplacement des lettres إ, آ, et أ par ا.
- Remplacement du caractère final ى par ي.
- Remplacement du caractère final ة par ه.

3.1 Suppression des articles définis

Après l'étape de prétraitement, nous procédons à la suppression des articles définis situés en début de mot, selon des règles basées sur la longueur du mot. Contrairement aux approches antérieures où les articles définis étaient intégrés dans l'ensemble des préfixes à éliminer, notre méthodologie propose de les traiter comme une étape distincte du processus de racinisation.

Cette séparation vise à améliorer la précision de l'extraction des racines : en effet, la suppression isolée de l'article défini permet, dans de nombreux cas, de générer directement la racine correcte, tout en évitant les traitements excessifs qui pourraient conduire à des erreurs d'interprétation morphologique. Ainsi, cette approche permet de limiter les suppressions successives susceptibles d'altérer le radical initial du mot.

Les règles de suppression sont fondées sur la longueur du mot en entrée. Cette contrainte vise à éviter l'élimination de lettres structurelles fondamentales. Il est observé que, dans certaines situations, une fois l'article défini supprimé, le radical est immédiatement identifiable, rendant inutile tout traitement supplémentaire. À l'inverse, si cette suppression est suivie par l'élimination d'un préfixe ou d'un suffixe, le mot résultant peut ne plus refléter la racine correcte.

Pour résoudre ce problème, une étape de validation est introduite, consistant à comparer la sortie du stemmer avec une liste lexicale de racines de référence. Dans notre approche, nous implémentons l'Algorithme 1 pour effectuer cette suppression, en tenant compte des longueurs spécifiques des mots, comprises entre quatre et sept caractères. La quantité de lettres à supprimer dépend des critères présentés dans le Tableau 6.

Par exemple, à partir du mot normalisé « المستبعدون », l'article défini « ال » est supprimé, transformant le mot en « مستبعدون ».

Def4 =	["فبال", "لبال", "وبال", "كال", "وال"]
Def3 =	["بال", "فال", "وال", "كال", "ولل", "الا"]
Def2 =	["لل", "ال", "اا", "لي"]
Def1 =	["ل"]

Tableau 6 Listes d'affixes

Algorithme 1 de suppression des articles définis basé sur la longueur du mot

Entrée : mot arabe

Sortie : mot arabe sans article défini

1. Si la longueur du mot est supérieure ou égale à 7 et que le mot commence par un article défini de type **def4** ,
   alors supprimer cet article défini **def4** du début du mot.
2. Si la longueur du mot est supérieure ou égale à 6 et que le mot commence par un article défini de type **def3** ,
   alors supprimer cet article défini **def3** du début du mot.
3. Si la longueur du mot est supérieure ou égale à 5 et que le mot commence par un article défini de type **def2** ,
   alors supprimer cet article défini **def2** du début du mot.
4. Si la longueur du mot est supérieure ou égale à 4 et que le mot commence par un article défini de type **def1 ,**
   alors supprimer cet article défini **def1** du début du mot.

Liste des préfixes

P5 = ["وليست", "فليست", "فاست", "لاست", "افاست"]

P4 = ["أتست", "ويست", "فاست", "واست", "انهم", "والم", "باست", "الاس", "كمست", "والا"]

P3 = [" ", "مست", "ولت", "فلي", "فلن", "فلل", "فان", "يست", "تست", "است", "وسي", "وسن", "بمس", "وست", "فلأ""]

P2 = [" ", "لآ", "أف", "وو", "ول", "فا", "وب", "فن", "لي", "فت", "فب", "في", "أن", "أي", "او", "أت", "وي", "وت", "سي", "ست", "سن", "اا", "لا", "تت", "بت", "مم", "كت", "مت", "مس"]

P1 = ["ا", "ن", "ت", "ي", "و", "س", "ف", "ب", "ل"]

Liste des suffixes

S5 = ["ناكمو", "ناهما", "كموها"]

S4 = [" ", "موهم", "موهن", "ناكم", "ائكم", "اءنا", "اءهم", "نوهن", "اؤهم", "ونهم", "ائهم", "ناهم", "ياته", "اتهم", "اتها", "توهن", "ائنا", "ونكم", ""]

S3 = [" ", "ونني", "وهم", "وها", "نهم", "هما", "ناه", "اءه", "ونه", "نالك", "تان", "نها", "كما", "تنا", "يهم", "ائك", "انك", "ونك", "ونه", "وكم", "تهن", "تان", "هات", "نكم", "تهم", "تها", "وهن", ""]

S2 = [" ", "كن", "ها", "يغ", "هن", "كم", "تن", "ات", "وا", "تا", "تم", "نا", "ون", "ين", "ان", "اك", "وه", "وت", "وك", "هم", "ما", "ني", ""]

S1 = ["ه", "ة"]

3.2 Suppression des suffixes et des préfixes

Le processus de racinisation légère (light stemming) en arabe consiste généralement à supprimer un ensemble de préfixes, de suffixes ou les deux, voire à dériver ou infléchir le mot. Cette tâche demeure toutefois opaque et complexe, car elle présente plusieurs problèmes, notamment la suppression incorrecte de préfixes ou de suffixes. Ce phénomène peut entraîner la suppression de lettres essentielles du mot, conduisant ainsi à une altération du sens, voire à une modification totale du mot d'origine.

Bien que certaines méthodes de racinisation légère aient été proposées, nous n'avons pas identifié de travaux qui détaillent explicitement les affixes à supprimer. Dans ce travail, nous avons tenté de supprimer les chaînes de caractères susceptibles d'être des affixes,

dans la mesure où ces derniers apparaissent fréquemment au début ou à la fin des mots arabes.

Les algorithmes existants de racinisation légère suppriment généralement un nombre limité de préfixes et de suffixes, sans reconnaissance des motifs internes (infixes) permettant d'identifier la racine, ce qui conduit souvent à des résultats inefficaces.

Dans cette étude, et à la suite de l'analyse de plusieurs variantes de racinisation légère, nous proposons un nouvel algorithme de racinisation pour la langue arabe. Celui-ci tente d'identifier et de supprimer les affixes de différentes longueurs.

Lors de la phase de prétraitement du mot (مستبعدون), la suppression du suffixe (ون) en fin de mot permet d'obtenir (مستبعد).
L'objectif principal de l'algorithme 2 présenté est de supprimer les suffixes en fonction de règles basées sur la longueur des mots.

Cependant, dans de nombreux cas, cette suppression peut entraîner l'élimination de lettres fondamentales du mot. Par exemple, la suppression de la lettre (ن) du mot (بنان) donne (بنا), ce qui altère le sens du mot initial. De ce fait, ces cas particuliers ne sont pris en compte que dans la phase finale de l'algorithme principal.

Algorithme 2 : Algorithme de suppression des suffixes

Entrée : mot arabe

Sortie : mot arabe sans suffixes

1. Si la longueur du mot est ≥ **8** et que le mot se termine par un suffixe de type **S5** :
   → Supprimer le suffixe **S5** à la fin du mot
2. Si la longueur du mot est ≥ **7** et que le mot se termine par un suffixe de type **S4** :
   → Supprimer le suffixe **S4** à la fin du mot
3. Si la longueur du mot est ≥ **6** et que le mot se termine par un suffixe de type **S3** :
   → Supprimer le suffixe **S3** à la fin du mot
4. Si la longueur du mot est ≥ **5** et que le mot se termine par un suffixe de type **S2** :
   → Supprimer le suffixe **S2** à la fin du mot

Remarque :
Les suffixes d'un seul caractère (**S1**) ne sont pas supprimés si le mot a une longueur inférieure à 4 caractères.

3.2.1. Suppression des préfixes

Dans la figure suivante, l'Algorithme 3 présenté fournit des détails sur le processus de suppression des préfixes d'un mot. Comme mentionné précédemment, la suppression des préfixes dans notre approche n'est pas fondée sur la morphologie de la langue arabe, mais plutôt sur des règles liées à la longueur des mots et une liste spécifique de préfixes

Par exemple, le suffixe (و، د، ع، ب، ت، س، م، ن) a été supprimé de la fin du mot, ce qui a transformé le mot en une autre forme.

À l'étape suivante (troisième étape), le préfixe (م، ت) a été supprimé du début du mot, ce qui a réduit le mot à (بعـ).

Finalement, ce mot a été considéré comme la racine correcte (بَعَدَ), qui se distingue de l'adverbe (بَعْدُ), selon la vocalisation correcte du mot.

Algorithme 3 : Algorithme de suppression des préfixes

Entrée : mot arabe

Sortie : mot arabe sans préfixes

- Si la longueur du mot est ≥ **8** et que le mot commence par un préfixe de type **P5** :
  → Supprimer **P5** du début du mot
- Si la longueur du mot est ≥ **7** et que le mot commence par un préfixe de type **P4** :
  → Supprimer **P4** du début du mot
- Si la longueur du mot est ≥ **6** et que le mot commence par un préfixe de type **P3** :
  → Supprimer **P3** du début du mot
- Si la longueur du mot est ≥ **5** et que le mot commence par un préfixe de type **P2** :
  → Supprimer **P2** du début du mot
- Si la longueur du mot est ≥ **4** et que le mot commence par un préfixe de type **P1** :
  → Supprimer **P1** du début du mot

3.2.2. Algorithme de découpage (Blight)

Cette étude propose un algorithme de découpage léger pour les mots arabes visant à améliorer l'efficacité de l'extraction des racines. Cet algorithme renforce le processus de dérivation en ajoutant une liste étendue de préfixes et suffixes à supprimer, comme présenté dans le Tableau .

Les règles d'élimination de ces préfixes et suffixes reposent sur la longueur du mot, plutôt que sur les propriétés morphologiques traditionnelles de la langue

arabe, conformément aux Algorithmes 2 et 3. L'algorithme complet du stemmer arabe léger proposé, nommé Blight, est détaillé dans l'Algorithme 4

l'Algorithme 4 illustre comment la combinaison des étapes 1. 2 et 3 de l'Algorithme 4 — incluant les phases de prétraitement ainsi que la suppression des préfixes et suffixes — contribue à améliorer le processus de découpage selon une approche légère et efficace.

Lors de la première étape, le texte original est converti en une liste de mots. Ensuite, certaines lettres susceptibles d'apparaître sous différentes formes dans le texte arabe, telles que la hamza (ء), qui peut se présenter sous les formes (أ، إ، آ), sont unifiées en une seule forme (ا) afin de réduire l'ambiguïté lors du traitement. De plus, l'alif maqṣūra (ى) en fin de mot est convertie en ya (ي), et la tāʾ marbūṭa (ة) est remplacée par le ha (ه).

À la deuxième étape, les mots fonctionnels (stop words) ainsi que les mots arabicisés (emprunts linguistiques) sont supprimés.

Lors de la troisième étape, les mots obtenus sont comparés à une liste préétablie de mots d'origine (awzān). Une fois les trois lettres racines extraites, elles sont comparées à un lexique de racines arabes connues, Si la racine est validée, le mot est conservé sous cette forme ; dans le cas contraire, le processus se poursuit à l'étape suivante.

À la quatrième étape, les articles définis sont supprimés selon des règles fondées sur la longueur du mot, puis les préfixes et suffixes sont éliminés en suivant des règles similaires. Il est important de noter que les suffixes d'une seule lettre ne sont supprimés que si cela ne conduit pas à la suppression d'une lettre originale de la racine.

Enfin, la troisième étape est répétée pour assurer la validité des mots résultants après la suppression des affixes.

Les résultats produits par le stemmer Blight ont été évalués par comparaison avec ceux obtenus par d'autres stemmers arabes. La section suivante présente en détail la méthodologie et les résultats de cette évaluation.

Algorithme 4 : Développement d'un stemmer pour l'arabe (Blight)

Entrée : texte arabe

Sortie : texte arabe racinisé (après stemming)

Étape 1 : Normalisation des caractères

1.1 Supprimer les caractères non alphabétiques

1.2 Remplacer les formes de Hamza (آ، إ، أ) par une seule forme unifiée : ا

1.3 Remplacer la alif maqṣūra **(ى)** finale par **ي**

1.4 Remplacer la tāʾ marbūṭa (ة) finale par **ه**

Étape 2 : Nettoyage du texte

2.1 Supprimer les mots vides (stop words)

2.2 Supprimer les mots arabisés (d'origine étrangère)

Étape 3 : Suppression des affixes

3.1 Supprimer les articles définis selon les règles de l'Algorithme 1

3.2 Supprimer les suffixes selon les règles de l'Algorithme 2

3.3 Supprimer les préfixes selon les règles de l'Algorithme 3

3.4 3.4 comparés à une liste préétablie de mots d'origine

Étape 4 — Post-traitement morphologique :

Une fois les affixes supprimés, certains mots peuvent encore contenir des lettres résiduelles qui ne font pas partie de la racine, en particulier des variantes de l'*alif* ou des lettres ajoutées par dérivation.

L'étape de post-traitement permet d'affiner les racines extraites en appliquant plusieurs règles de nettoyage morphologique :

- **Suppression des alif ou ya maqṣūra finales** : certaines formes grammaticales ou orthographiques introduisent un ا ou un ى en fin de mot, Lui ne fait pas partie de la racine.

  Suppression des alif internes superflues : de nombreux dérivés arabes insèrent un ا dans le mot pour des raisons prosodiques ou morphologiques ; cette lettre n'appartient pas à la racine.

  Suppression conditionnelle des alif du milieu : certains schèmes (awzān) insèrent un ا au milieu de la forme dérivée ; cette fonction affine ce traitement.

  Suppression des alif après la lettre qāf (ق) : il est fréquent que des dérivés contiennent la séquence قا qui n'est pas toujours morphologiquement pertinente pour la racine.

Poids à trois lettres				**Poids à quatre lettres**	**Poids à cinq lettres**
مفعول	افعلال	فعلت	فعائل	مفعلل	مفتعل
افعال	استفعال	افعنلل	فعاليل	تفعلل	مفعَّل
فعلة	فعلعل	يفعول	فعلى	فعللة	مستفعل
افاعل	فعالى	فعول	فواعل	فعلل	مفاعيل
تفعيل	مفعل	فعال	فعولي	فعالل	فعالة
تفاعل	مفاعل	فاعلات	افعوعل	مفاعلة	
انفعال	متفعل	افعلل	فعلياء	فعلال	
فيعل	فاعلت	فعلاء	فاعل	فعاعيل	

	فعللان	فعيل	أفعل	فعنلى	فاعول
		يفتعل	فعلان	منفعل	افتعال
		افعول	فعلوان	يستفعل	تفتعل

Tableau *7* Classification des poids morphologiques arabes selon le nombre de lettres

4- Sources et Données Utilisées

Dans le cadre de l'évaluation de l'algorithme de racinisation proposé, une ressource essentielle a été utilisée : la liste des racines arabes développée par le chercheur Taha Zerrouki. Il s'agit d'une ressource libre largement adoptée dans le domaine du traitement automatique de la langue arabe, notamment pour les tâches de racinisation et d'analyse morphologique.

Cette liste intègre trois ressources lexicographiques complémentaires :

- Khalil Arabic Morphological Database : 7 503 racines
- Arabic Roots List de Taha Zerrouki : 7 946 racines
- Liste de racines intégrée au stemmer de Khoja : 4 748 racines

Ce découpage reflète la prédominance des racines trilittérales dans le système morphologique de l'arabe, tout en incluant également des racines quadrilittérales et quelques cas plus rares de racines pentalittérales.

Cette diversité est cruciale pour garantir une couverture morphologique optimale dans le cadre du développement d'un système de racinisation performant.

4-1 Répartition des racines par longueur

Longueur du radical	**Khalil**	**Safari**	**Khoja**
2 lettres	0	16	0
3 lettres	5 674	5 582	3 822
4 lettres	1 829	2 294	923
5 lettres	0	54	3
Total	**7 503**	**7 946**	**4 748**

Tableau 8 : Distribution des racines par longueur selon les ressources

4-2 Fusion de ressources morphologiques arabes

Afin de garantir la qualité et la robustesse de notre système de racinisation, nous avons construit un lexique de racines de référence à partir de ces différentes sources complémentaires.

Nous avons ensuite procédé à une phase de fusion et de nettoyage de ces listes, visant à produire une liste unifiée et consolidée de racines correctes, qui servirait de base pour l'évaluation de notre système.

Après déduplication, normalisation et contrôle de cohérence des racines issues de ces trois corpus, nous avons obtenu une liste finale de racines correctes.

Longueur du radical	**Nombre de racines**
2 lettres	17

3 lettres	7 100
4 lettres	3 286
5 lettres	54
Total	**10 457**

Tableau 9: Distribution finale des racines correctes

Cette liste consolidée a été utilisée comme référence lexicale pour valider les racines générées par notre système de racinisation, permettant ainsi d'évaluer de manière rigoureuse sa précision et sa couverture morphologique.

- **7 100 racines trilittères**, qui représentent la majorité des racines dans la langue arabe,
- **3 286 racines quadrilitères**, plus complexes et moins fréquentes.
- **54 racines pentalittérales**, cas rares mais inclus pour assurer une couverture morphologique complète.
- **17 racines bilittérales**, généralement considérées comme vestiges ou formes spécifiques dans certaines catégories lexicales

Les racines sont présentées sous la forme d'un fichier texte, avec une racine par ligne, ce qui facilite leur intégration dans les traitements automatiques. De plus, elles sont normalisées, c'est-à-dire que toutes les formes de la Hamza sont unifiées en (ء) afin de garantir une cohérence morphologique et orthographique dans les processus de comparaison ou de correspondance.

Cette ressource a servi de référence de validation pour :

- Vérifier l'exactitude des racines extraites automatiquement par l'algorithme,
- Mesurer la performance du stemmer selon des métriques standard (précision, rappel, F-mesure).

5 CORPUS UTILISER

5-1 Extraits de Wikipédia en arabe :

Les extraits de Wikipédia en arabe désignent un ensemble de documents ou de textes issus des articles de l'encyclopédie Wikipédia, rédigée en langue arabe. Ce corpus constitue une ressource linguistique précieuse, fréquemment exploitée dans le cadre de recherches en traitement automatique de la langue naturelle (TALN), en racinisation, indexation, ou encore pour l'entraînement de modèles de classification automatique et de résumé de texte.

Ce corpus se distingue par sa richesse lexicale et sa diversité thématique, couvrant des domaines variés tels que la géographie, l'histoire, les sciences ou la culture. Utilisant majoritairement un arabe standard moderne (MSA), il offre un matériau linguistique pertinent pour les travaux académiques et les expérimentations en linguistique computationnelle.

Les informations relatives à ce corpus, compilées par Motaz Saad du département d'informatique de la Faculté de technologie de l'information de l'Université islamique de Gaza, sont résumées dans le Tableau 10. Initialement lancé en 2003 avec 655 articles, le corpus a connu une

croissance rapide, atteignant 459 208 articles au 20 janvier 2017, puis s'étendant à 953 507 documents extraits au 20 septembre 2019.

En termes de volume, ce corpus comprend 123 079 742 mots (*tokens*) et un vocabulaire total de 4 437 963 mots uniques (*types*).

Le corpus est disponible en ligne sur https://github.com/motazsaad/arWikiExtracts pour la communauté de recherche.

5-2 Extraits du corpus du 20 septembre 2019

Documents	Words	Vocabulaire
953,507	123,079,742	4,437,963

Tableau 10 : Statistiques générales du corpus Arabic Wikipedia

ORDER	Word		Frequency	ORDER	Word	Frequency
0	في		5312695	50	مثل	133671
1	من		3181089	51	مدينة	132187
2	على		1332843	52	بعض	131032
3	إلى		1003987	53	تحت	130221
4	هو		670619	54	وكان	129045
5	أن		653989	55	بلغ	126150
6	عام		582445	56	وهي	124714
7	التي		536241	57	تعداد	117963
8	مع		508169	58	المملكة	113849
9	و		506597	59	عبد	113425
10	عن		486752	60	أكثر	112491
11	أو		451778	61	سكانها	112014
12	،		419063	62	له	111192
13	هي		412564	63	أنه	110787
14	كان		380104	64	يمكن	110273
15	.		342948	65	وكانت	107096
16	بين		326676	66	تقع	107089
17	ما		310063	67	المتحدة.	106634
18	هذه		305964	68	لاعب	105986
19	بعد		289098	69	ومن	104457

20	الذي		287346	70	:	103676
21	كانت		280527	71	عشر	102766
22	حيث		273788	72	قدم	101331
23	هذا		271383	73	محمد	100694
24	عدد		263233	74	أي	99390
25	بن		239084	75	يتم	96128
26	وقد		216643	76	بشكل	95970
27	ولد		216060	77	والتي	94929
28	-		215931	78	فيها	93238
29	سنة		215680	79	ونسبة	93236
30	كما		215218	80	الله	92375
31	الولايات		214812	81	العديد	91658
32	تم		213640	82	أول	90553
33	ذلك		205158	83	أما	89793
34	لا		194852	84	ولكن	87020
35	قبل		190329	85	بـ	86839
36	وفي		184865	86	)	86587
37	خلال		184817	87	منطقة	85899
38	حتى		177186	88	إحدى	83989
39	قد		172797	89	منها	83139
40	وهو		162748	90	يكون	82514
41	ثم		152857	91	عند	82475
42	غير		148585	92	القرن	81909
43	نسمة		148568	93	أحد	81603
44	نسبة		144694	94	عندما	81547
45	المتحدة		144227	95	أمريكي،	80982
46	كرة		143590	96	منذ	79546
47	كل		143502	97	لكرة	79456
48	حسب		138799	98	الذين	79440
49	لم		138753	99	جميع	79175

Tableau 11 : les mots les plus fréquents dans le corpus Arabic Wikipedia.

Le Tableau 11 présente les 100 mots les plus fréquents dans le corpus Arabic Wikipedia. Ces mots représentent majoritairement des outils grammaticaux, prépositions, pronoms ou mots fonctionnels fréquents dans la langue arabe, tels que « في » (5 312 695 occurrences), « من » (3 181 089) ou encore « على » (1 332 843). On observe également la présence de noms propres, de verbes courants (« كان », « ولد », « تم »), ainsi que de mots liés à des thématiques spécifiques comme la géographie (« منطقة », « المملكة », « مدينة ») ou la démographie (« عدد », «

تعداد » ,« سكانها »). Ce classement illustre non seulement la richesse lexicale du corpus, mais aussi les structures syntaxiques récurrentes utilisées dans les articles de la Wikipédia arabe.

6 Outils et langages utilisés :

Le développement et l'évaluation du stemmer arabe **Blight** ont été réalisés à l'aide des technologies et outils suivants :

- **6-1 Langage de programmation : Python**
  Python a été choisi pour sa richesse en bibliothèques de traitement du langage naturel (NLP), sa lisibilité, et sa large communauté. Il a permis d'implémenter efficacement les différentes étapes du système de racinisation : normalisation, suppression conditionnelle des affixes, détection des schèmes morphologiques (awzān), et évaluation.
- **6-2 Environnement de développement : Google Colab**
  Google Colaboratory a été utilisé comme plateforme de développement, permettant d'exécuter le code Python dans un environnement cloud sans configuration locale, avec un accès facile aux fichiers et une exécution rapide sur de grands volumes de données.

- **6-3 Bibliothèques principales :**
  - re : pour la manipulation des expressions régulières et la détection de motifs morphologiques.

- nltk : pour l'utilisation des stopwords arabes et certaines fonctions de traitement du texte.
- Collections : pour le comptage des fréquences et l'analyse des résultats.

7 Évaluation expérimentale et comparaison avec l'algorithme de Khoja

7.1. Protocole expérimental

Pour évaluer les performances de l'algorithme de racinisation hybride proposé (Blight), une série de tests a été menée sur un large corpus de textes arabes. L'objectif est de comparer cet algorithme à l'algorithme de type Khoja, qui constitue une référence classique dans le domaine.

Le corpus utilisé contient plus de 7,9 millions de mots (7 951 541), provenant de diverses sources textuelles en arabe moderne. Ces textes sont volontairement choisis sous une forme brute, non structurée, afin de simuler des conditions réalistes d'utilisation.

Avant l'application des algorithmes, un prétraitement est effectué sur le corpus :

- Normalisation des caractères arabes (suppression des variantes orthographiques) ;
- Suppression des diacritiques et autres marques diacritiques ;
- Tokenisation en mots individuels.

Une liste de racines de référence validées a été constituée à partir de ressources lexicographiques reconnues, afin de permettre une évaluation fiable.

7.2. Méthodologie d'évaluation

Les résultats des algorithmes sont évalués à l'aide des métriques suivantes :

- **True Positives (TP)** : nombre de racines correctes extraites par l'algorithme.
- **False Positives (FP)** : nombre de racines incorrectes produites.
- **False Negatives (FN)** : nombre de racines attendues qui n'ont pas été extraites.

À partir de ces valeurs, les mesures suivantes sont calculées :

- **Précision (Precision)** : proportion de racines extraites qui sont correctes calculée par la formule :

$$\text{Précision} = \frac{TP}{TP + FP}$$

- **Rappel (Recall)** : proportion de racines correctes extraites par rapport à l'ensemble des racines attendues calculée par la formule :

$$\text{Rappel} = \frac{TP}{TP + FN}$$

- **F1-score** : moyenne harmonique entre la précision et le rappel, donnant une mesure globale de la qualité calculée par la formule :

$$F1\text{-score} = 2 \times \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}}$$

Les algorithmes sont appliqués au même corpus, en utilisant le même jeu de racines de référence.

7-3 Comparaison des performances : Blight et Khoja stemmer

Afin d'évaluer l'efficacité du stemmer arabe proposé **Blight**, nous avons mené une expérimentation comparative avec le célèbre algorithme de racinisation de Khoja, considéré comme une référence dans le domaine du traitement morphologique de l'arabe.

L'évaluation a été réalisée sur un large corpus de plus de 7,8 millions de mots originaux, en utilisant une liste de racines de référence et des métriques standards de l'analyse de performance.

7-4 Résultats obtenus

Système	TP	FP	FN	Précision	Rappel	F1-score
BLIGHT	5 367 702	2 474 772	4 165	**68.44 %**	**99.92 %**	**81.24 %**
KHOJA	4 867 112	3 084 429	5 177	61.21 %	99.89 %	75.91 %

Tableau 12 Résultats obtenus entre Blight et Khoja stemmer

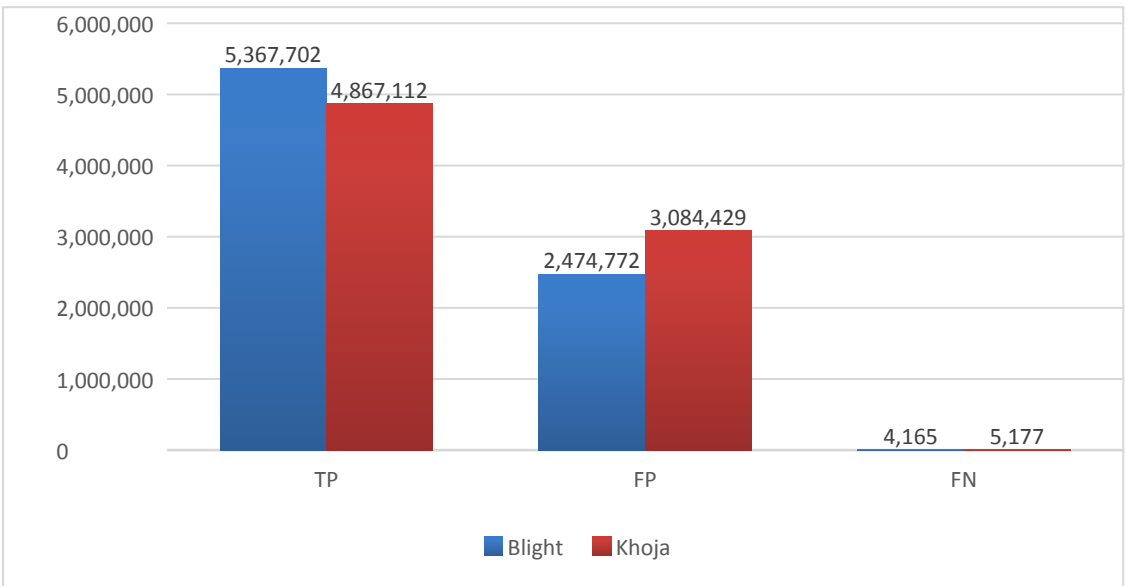


Figure 5 présentations graphiques des résultats de TP ,FP et FN

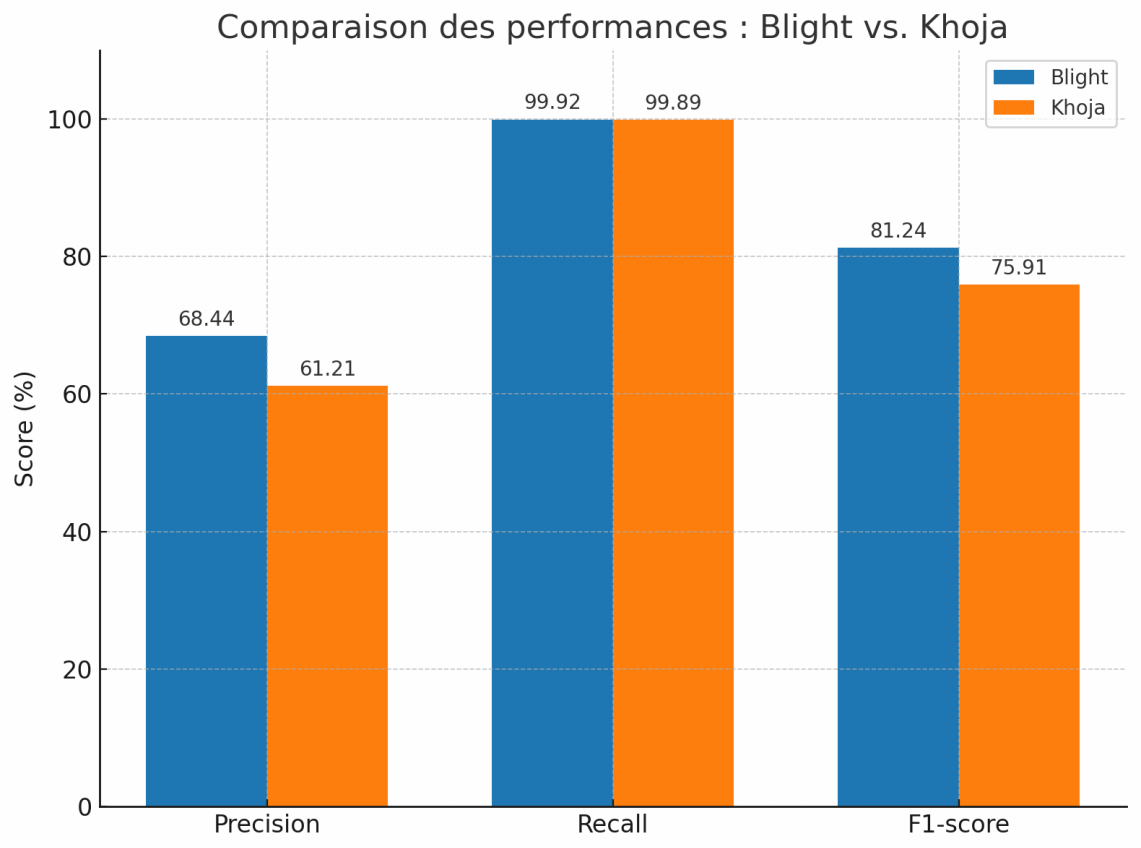


Figure 6 Comparaison des performances entre les stemmers Blight et Khoja

7-5 Analyse comparative

Les résultats expérimentaux obtenus dans le cadre de cette étude permettent de mettre en évidence les apports de l'approche hybride proposée par le stemmer *Blight* en comparaison avec le stemmer morphologique de référence, celui de *Khoja*.

Sur un corpus de test conséquent, comprenant plus de **7,8** millions de mots arabes, *Blight* affiche des performances globales supérieures à celles du stemmer de *Khoja*. Le score F1 obtenu par *Blight* s'élève à **81,24 %**, contre **75,91 %** pour *Khoja*, traduisant ainsi un meilleur équilibre entre précision et rappel.

Dans le détail, *Blight* atteint une précision de **68,44 %**, sensiblement supérieure à celle de *Khoja* (**61,21 %**). Cette amélioration reflète la capacité du stemmer proposé à limiter le nombre de faux positifs (FP), en produisant des racines linguistiquement plus correctes et plus proches des racines de référence. Le rappel reste quant à lui élevé et comparable entre les deux approches (**99,92 %** pour *Blight* contre **99,89 %** pour *Khoja*), ce qui démontre que le stemmer *Blight* parvient également à couvrir l'essentiel des racines attendues, sans perte significative de couverture lexicale.

L'amélioration observée en termes de précision peut être attribuée à plusieurs facteurs :

- l'intégration d'une stratégie fine de suppression conditionnelle des affixes,
- l'utilisation de motifs morphologiques (*awzān*) adaptés à la morphologie arabe,

- et l'application de règles de post-traitement ciblées, telles que celles implémentées dans la fonction post_blight_rules.

Ces mécanismes permettent de mieux contrôler le processus de racinisation, en évitant notamment les erreurs de sur-stemming fréquentes dans les approches purement morphologiques.

En revanche, le stemmer de *Khoja*, bien qu'étant une référence largement utilisée dans la littérature, tend à générer davantage de faux positifs du fait de sa dépendance aux règles morphologiques générales et de son manque de filtrage contextuel.

En conclusion, les résultats obtenus démontrent l'intérêt et l'efficacité de l'approche hybride développée dans *Blight*, qui combine la légèreté et la rapidité d'un stemmer léger avec la rigueur linguistique d'une analyse morphologique par schèmes. Cette combinaison permet d'atteindre un meilleur compromis entre couverture lexicale et qualité des racines extraites, et représente ainsi une contribution pertinente au domaine du traitement automatique de la langue arabe.

8-Conclusion du Chapitre 4

Au terme de cette étude comparative, les résultats obtenus confirment la pertinence et l'efficacité de l'approche hybride incarnée par le stemmer *Blight*, dans le champ exigeant du traitement automatique de la langue arabe.

L'analyse approfondie des performances met en évidence des avancées notables, en particulier en matière de qualité des racines extraites et de précision du processus de racinisation, tout en maintenant

un niveau de rappel exceptionnellement élevé. Ces résultats traduisent la capacité du système à modéliser avec finesse les schémas morphologiques complexes de l'arabe, tout en limitant les erreurs structurelles.

Comparé au stemmer de Khoja — référence historique et largement utilisée — *Blight* démontre une maîtrise plus fine du découpage morphologique, fruit d'une intégration harmonieuse de plusieurs stratégies complémentaires : normalisation avancée, suppression conditionnelle des affixes, exploitation systématique des *awzān* (modèles morphologiques), et application de règles de post-traitement linguistiquement fondées.

Ces résultats soulignent l'intérêt de poursuivre les recherches dans cette voie hybride, qui conjugue la légèreté algorithmique des méthodes classiques avec une compréhension enrichie des structures propres à la langue arabe. Ils illustrent également la nécessité d'adapter les techniques de traitement automatique aux spécificités morpho-syntaxiques profondes de chaque langue.

Dans le cas de l'arabe, langue caractérisée par une richesse morphologique exceptionnelle et par une grande variabilité dérivationnelle, l'intégration explicite de savoirs linguistiques dans les algorithmes représente une piste particulièrement prometteuse. C'est dans cette perspective que le stemmer *Blight* constitue une contribution concrète et originale, en ouvrant la voie vers des outils plus précis, plus robustes et mieux adaptés aux besoins croissants des applications réelles en TALN pour la langue arabe.

Conclusion Générale

Conclusion Générale

Le traitement automatique de la langue arabe (TALA) constitue un champ de recherche riche et complexe, à la croisée de l'informatique, de la linguistique et de l'intelligence artificielle. La langue arabe, avec sa richesse morphologique, sa structure dérivationnelle non-concaténative, l'absence fréquente de voyelles diacritiques, et sa grande diversité dialectale, présente des défis spécifiques qui ne se posent pas de la même manière pour les langues occidentales. Parmi les tâches fondamentales du TALA, la racinisation joue un rôle clé pour de nombreuses applications, telles que la recherche d'information, la traduction automatique ou encore l'analyse de sentiments.

Au fil de ce travail, nous avons exploré les fondements du traitement automatique de la langue naturelle (TALN), les particularités linguistiques de la langue arabe, ainsi que les différentes approches classiques et modernes de racinisation. Nous avons mis en évidence les limites des méthodes purement basées sur des règles morphologiques, et montré que les approches statistiques et hybrides offrent une meilleure flexibilité et une capacité de généralisation accrue, notamment lorsqu'elles s'appuient sur des corpus représentatifs et des ressources linguistiques comme l'Arabic WordNet.

Dans cette perspective, nous avons proposé et développé un stemmer hybride pour l'arabe, baptisé Blight. Celui-ci combine :

- une normalisation avancée des textes,
- une suppression conditionnelle et contrôlée des affixes,

- une exploitation systématique des schèmes morphologiques (awzān),
- et des règles de post-traitement linguistiquement motivées.

Les résultats expérimentaux comparant Blight au stemmer de Khoja, une référence dans le domaine, démontrent les apports de notre approche. Blight améliore significativement la précision tout en maintenant un rappel très élevé, traduisant ainsi une meilleure maîtrise du processus de racinisation et une réduction notable des erreurs. Cette étude confirme l'intérêt d'une approche hybride, qui combine la puissance des méthodes statistiques avec une prise en compte fine de la structure linguistique propre à l'arabe.

Au-delà des indicateurs chiffrés, notre recherche illustre l'importance d'adapter les techniques de traitement aux spécificités profondes de chaque langue. Pour l'arabe, langue à forte complexité morphologique, l'intégration de connaissances linguistiques explicites dans les algorithmes constitue une voie prometteuse pour améliorer la qualité des traitements automatiques.

Ce travail ouvre également plusieurs perspectives :

- l'extension du stemmer Blight aux variantes dialectales,
- l'intégration de modèles neuronaux avancés, tels que BERT ou AraBERT,
- et le renforcement des capacités de prise en compte du contexte sémantique global.

Nous espérons que cette contribution apportera une pierre utile à l'édifice des outils dédiés au traitement automatique de la langue arabe, et qu'elle inspirera de futurs travaux visant à construire des systèmes plus précis, plus robustes et mieux adaptés aux besoins concrets des applications en TALN.

BIBLIOGRAPHIE:

1 [Miled97] H. Miled, C. Olivier, M. Cheriet, K. Romeo-Pakker. Une Méthode Rapide de Reconnaissance de l'Ecriture Arabe Manuscrite. Seizième colloque gretsi, 15-19 septembre, Grenoble, 1997.

2 Abainia, K., Djeddi, C., & Belaïd, A. (2017). Arabic stemming: challenges and solutions. ACM Computing Surveys (CSUR), 50(6), 1–33.

3 Abdelali, A., et al. (2016). Farasa: A Fast and Accurate Arabic Part-of-Speech Tagger. Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016).

4 Al-Kharashi, I., & Evens, M. (1994). *Comparing words, stems, and roots as indexing terms in an Arabic information retrieval system*. JASIS

5 Barbary, H., et al. (2020). CAMeL Tools: A Unified Toolkit for Arabic NLP. Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020).

6 Buckwalter, T. (2002). *Buckwalter Arabic Morphological Analyzer*. Linguistic Data Consortium.

7 character recognition systems. Jurnal Teknologi, 36(E), Universiti Teknologi Malaysia, Jun 2002

8 Choi, J., et al. (2018). Named Entity Recognition for Arabic Using Deep Learning. Proceedings of the 27th International Conference on Computational Linguistics.

9 D.Arrivault. Apport des Graphes dans la Reconnaissance Non-Contrainte de Caractères Manuscrits Anciens. Thèse de doctorat, Université de Poitiers, 2002.

10 Darwish, K., & Mubarak, H. (2016). Farasa: A fast and furious segmenter for Arabic. In Proceedings of NAACL.

11 Darwish, K., et al. (2009). Light Stemmer: A Modern Arabic Stemmer for Search Engines. Proceedings of the 7th International Conference on Arabic Language Processing.

12 Devlin, J., et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019.

13 El-Haj, M. (2012). Arabic Word Sense Disambiguation Using Contextual Features. Proceedings of the 2012 International Conference on Natural Language Processing.

14 Habash, N., et al. (2009). Arabic Dialect Identification and Sentiment Analysis. Journal of Natural Language Engineering.

15 Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing (3rd ed.). Stanford University.

16 K.Jumari, M.A Ali. A survey and comparative evaluation of selected off-line arabic handwritten

17 Khoja, S., & Garside, R. (1999). Stemming Arabic Text. University of Lancaster.

18 Larkey, L. S., Ballesteros, L., & Connell, M. E. (2007). Light stemming for Arabic information retrieval. Arabic Computational Morphology. Springer, Dordrecht.

19 Burrow. Arabic Handwriting Recognition. School of Informatics, University of Edinburgh, 2004.

20 Pasha, A., Al-Badrashiny, M., Diab, M., et al. (2014). MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. LREC.

21 Siham Boulaknadel « Traitement Automatique des Langues et Recherche d'Information en langue arabe dans un domaine de spécialité :Apport des connaissances morphologiques et syntaxiques pour

l'indexation », Thèse de doctorat , Université de Nantes,2008.

22 Taghva, K., Beckley, R., & Sadeh, M. (2005). A stemming algorithm for the Arabic language. ITCC.

23 Zerrouki, T., & Balla, A. (2017). Tashaphyne: A Python Light Stemmer for Arabic. GitHub Repository.

24 Taha Zerrouki, Amar Balla. *Tashkeela: Novel corpus of Arabic vocalized texts, data for auto-diacritization systems*. Data in Brief, 2017.

25 Motaz Saad and Basem Alijla (2017). *WikiDocsAligner: an off-the-shelf Wikipedia Documents Alignment Tool*. in The Second Palestinian International Conference on Information and Communication Technology (PICICT 2017).