Contributions à la correction automatique des erreurs syntaxiques dans la langue Arabe

Moukrim Chouaib

N° d'ordre : 86/2020

Université Hassan II de Casablanca

Thèse de Doctorat

Présentée par :

Mr Chouaib MOUKRIM

Spécialité :

Informatique

Sujet de la thèse :

Contributions à la correction automatique des erreurs syntaxiques dans la langue Arabe

Thèse présentée et soutenue à Casablanca le 10/01/2020 à 10h00 Amphi 5, devant le jury composé de :

Pr. BENABBOU Faouzia	**FSBM**	**Présidente et Rapporteuse**
Pr. HAMDANI Abdelfattah	**INREA Rabat**	**Rapporteur**
Pr. AZOUAZI Mohamed	**FSBM**	**Rapporteur**
Pr. SADIQ Abdelalim	**FS Kenitra**	**Examinateur**
Pr. BENLAHMAR El Habib	**FSBM**	**Co- Directeur de thèse**
Pr. TRAGHA Abderrahim	**FSBM**	**Directeur de thèse**

Etablissement : Faculté des Sciences Ben M'SIK
CEDoc : Sciences et Applications.

Nom du laboratoire : Laboratoire Technologie d'Information et Modélisation (LTIM)

قال تعالى :

[سورة يوسف : 2]

ملخص

تعدّ المعالجة الآليّة للّغات الطّبيعية مجالًا متناميًّا للبحث في علوم الحاسوب والعلوم المعرفيّة، وذلك باستخدام العديد من الأساليب التّجريبيّة. فالنّحو هو واحد من أهمّ خصائص اللّغة الطّبيعيّة. حيث انّه يحتوي على مجموعة من القواعد الهيكليّة الّتي يتمّ استعمالها بين المتحدّثين بلغتهم الأصليّة للسّماح بالتّواصل السّلس. التّصحيح الآلي للأخطاء النّحوية هو من بين تطبيقات مجال المعالجة الآليّة للّغات الطّبيعيّة التي تقوم بتصحيح الأخطاء النّحويّة في جملة معيّنة باستخدام النّماذج الحاسوبيّة (علوم الحاسوب) واللّسانيات.

تتناول هذه الأطروحة إشكاليّة الأخطاء النّحوية في اللّغة العربيّة ولتحقيق هذا الهدف، اقترحنا حلّين:

يستند الحلّ الأول الى مقاربة جديدة تعتمد على التوليد الالي للجمل الصّحيحة. أوّلاً، نستخلص الكلمات من الجملة الأصليّة؛ ثمّ، من خلال هذه الكلمات وبفضل وصف منطقيّ لقواعد اللّغة العربيّة في الأنطولوجيا، نقوم بإنشاء جميع الجمل الممكنة الصّحيحة من النّاحية النّحويّة. ونقارن بعد ذلك الجملة الأصليّة مع الجمل التي تمّ إنشاؤها للكشف عن الأخطاء المحتملة. أخيرًا، إذا لم يعثر النّظام في مرحلة التّصحيح على أيّ جملة تشبه الجملة الأصليّة، يتمّ اقتراح الجمل الصّحيحة تلقائيًّا. تمّ إجراء اختبارات ناجحة باستخدام مجموعة من الجمل العربيّة. حيث حقق النّظام معدّل دقّة حوالي 92 في المائة ومعدل استرداد حوالي 84 في المائة، على ضوء ذلك و من خلال النّتائج التي تمّ الحصول عليها، نستنتج أن هذه المقاربة واعدة.

نتعامل في الحلّ الثّاني مع تصحيح أخطاء الإعراب، الّذي هو جزء من الأخطاء النّحويّة باستخدام المحلّل النّحوي ("Stanford_Parser") بالإضافة إلى الأنطولوجيا التي تحتوي على قواعد اللّغة العربيّة. أوّلاً، نقوم بتقسيم النّص إلى جمل. ثانياً، نستخلص سمات كل كلمة على حدة بالإضافة الى العلاقات النّحوية المحصّل عليها من هذا المحلّل النّحوي؛ ثمّ ندرس العلاقات التي تمّ الحصول عليها ونقارنها بالعلاقات الموجودة في الأنطولوجيا. أخيرًا، نقوم بمقارنة الجملة الأصليّة مع الجملة التي تمّ تصحيحها لاكتشاف الخطأ.

الكلمات المفتاحيّة : اللّغة العربيّة، الأخطاء النّحوية، المعالجة الآليّة للّغات الطّبيعيّة، الأنطولوجيا، التّحليل النّحوي.

Résumé

Le Traitement Automatique de Langues (TAL) est un domaine de recherche en plein essor en informatique et en sciences cognitives utilisant de nombreuses méthodes expérimentales. La syntaxe est l'une des propriétés les plus importantes de la langue. Elle contient un ensemble de règles structurelles par lesquelles les unités linguistiques se combinent en phrases qui sont partagées entre les locuteurs natifs afin de permettre une communication fluide. La correction automatique des erreurs syntaxiques est parmi les applications de TAL, elle vise à corriger les erreurs syntaxiques dans une phrase source donnée en se basant sur des modèles informatiques et linguistiques.

La présente thèse de doctorat traite le problème des erreurs syntaxiques dans la langue arabe. Pour réaliser cet objectif nous avons proposé deux solutions :

La première est une nouvelle approche basée sur la génération automatique de phrases correctes. Tout d'abord, nous extrayons les mots de la phrase concernée. Ensuite, à partir de ces mots et grâce à une description logique des règles de la grammaire arabe dans l'ontologie nous générons toutes les phrases possibles. Nous comparons ensuite la phrase d'origine avec les phrases (correctes syntaxiquement) générées pour détecter d'éventuelles erreurs. Enfin, dans la phase de correction, si le système ne trouve aucune phrase qui ressemble à la phrase d'origine, les phrases alternatives correctes seront automatiquement proposées. Des tests réussis ont été effectués à l'aide d'un corpus de phrases arabes. Le système mis en œuvre a atteint un taux de précision d'environ 92% et un taux de rappel d'environ 84%. En observant les résultats obtenus, nous concluons que cette approche est prometteuse.

La deuxième solution traite la correction des erreurs syntaxiques, particulièrement de désinence casuelle, en utilisant l'analyseur syntaxique «Stanford Parser » ainsi que l'ontologie de la grammaire Arabe qui contient les règles de la langue arabe. En premier lieu, nous segmentons le texte en phrases. En second lieu, nous extrayons les traits morpho-syntaxiques de chaque mot avec les relations syntaxiques provenant du parseur Stanford. Ensuite, nous traitons les relations obtenues avec l'ontologie. En dernier lieu, nous comparons la phrase d'origine avec la phrase corrigée afin de détecter l'erreur.

Mots clés : La langue Arabe, les erreurs syntaxiques, le traitement automatique des langues, les ontologies, l'analyse syntaxique, Stanford Parser.

Abstract

Natural Language Processing (NLP) has been a growing area of research in computer science and cognitive sciences, using many experimental methods. The syntax is one of the most important properties of natural language. It contains a set of structural rules that are shared among native speakers to allow smooth communication. Automatic syntax error correction is an NLP application that attempts to correct syntactical errors in a given source sentence using computational (computer sciences) and linguistic models.

This thesis explores the problem of syntactic errors in the Arabic language. To achieve this goal, we have proposed two solutions:

The first is a new approach based on the automatic generation of correct sentences. First, we extract the words from the considered sentence and we then generate all the possible sentences that are syntactically correct; based on a logical description of the rules of Arabic grammar in the ontology. We will afterwards compare the original sentence with the generated sentences to detect any eventual errors followed by the correction phase. In case the system has not found a sentence that looks similar to the original sentence, the correct alternative sentences are automatically offered. Successful tests were performed using a set of Arabic sentences. The implemented system achieved an accuracy rate of about 92% and a recall rate of about 84%. By observing the results obtained, we conclude that this approach is promising.

The second solution deals with the correction of syntactic errors, particularly of the case ending using “Stanford Parser” as well as an ontology which contains the rules of the Arabic language. First, we segment the text into sentences. Secondly, we extract the annotations of each word with the syntactic relations coming from this parser. Then we treat the relations obtained with our ontology. Finally, we compare the original sentence with the corrected one to detect the error.

Keywords: Arabic language, syntactic errors, natural language processing, ontologies, syntactic parsing, Stanford Parser.

Remerciements

En premier lieu, je remercie Dieu, le Tout-Puissant pour ses faveurs et ses grâces, de m'avoir donné le courage et la patience de mener ce travail.

Au terme de ce travail, je tiens à exprimer ma profonde gratitude à Monsieur Abderrahim TRAGHA Professeur à la faculté des sciences Ben M'sik, pour les conseils qu'il m'a prodigué, son judicieux encadrement ainsi que le soutien qu'il m'a apporté tout au long de la réalisation de cette these.

Ainsi, j'exprime ma profonde reconnaissance à Monsieur El Habib BENLAHMAR, Professeur à la faculté des sciences Ben M'sik, pour son Co-encadrement, ses encouragements, et ses précieuses directions qui m'ont permis de mener à bien ce travail.

Je tiens à exprimer mes gratitudes à tous les professeurs de département mathématiques & informatique qui m'ont assisté tout au long de ce travail.

Mes remerciements s'adressent également à Madame BENABBOU Fouzia, HAMDANI Abdelfettah, AZOUAZI Mohamed qui ont bien voulu rapporter cette thèse. Je leur suis extrêmement reconnaissante d'avoir donné de leur temps pour évaluer ce travail.

Je suis honoré par la présence de Monsieur SADIQ Abdelalim, professeur à la Faculté des sciences Kenitra Université Ibn Tofail. Qu'il trouve ici mes sincères remerciements d'avoir accepté d'être examinateur de ce travail.

Mes remerciements ainsi que mon infection à mes très chers parents pour leur irremplaçable et inconditionnel soutien. Merci pour les sacrifices que vous avez consentis pour mon instruction et mon bien être. Qu'Allah vous garde, vous procure santé, bonheur, et longue vie.

Un grand merci à mon frère et mes soeurs qui m'ont supporté tout au long de la réalisation de ce travail.

Chouaib MOUKRIM

Table des matières

Liste des abréviations

C

CoNLL: Conference on Natural Language Learning

H

HOO: Helping Our Own

M

ML: Machine Learning

MT: Machine Translation

N

NLP: Natural Language Processing

O

OWL: Ontology Web Language

OSA: Ontologie de la syntaxe arabe

P

POS: Part Of Speech

R

RDF: Resource Description Framework

RDFS: Resource Description Framework Schema

S

SOA: l'architecture orientée services

SPARQL: standard Protocol and RDF Query Language

SMT: Statistical Machine Translation

SUMO: suggested Upper Merged Ontology

SW: Web sémantique

W

W3C: World Wide Web Consortium

X

XML: extensible markup language

Liste des figures

Liste des tableaux

Introduction générale

MOTIVATIONS

Ces dernières années, la correction automatique des erreurs syntaxiques est devenue de plus en plus utile pour la recherche dans le domaine de Traitement Automatique des Langues (TAL). Parmi les raisons fortes, nous pouvons citer : l'émergence d'une gamme d'applications pratiques, l'intégration de composants pour la vérification orthographique et syntaxique à de nombreuses applications tels que les logiciels de traitement de texte, la messagerie électronique et les navigateurs Web. Ces composants facilitent ainsi la rédaction de textes sans erreurs, l'incorporation de solutions plus sophistiquées dans des systèmes pour la correction complète et l'Apprentissage des Langues Assisté par Ordinateur (ALAO). De plus, la correction entièrement automatique des textes est utilisée dans le cadre du prétraitement ou du post-traitement dans plusieurs tâches du TAL, telle que la Récupération d'Informations (RI), la Reconnaissance Optique de Caractères (ROC), la reconnaissance automatique de la parole et la Traduction Automatique (TA). Malgré sa popularité, la correction automatique des erreurs syntaxiques est encore loin d'être résolue complètement.

Dans le cadre des systèmes de la correction automatique des erreurs en langue arabe, plusieurs tentatives ont été déjà effectuées au niveau d'orthographe [1, 2, 3]. En revanche, en ce qui concerne le niveau syntaxique, les travaux de recherche sont extrêmement limités et moins avancés. À notre connaissance, il n'y a que le système de Shaalan, qui a développé un vérificateur de grammaire “Arabic GramCheck” [4]. Cependant, il ne traite que les erreurs syntaxiques courantes. En plus, il n’a connu aucune nouvelle version depuis son apparition en 2005. Dernièrement un nouveau projet en cours de développement avec l'utilisation de Deep Learning (DL) et les réseaux de neurones [5] a été lancé. L'élément essentiel de ce projet est la création d'un corpus annoté qui permet d'afficher toutes les erreurs syntaxiques. Or, le corpus disponible ne désigne pas ces types d'erreurs, ce qui représente un très grand challenge pour ce projet.

En comparaison avec les autres langues latines, particulièrement l’anglais (e.g. Grammarly[1], Ginger[2], etc.) et le français (e.g. Cordial[3]) , de tels systèmes existent déjà, et des recherches sont toujours renouvelées.

[1] https://www.grammarly.com/ Dernier accès le 01/09/2019
[2] https://www.gingersoftware.com/ Dernier accès le 01/09/2019
[3] https://www.cordial.fr/ Dernier accès le 01/09/2019

Dans cette thèse, nous traitons des erreurs syntaxiques en langue arabe. Celle-ci est l'une des langues les plus couramment utilisées. C'est la cinquième langue au monde qui compte un plus grand nombre de locuteurs natifs [6]. Il existe des outils et des ressources, tels que des tagueurs et des analyseurs syntaxiques, développés pour la langue arabe aussi pour toutes les autres langues. Ces moyens permettent aux chercheurs en traitement automatique des langues de développer des algorithmes avancés et de produire des résultats comparables.

OBJECTIFS ET HYPOTHÈSES

L'objectif principal de cette thèse est de développer des algorithmes et des méthodes efficaces pour la correction automatique des erreurs syntaxiques produites par les apprenants de l'arabe ou les erreurs courantes.

Pour aboutir à cet objectif, nous avons défini les tâches suivantes :

1. Recueillir des exemples d'erreurs syntaxiques.
2. Choisir la métrique d'évaluation la plus adéquate.
3. Proposer des nouvelles méthodes et approches capables de corriger un nombre important d'erreurs
4. Construire un système de correction automatique des erreurs syntaxiques efficace.

Toutes ces tâches sont complétées et décrites dans la thèse.

CONTRIBUTIONS DE LA THÈSE

Cette thèse apporte deux contributions au domaine de la correction automatique des erreurs syntaxiques aux niveaux théorique et technique.

Tout d'abord, nous présentons un état de l'art des méthodes et des approches historiques et des technologies dans le domaine de traitement automatique des langues naturelles et spécialement de l'arabe. Après, nous montrons les approches de détection et de correction automatiques des erreurs. Cette partie de la thèse peut également servir aux lecteurs non familiarisés avec ce domaine.

Ensuite, nous présentons en détail nos nouvelles approches. La première pour la détection et la correction des erreurs syntaxiques basée sur la génération automatique des phrases correctes. La deuxième approche pour la correction automatique des erreurs de désinence casuelle basée sur l'analyse syntaxique.

ORGANISATION DE LA THÈSE

La présente thèse s'articule autour de cinq chapitres, il est divisé en deux parties, la première partie présente le contexte général de la recherche ainsi qu'un état de l'art sur la détection et la correction d'erreurs syntaxiques. Elle se compose de trois chapitres.

Le premier chapitre introduit le contexte général, l'historique et les travaux de traitement automatique des langues. Nous y abordons les différents niveaux de traitement de la langue naturelle.

Le deuxième chapitre est consacré au traitement automatique de la langue arabe, il définit les particularités et les complexités de la langue arabe aux niveaux morphologique et syntaxique et montre le niveau élevé d'ambiguïté vis-à-vis d'autres langues. Il présente également quelques applications de TALA.

Le troisième chapitre expose un état de l'art pour la tâche de correction automatique des erreurs syntaxique.

la deuxième partie de cette thèse est dédiée aux contributions scientifiques. Elle est consacrée à la présentation des méthodes de correction syntaxique basées sur les nouvelles approches que nous avons proposées et cette partie contient deux chapitres.

Le quatrième chapitre décrit une approche novatrice de la correction automatique des erreurs syntaxiques dans les textes arabes basée sur la génération automatique de phrases correctes.

Dans le dernier chapitre nous présentons une méthode de la correction des erreurs syntaxiques de désinence casuelle en Arabe basée sur l'analyse syntaxique.

NOTES DE PUBLICATIONS ET COMMUNICATIONS

PUBLICATIONS :

- Revue JKSUCIS, Elsevier (ScienceDirect)
  C. Moukrim, A. Tragha, E. H. Benlahmer et al., An innovative approach to autocorrecting grammatical errors in Arabic texts, Journal of King Saud University–Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.02.005 (Février 2019).
  Indexing: **Web Of Science, Scopus.**
- Revue IJACSA, The Science and Information Organization
  Chouaib MOUKRIM, Abderrahim TRAGHA and El Habib BENLAHMER, "The Correction of the Grammatical Case Endings Errors in Arabic Language"

International Journal of Advanced Computer Science and Applications(IJACSA), 10(11), 2019, http://dx.doi.org/10.14569/IJACSA.2019.0101138 (Novembre 2019).
Indexing: **Web Of Science, Scopus.**

- Revue IJSC, Medwell Journals
  C. Moukrim, A. Tragha, E. H. Benlahmer, The automatic generation of Arabic sentences based on a minimalist approach, International Journal of Soft Computing (Novembre 2018)

COMMUNICATIONS DANS LES CONGRÈS NATIONAUX ET INTERNATIONAUX :

- Chouaib Moukrim, Abderrahim Tragha and El Habib Benlahmer, La détection et la correction des erreurs d'accord en langue Arabe, The Second National Doctoral Symposium On Arabic Language Engineering(JDILA'15), ENSA, Fes, Maroc (11 Octobre 2015)
- Chouaib Moukrim, Tarik Almalki, Abderrahim Tragha and El Habib Benlahmer, De la génération des phrases par l'ontologie à la correction des erreurs syntaxiques, 4ème Journée sur les Technologies d'Information et de Modélisation (TIM'16), FSBM, Casablanca, Maroc (2 Juin 2016)
- Chouaib Moukrim, Abderrahim Tragha and El Habib Benlahmer, Tarik Almalki, Une nouvelle approche pour la correction des erreurs syntaxiques en langue Arabe, Journée de Mathématiques et Applications (JMA'16), FSBM, Casablanca, Maroc (12 Juillet 2016)
- Chouaib Moukrim, Abderrahim Tragha and El Habib Benlahmer, Tarik Almalki, A new approach for correcting syntactic errors in Arabic language, The 6th International Conference in Arabic Language Processing (ICALP'17), ENSA, Fes, Maroc (11, 12 Octobre 2017)
- Chouaib Moukrim, Abderrahim Tragha and El Habib Benlahmer, La correction syntaxique des erreurs de désinence casuelle en langue Arabe, The 4th international conference on Information and Modeling Technologies (TIM'18), FSBM, Casablanca, Maroc (14 Juillet 2018)
- Chouaib Moukrim, Abderrahim Tragha and El Habib Benlahmer, La génération automatique des phrases correctes syntaxiquement basée sur une approche minimaliste, The Third National Doctoral Symposium On Arabic Language Engineering(JDILA'18), FSBM, Casablanca, Maroc (1 Novembre 2018)
- Chouaib Moukrim, Abderrahim Tragha and El Habib Benlahmer, Tarik Almalki, The correction of the grammatical case endings errors in Arabic language, The International Conference on Modern Intelligent Systems Concepts (MISC'2018), Mohammed V University-Faculty of Sciences, Rabat, Morocco (12,13 Décembre 2018)

Première partie

Contexte de recherche

Chapitre 1 : Le Traitement Automatique des Langues (TAL).

Chapitre 2 : Le Traitement Automatique de la Langue Arabe (TALA).

Chapitre 3 : Etat de l'art.

Chapitre 1

Le Traitement Automatique des Langues (TAL)

1. Introduction au TAL

Le traitement automatique des langues (TAL), ou traitement automatique des langues naturelles (TALN) est l'un des domaines les plus actifs en informatique et il a obtenu un meilleur éclairage au cours des dernières années. C'est un domaine permettant de gérer l'interaction entre un ordinateur et un utilisateur.

La langue naturelle est une langue humaine, par opposition aux langages informatiques. La différence entre eux réside dans la présence d'ambiguïtés. Aucun langage informatique bien conçu n'est ambigu. En revanche, toutes les langues naturelles connues présentent la propriété de l'ambiguïté. L'ambiguïté se produit lorsqu'une entrée peut avoir plus d'une interprétation. L'ambiguïté existe à tous les niveaux du langage humain.

Il n'existe pas une seule définition convenue qui satisferait tout le monde [7], mais certains aspects, qui feraient partie de la définition de toute personne bien informée. Nous proposons :

Définition : le traitement automatique du langage naturel est un ensemble de techniques informatiques motivées par la théorie afin d'analyser et représenter des textes naturels à un ou plusieurs niveaux d'analyse linguistique dans l'objective est de réaliser un traitement du langage à la ressemblance humaine pour une série de tâches ou d'applications.

Plusieurs éléments de cette définition peuvent être détaillés. Premièrement, la notion imprécise de « *ensemble de techniques informatiques* » est nécessaire car il existe des méthodes ou des techniques multiples qui permettent de choisir la réalisation d'un type particulier d'analyse linguistique.

Les « *textes naturels* » peuvent être de toutes les langues, modes, genres, etc. Les textes peuvent être oraux ou écrits. La seule exigence est qu'ils soient dans un langage utilisé par les êtres humains pour communiquer les uns avec les autres. En outre, le texte en cours d'analyse ne doit pas être spécifiquement construit aux fins de l'analyse, mais plutôt être rassemblé à partir d'un usage réel.

La notion de « *niveaux d'analyse linguistique* » (à expliquer plus en détail dans la section 3) fait référence au fait qu'il existe de nombreux niveaux interdépendants pour comprendre et extraire le sens d'un texte ou de mots prononcés. En effet, pour comprendre les langues naturelles, il est important de les distinguer. Nous pensons que les humains utilisent normalement tous ces niveaux, car chaque niveau transmet différents types de sens. Mais divers systèmes de TAL utilisent

plusieurs niveaux, ou combinaisons de niveaux d'analyse linguistique, et cela se voit dans les différences entre les diverses applications de TAL.

« *Le traitement du langage à la ressemblance humaine* » révèle que le TAL est considéré comme une discipline au sein de l'Intelligence Artificielle (IA). Et tandis que la pleine lignée du TAL dépend d'un certain nombre d'autres disciplines, le TAL aspire à une performance semblable à celle de l'Homme, il convient de le considérer comme une discipline de l'IA.

« *Pour une série de tâches ou des applications* » indique que le TAL n'est généralement pas considéré comme un objectif en soi, sauf peut-être pour les chercheurs en intelligence artificielle. Pour d'autres, le TAL est le moyen d'accomplir une tâche particulière. Par conséquent, vous disposez de systèmes de récupération d'informations utilisant le TAL, ainsi que la traduction automatique, les questions-réponses, etc.

Comme indiqué ci-dessus, l'objectif du TAL est « *la réalisation d'un traitement du langage à la ressemblance humaine* ». Le choix du mot « *traitement* » est très délibéré et ne devrait pas être remplacé par « *compréhension* ». Bien que le domaine du TAL ait été initialement appelé compréhension automatique des langues (CAL) dans les premiers jours de l'IA, il est bien convenu aujourd'hui que, même si l'objectif du TAL est un la compréhension, cet objectif n'a pas encore été atteint.

Il y a des objectifs plus pratiques pour le TAL, dont beaucoup sont liés à plusieurs application particulière pour laquelle elle est utilisée. Par exemple, un système de récupération d'informations basé sur le TAL a pour objectif de fournir des informations complètes et plus précises en réponse aux besoins réels des utilisateurs. L'objectif du système TAL ici est de représenter le sens et l'intention véritables de la requête de l'utilisateur, qui peuvent être exprimés aussi naturellement dans le langage courant comme s'ils parlaient à un bibliothécaire de référence. En outre, le contenu des documents recherchés sera représenté à tous les niveaux de signification, ce qui permettra de trouver une correspondance réelle entre le besoin et la réponse, quelle que soit la manière dont ils sont exprimés sous leur forme superficielle.

Comme dans la plupart des disciplines modernes, la lignée du TAL est en effet mixte et a encore aujourd'hui une forte emphase sur différents groupes dont les origines sont davantage influencées par l'une ou l'autre de ces disciplines. Les principaux contributeurs à la discipline et à la pratique du TAL sont les suivants :

- La linguistique - se concentre sur les modèles de langage formels et structurels et la découverte des universaux de la langue - en fait, le domaine du TAL était à l'origine appelé la linguistique computationnelle "Computational Linguistics".

- L'informatique - se préoccupe de développer des représentations internes de données et un traitement efficace de ces structures.
- La psychologie cognitive - considère l'utilisation du langage comme une fenêtre sur les processus cognitifs humains et a pour objectif de modéliser l'utilisation du langage de manière plausible sur le plan psychologique.

Bien que l'ensemble du domaine soit appelé traitement automatique du langage, il existe en fait deux domaines distincts. Le premier de ceux-ci fait référence à l'analyse du langage dans le but de produire une représentation significative, tandis que le second fait référence à la production de la langue à partir d'une représentation (la génération de la langue). Bien qu'une grande partie de la théorie et de la technique soient partagées par ces deux divisions, la génération des langues naturelles nécessite également une capacité de planification. C'est-à-dire que le système de génération nécessite un plan ou un modèle de l'objectif de l'interaction afin de décider de ce que le système doit générer à chaque point de l'interaction. Nous nous concentrerons sur l'analyse du langage naturel, car elle s'intéresse particulièrement à la bibliothéconomie et à la science de l'information.

2. Bref historique du traitement automatique des langues

La recherche dans le domaine du traitement automatique des langues (TAL) a été en cours depuis plusieurs décennies remontant à la fin des années 1940. La traduction automatique a été la première application informatique liée au TAL qui faisait partie, à l'époque, de l'IA.

Tandis que Weaver et Booth [8] ont lancé en 1946 l'un des premiers projets de traduction automatique basés sur l'expertise en matière de violation des codes de l'ennemi pendant la Seconde Guerre mondiale, il a été généralement admis que c'est le mémorandum de Weaver [9] qui a fait connaître l'idée de la traduction automatique et a inspiré de nombreux projets, il a déclaré :

> *" I have a text in front of me which is written in Russian but I am going to pretend that it is really written in English and that it has been coded in some strange symbols. All I need to do is strip off the code in order to retrieve the information contained in the text."*

Weaver a suggéré d'utiliser des idées tirées de la cryptographie et de la théorie de l'information pour la traduction automatique. Les travaux de recherche ont commencé dans diverses institutions de recherche aux États-Unis en quelques années.

Les premiers travaux de traduction automatique ont adopté une vision simpliste que la seule différence entre les langues réside dans leurs vocabulaires et les ordres de mots autorisés. Les systèmes développés dans cette perspective ont simplement utilisé la recherche dans le dictionnaire pour trouver les mots appropriés à la traduction et ont réorganisé les mots après la traduction pour les adapter à l'ordre des mots dans la langue cible, sans tenir compte de l'ambiguïté lexicale inhérente à la langue naturelle. Cela a produit des résultats médiocres. L'échec apparent a fait prendre conscience aux chercheurs que la tâche était beaucoup plus difficile que prévu et qu'ils avaient besoin d'une théorie du langage plus adéquate et une vision traduite en anglais par "Computational Linguistics". Cependant, ce n'est qu'en 1957 que Chomsky [10] a publié les structures syntaxiques "Syntactic Structures", introduisant l'idée de la grammaire générative, le champ a-t-il permis de mieux comprendre si la linguistique traditionnelle pouvait aider la traduction automatique.

Au cours de cette période, d'autres domaines d'application du TAL ont commencé à émerger, tels que la reconnaissance vocale. La communauté de traitement automatique de langue et la communauté de parole ont ensuite été divisées en deux camps, la communauté de traitement de langue dominée par la perspective théorique de la grammaire générative et refuse les méthodes statistiques, et la communauté de parole dominée par la théorie des informations statistiques et refuse la linguistique théorique.

En raison de l'évolution de la théorie syntaxique du langage et des algorithmes d'analyse syntaxique, il était exagéré dans les années cinquante de croire que les systèmes de traduction entièrement automatiques de haute qualité seraient en mesure de produire des résultats impossibles à distinguer de ceux des traducteurs humains, et de tels systèmes devraient être opérationnels dans quelques années. Ce n'était pas irréaliste car les connaissances linguistiques et les systèmes informatiques disponibles à l'époque étaient également impossibles en principe.

Les insuffisances des systèmes existants, et peut-être accompagnées d'un enthousiasme excessif, ont conduit à l'ALPAC "Automatic Language Processing Advisory Committee" à conclure que la traduction automatique n'était pas immédiatement réalisable et recommandait qu'elle ne soit pas financée [11]. Cela a eu pour effet de mettre un terme au développement des applications de traduction automatique ainsi que la plupart des travaux du TAL, au moins aux États-Unis.

Bien que les travaux sur le TAL aient considérablement diminué au cours des années qui ont suivi le rapport ALPAC, des progrès importants ont été enregistrés, à la fois sur le plan théorique et sur celui de la construction de systèmes prototypes. Les travaux théoriques menés à la fin des années 1960 et au début des années 1970 étaient centrés sur la question de savoir comment représenter la sémantique et élaborer des solutions informatiques, que les théories existantes en matière de grammaire n'étaient pas en mesure de produire. En 1963, Miller et Chomsky [12] ont introduit le modèle transformationnel de compétence linguistique. Cependant, les grammaires génératives transformationnelles étaient trop orientées vers la syntaxe afin de ne pas tenir compte des préoccupations sémantiques. De plus, ils ne se prêtaient pas facilement à une implémentation informatique. En réaction aux théories de Chomsky et d'autres chercheurs génératives transformationnels (linguistes qui suivent l'approche générative transformationnelle), notamment la grammaire des cas de Fillmore [13] qui a mis en évidence le fait que la structure syntaxique peut être prédite par les participants sémantiques, les réseaux sémantiques de Quillian et Raphael [14], et la théorie de dépendance conceptuelle de Schank [15], ont été développés pour expliquer les anomalies syntaxiques et fournir des représentations sémantiques. Les réseaux de transition augmentés de Woods [16] ont étendu le pouvoir de la grammaire syntagmatique en incorporant des mécanismes issus de langages de programmation tels que LISP. Parmi les autres

formalismes de représentation figuraient la sémantique des préférences de Wilks [17] et la grammaire fonctionnelle de Kay [18].

Parallèlement au développement théorique, de nombreux systèmes prototypes ont été développés. Le programme ELIZA de Weizenbaum [19] a été conçu pour reproduire une conversation entre un psychologue et un patient, simplement en reformulant la plupart des affirmations de l'utilisateur en question, et en les lui posant. SHRDLU de Winograd [20] a simulé un robot manipulant des blocs sur une table. Malgré ses limites, il a montré que la compréhension du langage naturel était effectivement possible pour l'ordinateur. PARRY de Colby [21] a tenté d'incarner une théorie de la paranoïa dans un système, et considérait ELIZA comme un agent clinique potentiel qui pourrait, dans un cadre de temps partagé, gérer plusieurs centaines de patients par heure de manière autonome. LUNAR a été développé par Woods [22] comme système d'interface avec une base de données contenant des informations sur les échantillons de roches lunaires à l'aide d'un analyseur syntaxique de réseau de transition augmenté ("Augmented Transition Network"—ATN) et d'une sémantique procédurale, ce système a fait l'objet d'une démonstration informelle lors de la deuxième conférence scientifique annuelle lunaire en 1971.

À la fin des années 1970, l'attention s'est tournée vers les questions sémantiques, les phénomènes discursifs et les objectifs et plans de communication. Grosz [23] a analysé les tâches orientées vers les dialogues et a proposé une théorie de partitionner le discours en unités en fonction de ses conclusions sur la relation entre la structure d'une tâche et la structure de la tâche orientée vers le dialogue. Mann et Thompson [24] ont développé la théorie de la structure rhétorique, attribuant la structure hiérarchique au discours. D'autres chercheurs ont également apporté des contributions significatives, notamment Hobbs et Rosenschein [25] ainsi que Reichman [26].

Cette période a également vu un travail considérable sur la génération du langage naturel. Le planificateur de discours de McKeown "TEXT" [27] et le générateur de réponse de McDonald "MUMBLE generator" [28], ont utilisé des « schémas rhétoriques » pour produire des descriptions déclaratives sous forme de courts textes, généralement des paragraphes. La capacité de TEXT pour générer des réponses cohérentes en ligne a été considérée comme une réalisation majeure.

Au début des années 1980, motivées par la disponibilité de ressources informatiques, la prise de conscience croissante au sein de chaque communauté conduit à des solutions isolées aux problèmes du TAL et une poussée générale vers des applications fonctionnant avec la langue dans un contexte large et réel, les chercheurs ont commencé à réexaminer des approches non symboliques qui avaient perdu leur popularité dans les premiers jours. À la fin des années 1980, des approches symboliques issues de l'intelligence artificielle avaient été utilisées pour

résoudre de nombreux problèmes importants liés au TAL et les approches statistiques se sont révélées complémentaires à bien des égards les approches symboliques.

Au cours des vingt dernières années du millénaire, le secteur a connu une croissance rapide. Cela peut être attribué à :

- Augmentation de la disponibilité de grandes quantités de textes électroniques (corpus).
- La disponibilité des ordinateurs avec une vitesse et une plus grande performance.
- L'avènement de l'Internet.

Les approches statistiques ont permis de résoudre de nombreux problèmes génériques en linguistique computationnelle telle que l'identification partielle de la parole, la désambiguïsation du sens des mots, etc. et sont devenus la norme dans le TAL. Les chercheurs en TAL développent actuellement des systèmes de nouvelle génération qui traitent assez bien le texte général et tiennent compte d'une bonne partie de la variabilité et de l'ambiguïté de la langue.

3. Les niveaux de TAL

Tout système qui tente de comprendre une langue naturelle doit avoir une connaissance approfondie de la structure de la langue - son vocabulaire, comment les mots se combinent pour former des phrases, le sens des mots et la manière dont ils se combinent pour donner une signification aux phrases.

La méthode la plus explicative pour présenter ce qui se passe réellement dans un système de traitement automatique des langues consiste à utiliser les niveaux linguistiques. Les recherches psycholinguistiques suggèrent que le traitement de la langue est beaucoup plus dynamique, car les niveaux peuvent interagir dans divers ordres. Certes, nous utilisons fréquemment les informations que nous tirons ce qui est généralement considéré comme un niveau de traitement supérieur pour faciliter le niveau d'analyse inférieur.

3.1. LE NIVEAU PHONOLOGIQUE

La phonologie est une partie de la linguistique qui fait référence à la disposition systématique des sons à l'intérieur et à travers les mots. Le terme phonologie vient du grec ancien et se compose de deux morceaux phono- qui signifie voix ou son, et le suffixe –logie désigne une science ou étude scientifique d'un sujet. La phonologie inclut l'utilisation sémantique du son pour coder le sens de tout langage humain. En effet, ce niveau gère trois types de règles : les règles phonétiques, les règles phonémiques et les règles prosodiques.

1. les règles phonétiques définissent les contraintes sur la combinaison des sons dans les mots.
2. les règles phonémiques définissent les variations de prononciation lorsque les mots sont prononcés ensemble.
3. les règles prosodiques sont utilisées pour définir la fluctuation du stress et de l'intonation au travers d'une phrase.

Dans les systèmes de TAL qui traitent la parole, il existe des recherches en traitement automatique de la parole, comme les systèmes de transcription automatiques qui permettent de produire des corpus oraux annotés [29]. Les ondes sonores sont analysées et codées en un signal numérique pour une interprétation selon diverses règles ou par comparaison avec le modèle d'un langage particulier utilisé.

3.2.LE NIVEAU MORPHOLOGIQUE

Ce niveau traite la structure interne des mots, composés de morphèmes - unité linguistique minimale (i.e. non décomposable) porteuse de sens. Par exemple, le mot « préinscription » peut être analysé morphologiquement en trois morphèmes distincts : le préfixe « pré » dont le sens est « devant, avant », la racine « inscrip » et le suffixe « tion ». Puisque la signification de chaque morphème reste la même pour tous les mots, l'Homme peut décomposer un mot inconnu en ses morphèmes constitutifs afin de comprendre sa signification. De la même manière, un système TAL peut reconnaître le sens transmis par chaque morphème afin de gagner et de représenter un sens. Par exemple, en anglais l'ajout du suffixe—"ed" à un verbe indique généralement que l'action du verbe a eu lieu dans le passé.

3.3.LE NIVEAU LEXICAL

A ce niveau, les humains, ainsi que les systèmes de TAL, interprètent la signification des mots individuels. Plusieurs types de traitement contribuent à la compréhension au niveau des mots - le premier d'entre eux est l'affectation d'un seul étiquette POS "Part-Of-Speech" à chaque mot. Dans ce traitement, les mots pouvant fonctionner comme plusieurs POS se voient attribuer l'étiquette la plus probable, en fonction du contexte dans lequel ils se produisent.

De plus, au niveau lexical, les mots qui n'ont qu'un seul sens ou une seule signification possible, peuvent être remplacés par une représentation sémantique de ce sens. La nature de la représentation varie en fonction de la théorie sémantique utilisée dans le système du TAL. Le niveau lexical peut exiger un lexique, et l'approche particulière adoptée par un tel système déterminera si un lexique sera utilisé, ainsi que la nature et l'étendue des informations codées dans le lexique.

Les lexiques peuvent être assez simples, avec seulement les mots et leurs POS, ou peuvent être de plus en plus complexes et contenir des informations sur la classe sémantique du mot, ses arguments et les limites sémantiques de ces arguments, le sens dans la représentation sémantique utilisée dans un système particulier, et même du champ sémantique dans lequel chaque sens d'un mot polysémique est utilisé.

3.4.LE NIVEAU SYNTAXIQUE

La syntaxe est une branche clé de la linguistique. Elle se concentre sur l'étude scientifique de la structure de la phrase en tant qu'unité indépendante. L'ordre des mots, les relations de dépendance entre ces mots et, dans certaines langues, les relations d'accord ainsi que le marquage des cas figurent parmi les points qui ont retenu l'attention de la plupart des chercheurs. L'objectif final de la syntaxe est de produire une description formelle aux régularités sous-jacentes vis-à-vis de

l'organisation de la phrase et de déterminer les principes qui régissent les relations de combinaison et de dépendance des mots et des séquences de mots dans la phrase.

Ce niveau se focalise sur l'analyse des mots d'une phrase afin de découvrir la structure grammaticale de la phrase. Cela nécessite à la fois une grammaire et un analyseur. L'output de ce niveau de traitement est une représentation de la phrase qui révèle les relations de dépendance structurelles entre les mots. Il existe différentes grammaires qui peuvent être utilisées et qui, à leur tour, auront une incidence sur le choix d'un analyseur. La syntaxe transmet un sens dans la plupart des langues parce que l'ordre et la dépendance contribuent au sens. Par exemple les deux phrases : « Le chien a poursuivi le chat. » et « Le chat a poursuivi le chien.», ne diffèrent que par la syntaxe, pourtant transmettent des significations très différentes. Dans ce qui suit, nous présentons en détail les différentes approches d'analyse syntaxique.

3.4.1. L'ANALYSE SYNTAXIQUE

L'analyse syntaxique comprend un ensemble de processus mentaux qui comblent le fossé entre les processus sémantiques au niveau des mots et au niveau du discours. Ces processus servent à identifier ou à récupérer des dépendances entre les mots d'une phrase. À l'aide des dépendances structurelles, des informations conceptuelles fournies par le contenu des mots régissant comment attribuer les rôles thématiques. À titre d'exemple, le contenu des mots embarrasser, infirmière et médecin. Ces mots ne suffisent pas pour permettre de dire qui fait quoi, à qui (comment, quand et où). Les indices syntaxiques et les processus d'analyse syntaxique fournissent ces informations nécessaires. En effet, la signification attribuée à un énoncé donné dépend de plusieurs facteurs, aucune théorie de la compréhension d'une langue ne peut être complète sans prendre en compte la syntaxe (ou la grammaire) et l'analyse syntaxique. Comparée à la morphologie, la syntaxe se distingue par le fait qu'elle se concentre sur les relations entre les mots, tandis que la morphologie se concentre sur les variations de formes de mots. Notez que certains courants linguistiques considèrent que le morphème est l'unité de base de la syntaxe. Cela nous amène à considérer que les processus de création de mots et de construction de phrases sont de même nature. Donc, dans ce cas, nous nous référons à la morphosyntaxe.

Comme la linguistique est une discipline descriptive et non normative, il convient de commencer par une clarification du concept descriptif. Dans le domaine du TAL, la grammaire n'est pas un ensemble de règles qu'un locuteur doit suivre, car sa production est considérée comme étant bien formée (grammaire normative), mais plutôt une description des phénomènes syntaxiques utilisés par toute communauté linguistique à un moment donné. Selon Chomsky [30], la grammaire est un appareil capable d'effectuer des jugements de grammaticalité, c'est-à-dire de classer les séquences d'unités d'entrée (chaînes lexicales) en deux groupes : les chaînes bien formées et mal formées (voir figure 1).

Figure 1 Le rôle de la grammaire selon Chomsky

L'analyse syntaxique se fait par rapport à une grammaire, essentiellement un ensemble de règles indiquant quelles combinaisons des étiquettes POS génèrent des structures considérées syntaxiquement bien formées. Par conséquent, la phrase suivante :

- J'ai vu une voiture

Cette phrase est jugée syntaxiquement bien formé, puisque « sujet + verbe + déterminant + nom » forme une phrase valide, par contre :

- J'ai vu voiture une

Cette phrase est jugée non grammaticale, puisque le déterminant doit précéder le nom pour constituer avec lui un groupe nominal.

La syntaxe met l'accent sur la forme linguistique de la phrase sans accorder une importance primordiale au sens, objet d'étude pour la sémantique. Si nous voulons simplifier, nous pouvons dire que la syntaxe se concentre sur les relations entre les signes linguistiques, alors que la sémantique se concentre sur les relations entre ces signes et ceux qu'ils signifient, ainsi que sur le sens général de la phrase qui sera produit par la syntaxe. Cependant, les limites des deux disciplines ne sont pas très claires. En effet, il est largement admis que la complémentarité de ces deux sources de connaissance est indispensable pour pouvoir comprendre correctement une phrase, en particulier dans le cas d'ambiguïtés syntaxiques ou d'anomalies sémantiques. L'analyse sémantique implique l'identification de différents types de mots ou d'expressions, entre autres, la reconnaissance d'un mot ou d'une expression en tant que nom propre, ainsi que l'identification du rôle qu'ils jouent dans la phrase, tel que le sujet ou l'objet. Les rôles sémantiques peuvent différer des rôles syntaxiques, par exemple les deux phrases (1) et (2) ont des structures syntaxiques différentes mais elles ont la même signification :

« La Cour suprême a condamné Samsung. » (1)
Et
« Samsung a été reconnu coupable par la Cour suprême. » (2)

3.4.2. LES APPROCHES LINGUISTIQUE DE L'ANALYSE SYNTAXIQUE

Selon les linguistes, l'analyse syntaxique consiste à décomposer les phrases en unités syntaxiques et à identifier de relations de dépendance. Elle conduit

souvent à une représentation graphique sous la forme de parenthèses d'après l'approche originale proposée par le linguiste américain Charles Francis Hockett [31]. Généralement, cela prend la forme d'un diagramme en arbre.

Il existe plusieurs formalismes pour représenter l'analyse syntaxique d'un texte. Par contre, la quasi-totalité de la littérature porte sur deux formats de représentation syntaxique, à savoir la représentation syntagmatique et la représentation par dépendance :

i. Le modèle syntagmatique

L'analyse syntaxique par constituants divise tout d'abord la phrase en plusieurs groupes de mots appelés syntagmes. Un syntagme est un intermédiaire entre l'ensemble global qui est la phrase et la division unitaire composée des mots. Le principe de ce concept est de pouvoir subdiviser logiquement la phrase en groupes de plus en plus petits. Le syntagme est en fait un ensemble de mots, ou de plus petits syntagmes, ayant un rôle linguistique commun dans la phrase. La décomposition de la phrase en ces constituants immédiats constitue le point de départ dans l'élaboration d'une grammaire syntagmatique—étudié par les linguistes comme Noam Chomsky [32]. Pour pouvoir dire si une séquence de la chaîne parlée est constituant de la phrase (un syntagme) ou non, on recourt, suivant la position épistémologique qu'on défend, à des critères formelles (La séquence peut-elle être permutée ou substituée ?) ou substantielles (Le syntagme est-t-il en relation syntaxique avec d'autres syntagmes de la phrase ?). Dans tous les cas, il ne suffit pas de décomposer la phrase en ses constituants, puis de nouveau chaque constituant en ses propres constituants, et ainsi de suite jusqu'aux unités minimales de la phrase, il faut encore attribuer chacun de ces constituants ou syntagmes à «sa» catégorie.

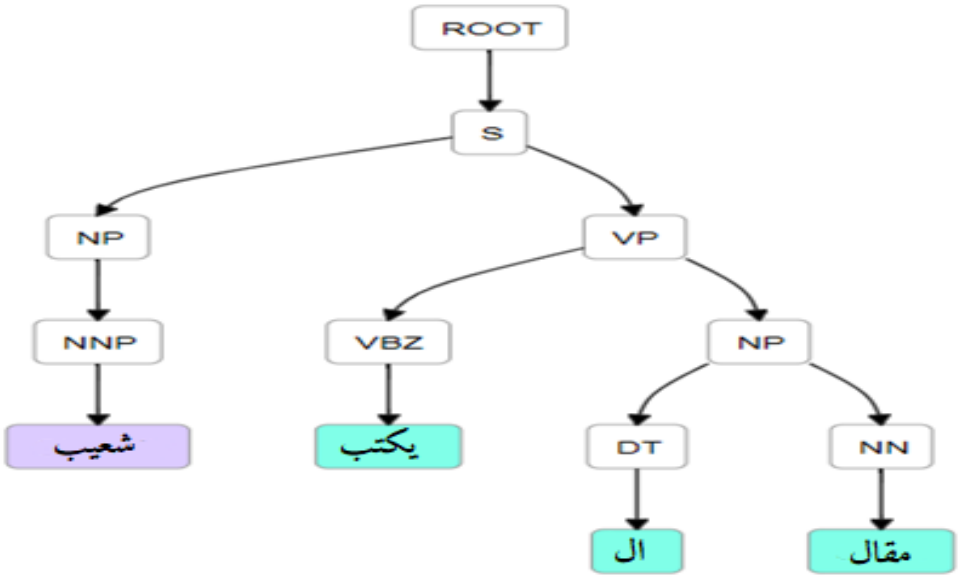

Figure 2 Syntaxe de la phrase "يكتب شعيب المقال" représentée par constituants

La structure syntagmatique de la phrase peut être rendue par un système de règles de réécriture. Ces règles, appelées règles syntagmatiques, opèrent sur des catégories ; elles permettent ensuite d'engendrer des arbres ou des parenthétisassions étiquetées. L'exemple suivant présente un ensemble des règles de réécriture qui permettent d'engendrer l'arbre syntagmatique et les parenthèses avec leurs étiquettes :

```
(ROOT
  (S
    (NP (NNP شعيب))
    (VP (VBZ يكتب)
      (NP (DT ال) (NN مقال)))))
```

ii. Le Modèle de dépendance

La dépendance comme connexion structurale s'établit entre deux mots, plus généralement, entre deux positions syntaxiques, de sorte que chaque mot est dépendant syntaxiquement d'un autre mot de la phrase, sauf le verbe principal de la phrase qui sera désigné comme la racine de la phrase. Chaque mot, à l'exception de la racine, a exactement une et une seule tête dont laquelle il est dépendant. La structure d'arbres peut encore être utilisée pour représenter visuellement l'analyse syntaxique par dépendance.

Les mots de la phrase représentent les nœuds de l'arbre et les relations de dépendance représentent les arcs reliant ces nœuds. Dans la Figure 4, nous représentons visuellement la structure par dépendance de la même phrase représentée précédemment :

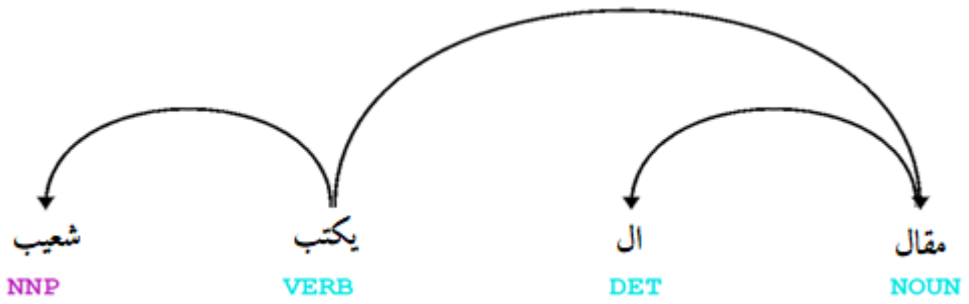

Figure 3 Syntaxe de la phrase "يكتب شعيب المقال" représentée par dépendance

Le linguiste français Lucien Tesnière a publié les principes du modèle basé sur les structures de dépendance [33], son origine est attribuée aux travaux des grammairiens arabes au 8ième siècles (tel que Sibawaih, décédé en 798) [34]. Contrairement à l'analyse syntagmatique, il ne reconnaît pas la prédication comme une relation syntaxique distincte de la subordination (de la dépendance). Pour lui, le sujet est subordonné du verbe exactement comme les compléments d'objet.

3.5. LE NIVEAU SÉMANTIQUE

C'est le niveau auquel la plupart des gens pensent que le sens est déterminé. Cependant, comme on peut le voir dans les définitions des niveaux ci-dessus, l'objective de tous les niveaux est de contribuer à la signification. Le traitement sémantique détermine les significations possibles d'une phrase en se concentrant sur les interactions entre les significations au niveau des mots dans la phrase. Ce niveau de traitement peut inclure la désambiguïsation sémantique de mots ayant plusieurs sens ; de manière analogue à la façon dont la désambiguïsation syntaxique des mots pouvant fonctionner comme plusieurs POS est accomplie au niveau syntaxique. La désambiguïsation sémantique permet à un seul sens de mots polysémiques d'être sélectionné et inclus dans la représentation sémantique de la phrase. Une large gamme de méthodes peut être mise en œuvre pour réaliser la désambiguïsation, certaines nécessitant des informations sur la fréquence à laquelle chaque sens se produit dans un corpus ou dans un usage général, certaines nécessitant une prise en compte du contexte local, et d'autres qui utilisent des connaissances pragmatiques du domaine du document.

3.6. LE NIVEAU PRAGMATIQUE

Ce niveau concerne l'utilisation intentionnelle de la langue dans des situations, ainsi que le contexte au-delà du contenu du texte pour la compréhension. Le but est d'expliquer comment une signification supplémentaire est lue dans les textes sans y être réellement encodée. Cela nécessite beaucoup de connaissances, y compris la compréhension des intentions, des plans et des objectifs. Certaines applications de TAL peuvent utiliser des bases de connaissances et des modules d'inférence.

Les systèmes de TAL actuels ont tendance à implémenter des modules pour atteindre principalement les niveaux de traitement les plus bas, pour plusieurs raisons. En premier lieu, ils n'exigent pas l'interprétation aux niveaux supérieurs. En second lieu, les niveaux inférieurs ont fait l'objet de recherches et d'une mise en œuvre plus approfondies. En dernier lieu, les niveaux inférieurs traitent de petites unités d'analyse, e.g. les morphèmes, les mots et les phrases, qui sont régis par des règles, par opposition aux niveaux plus élevés de traitement du langage qui traitent de textes et de connaissances, et qui ne sont régies que par des régularités.

4. Les applications du TAL

Les techniques du traitement automatique des langues peuvent être appliquées à une multitude domaines tels que la traduction automatique, l'extraction de l'information, le résumé automatique des textes, les systèmes de Question-Réponse etc. Les applications les plus fréquentes utilisant le TAL sont les suivantes :

4.1. TRADUCTION AUTOMATIQUE

La traduction automatique (TA) est l'une des applications les plus anciennes du TAL. Différents niveaux ont été utilisés pour la traduction automatique, allant des approches de niveau phonologique aux approches de niveau pragmatique, afin de la rendre aussi humaine que possible. Les travaux de recherche pour la TA ont commencé dès les années 1950 [35] et ont connu un développement rapide depuis les années 1990, en raison du développement des capacités de stockage et de performance des ordinateurs, ainsi que les corpus disponibles multilingues et bilingues.

L'objectif principal de la TA est de permettre aux utilisateurs d'évaluer le contenu dans des langues autres que les langues dans lesquelles ils parlent couramment. D'un point de vue formel, cela signifie que l'objectif de la TA est de transférer la sémantique d'un texte d'une langue d'entrée à une langue de sortie. Bien que les systèmes de la TA soient désormais populaires sur le Web, ils génèrent toujours un grand nombre de traductions incorrectes. Selon des recherches récentes, les principales erreurs responsables sont les erreurs de réorganisation (40%) et l'ambiguïté lexicale et syntaxique (30%) [36]. Par conséquent, la suppression de ces obstacles est un défi majeur pour les systèmes de traduction modernes. Un grand nombre d'approches de TA ont été développées au fil des années et pourraient potentiellement servir de remède.

Les systèmes de TA ont commencé par utiliser des méthodologies basées sur les règles [37]. Cependant, ces systèmes présentent un inconvénient majeur car ils reposent sur des règles élaborées manuellement, ce qui rend le développement de nouveaux modules de traduction pour différentes langues plus difficiles [38]. Les approches basées sur la statistique ont été développées pour traiter le problème d'évolutivité des méthodes basées sur les règles [39], pourtant certains problèmes déjà résolus ont été réapparus. La majorité de ces problèmes sont liés à l'ambiguïté, y compris les variations syntaxiques et sémantiques.

Les deux approches ont été combinées afin de résoudre les inconvénients de ces deux familles d'approches. Cette combinaison de méthodes s'appelle TA

hybride. Bien que les approches hybrides aient donné de bons résultats, elles souffrent encore de certains problèmes [40].

De nos jours, un nouveau paradigme de TA est apparu, appelé (“Neural Machine Translation”—NMT), qui repose sur des algorithmes de réseau neuronal. Cette approche a obtenu des résultats impressionnants [41]. Malgré tout, NMT reste une approche statistique avec quelques inconvénients sémantiques. Une solution possible aux problèmes repose sur l'utilisation du Web sémantique, qui est devenue un paradigme pour rendre la sémantique du contenu explicite afin qu'il puisse être utilisé par des machines. Les connaissances sémantiques explicites peuvent donner aux systèmes de TA le pouvoir de fournir des traductions avec une qualité sensiblement meilleure tout en restant évolutif [42].

4.2. RECHERCHE D'INFORMATIONS

La recherche d'informations (RI) “Information Retrieval”, est axée sur la récupération de documents potentiellement pertinents à partir d'une base de données volumineuses basée sur une requête utilisateur. Son objectif est de permettre à l’utilisateur un accès facile à l’information pertinente répondant à son besoin. Les systèmes de recherche d'informations classent les documents en fonction de leur estimation de l'utilité d'un document pour une requête utilisateur, et la plupart parmi eux attribuent un score numérique à chaque document et trient les documents en fonction de ce score. La nécessité de méthodes efficaces de recherche automatique d'informations a pris une importance croissante en raison de l'énorme explosion de la quantité de données non structurées.

Avec les connaissances ci-dessus, nous constatons que la recherche d'informations est très convenable pour répondre aux requêtes, car il s’agit d’une application informatique dont la tâche principale est de traiter les textes dans lesquels le TAL joue un rôle essentiel. Il s’agit de construire un système capable d'accepter une requête en langage naturel et renvoyer une liste de documents classés en fonction de leur pertinence estimée par rapport aux besoins d'informations du client.

Les systèmes de recherche d'informations ont principalement pris en compte deux perceptions différentes. La première perception comprend l’étude de diverses méthodes de recherche d’informations à l’aide de nombreuses expériences réalisées dans des environnements contrôlés [43]. Le deuxième point de vue concernant la recherche sur les relations internationales met l’accent sur l’aspect utilisateur et ses activités cognitives ainsi que sur la façon dont ces activités sont représentées par rapport aux systèmes de réplication intégrés. Le comportement cognitif des utilisateurs dans les différents processus des systèmes de recherche d'informations devrait être considéré comme un paradigme cognitif [44]. En outre, cette perception est qualifiée de « tradition orientée utilisateur » selon laquelle les informations sont

traduites comme une interprétation humaine des sources de données lors de l'interaction avec le système pour récupérer des informations.

4.3. EXTRACTION D'INFORMATIONS

L'extraction d'informations (EI) consiste à récupérer certains éléments d'information clés (telles que le nom de la personne, l'emplacement ou d'autres informations) à partir de gros volumes de texte en langue naturelle en les traitant automatiquement [45]. L'extraction d'informations exige la capacité d'effectuer une reconnaissance ou une normalisation d'entité nommée, mais ajoute la possibilité de caractériser les relations entre ces entités nommées.

Tout depuis le début, les chercheurs ont adopté des approches purement basées sur les règles, des approches purement basées sur l'apprentissage automatique, ou hybride. Le monde commercial accorde une grande importance à l'interprétabilité de l'approche basée sur les règles, ce qui facilite l'adoption, la compréhension, le débogage et la maintenance des programmes d'extraction d'informations. En outre, les programmes qui utilisent l'approche basée sur les règles sont appréciés pour permettre d'incorporer facilement des connaissances de domaines essentiels afin de cibler des problèmes métier spécifique [46]. Toutefois, les recherches universitaires les plus récentes dans ce domaine partent de l'hypothèse que l'apprentissage automatique est la meilleure approche pour résoudre les problèmes d'extraction d'informations [47].

4.4. QUESTION ANSWERING (QA)

La question-réponse est un domaine de recherche en pleine expansion qui regroupe des recherches issues de la recherche d'informations, l'extraction d'informations, et les outils de traitement automatique des langues. L'objective est de fournir automatiquement une réponse à une question posée par un humain en langue naturelle. À la différence des systèmes de recherche d'informations qui présentent aux utilisateurs un ensemble des documents relatifs à leurs questions, mais n'indiquent pas exactement les réponses correctes. La réponse aux questions en tant que tâche peut être divisée en trois sous-tâches distinctes, à savoir : l'analyse des questions, la récupération de documents et l'extraction de réponses [48].

Les systèmes de Question-Réponse constituent une bonne solution pour interroger des informations non structurées et structurées. C'est pourquoi un grand nombre de systèmes de Question-Réponse ont été développés dans différentes langues. Certaines langues, comme le latin, en particulier l'anglais, sont mieux servies. En raison de la popularité, de l'importance et des fonctionnalités de cette langue, des dizaines de systèmes sont disponibles depuis 1960, à l'instar du système BASEBAL [49]. Tandis que, d'autres langues n'ont pas atteint le niveau requis, telles que l'arabe et les langues sémitiques en général. Cela pourrait être lié aux

caractéristiques linguistiques de chaque langue ainsi qu'à la maturité des recherches scientifiques dans ces pays.

4.5. RÉSUMÉ AUTOMATIQUE DU TEXTE

Un résumé est un moyen de condenser une grande quantité d'informations (provenant d'un document ou de plusieurs documents) en une forme abrégée pour un utilisateur en sélectionnant les informations importantes et en éliminant les informations non importantes et redondantes.

Avec la quantité d'informations textuelles présentes sur le ("World Wide Web"—WWW), le domaine de résumé automatique du texte devient très important. Le processus consiste à condenser un texte source afin d'en produire une version plus courte en préservant son contenu d'information. Les outils de résumé automatique peuvent aider les utilisateurs à comprendre rapidement les principaux concepts des sources d'information. Ces applications utilisent les niveaux les plus élevés du traitement automatique des langues, tels que le niveau sémantique et pragmatique, pour fournir une représentation plus courte du document original plus grand sans perdre aucune information importante.

Tandis que l'approche pour résoudre le problème du résumé peut varier d'une étude à l'autre, les approches peuvent être classées en quatre catégories, à savoir des approches statistiques, d'apprentissage automatique, basée sur la sémantique et basée sur l'intelligence distribuée "swarm intelligence".

L'approche statistique traite certaines caractéristiques du texte afin d'identifier les parties importantes d'un document. L'objective est de sélectionner des phrases en fonction de leurs caractéristiques statistiques comme la fréquence des mots, la position de la phrase et les mots-clés, etc. et ne prend pas en compte l'aspect sémantique des phrases [50]. Il existe plusieurs travaux de recherche qui utilisent cette approche nous citons le travail d'Edmunson [51] qui a mis en place un système d'extraction des phrases, qui utilise non seulement la fréquence des mots mais aussi quelques traits (les phrases de repère, les mots du titre, l'emplacement des phrases). Les méthodes statistiques sont simples à mettre en œuvre. Cependant, elles n'engagent pas le sens des phrases et des mots, ce qui peut produire un résumé de faible qualité.

L'approche basée sur l'apprentissage automatique consiste à utiliser un ensemble de données d'apprentissage pour former le système de résumé, qui est modélisé comme un problème de classification. Les phrases sont classées en deux groupes : phrases résumées et phrases non résumées. La probabilité de choisir une phrase pour un résumé est estimée en fonction du document de formation et des résumés extractifs [52].

L'approche basée sur la sémantique consiste à identifier la relation entre les mots et les phrases à l'aide d'un dictionnaire, d'un étiqueteur (“Part-Of-Speech”—POS), d'un analyseur syntaxique et d'une sélection de phrases significatives pour générer un résumé [53]. Les méthodes basées sur la sémantique sont utiles car elles prennent en compte la signification de chaque phrase et mot, ce qui rendent un résumé cohérent et significatif, mais l'utilisation de ces techniques prend du temps et demande plus d'efforts que les autres techniques.

La dernière approche est basée sur l'intelligence distribuée “swarm intelligence”, l'intelligence distribuée (appelée aussi intelligence en essaim) est une branche de l'intelligence artificielle, qui repose sur la simulation informatique pour copier les interactions de la créature entre elles et avec leur environnement afin de résoudre un problème d'optimisation. Différents algorithmes basés sur le comportement d'intelligence distribuée ont été introduits pour les problèmes d'optimisation, de robotique, de routage, de data mining, de mise en cluster, etc. En ce qui concerne le résumé automatique du texte, les algorithmes les plus distingués, il y a “particle swarm optimization”, “cuckoo” et “bacterial foraging optimization”[54].

5. Les outils et techniques du TAL

L'étiquetage linguistique du texte se déroule généralement en couches successives. Les textes sont divisés en paragraphes, les paragraphes en phrases et les phrases en mots individuels. Les mots d'une phrase sont ensuite étiquetés par des étiqueteurs ("Part Of Speech"—POS) et d'autres caractéristiques, avant l'analyse de la phrase (soumise à une analyse grammaticale). Ainsi, les analyseurs syntaxiques s'appuient généralement sur des délimiteurs, des tokenizers, des stemmers et des étiqueteurs POS de phrases. Mais les applications n'exigent pas une suite complète de ces outils. Par exemple, tous les moteurs de recherche effectuent une étape de tokenisation, mais n'effectuent pas tous une partie d'étiquetage POS.

Dans cette section, nous décrivons quelques outils du TAL auxquels nous ferons référence tout au long de cette thèse. La plupart de ces outils sont potentiellement utiles pour toutes les tâches énumérées ci-dessus, beaucoup d'entre eux sont librement disponibles à des fins de recherche ; d'autres sont disponibles en tant que produits commerciaux.

5.1. LES OUTILS DE SEGMENTATION DES PHRASES

Afin d'analyser les phrases d'un texte, nous devons déterminer la portée de ces phrases et identifier leurs constituants.

Détecter correctement les limites d'une phrase n'est pas une tâche facile, car les signes de ponctuation marquant la fin d'une phrase sont souvent ambigus. Par exemple, le point peut désigner un point décimal, une abréviation, etc. Afin de lever l'ambiguïté des signes de ponctuation, les délimiteurs de phrases s'appuient souvent sur des expressions régulières ou des règles d'exception. D'autres outils de segmentation de phrases reposent sur des techniques empiriques et sont formés sur un corpus segmenté manuellement.

Les approches de la segmentation des phrases se divisent en trois catégories :

1. Approches basées sur les règles, utilisant des heuristiques manuelles et des listes d'abréviations
2. Approches d'apprentissage automatique supervisées, formées dans une configuration supervisée, c'est-à-dire avec un texte annoté.
3. Approches d'apprentissage automatique non supervisées qui nécessitent que des corpus bruts et non annotés.

Le tableau suivant présente quelques outils de segmentation du texte en phrases :

Tableau 1 outils de segmentation du texte en langue anglaise

Outils de segmentation	Approches	Site de téléchargement
CoreNLP	basée sur les règles	http://nlp.stanford.edu/software/corenlp.shtml
GATE	basée sur les règles	http://gate.ac.uk
MxTerminator	apprentissage supervisée	ftp://ftp.cis.upenn.edu/pub/adwait/jmx/
OpenNLP	apprentissage supervisée	http://opennlp.apache.org/
Splitta	apprentissage supervisée	http://code.google.com/p/splitta

5.2. LES OUTILS DE TOKENISATION

Les tokenizers (également appelés segmenteurs de phrases en mots) segmentent un flux de caractères en unités significatives appelées tokens. La tokenisation est en quelque sorte un prétraitement ; une identification des unités linguistiques de base à traiter. En effet, sans ces unités de base clairement séparées, il est impossible d'effectuer une analyse morphologique ou une génération du texte.

En principe, la création de tokens semble assez simple. Toutefois, les approches de tokenisation peuvent être appropriées pour certaines applications, mais elles peuvent conduire à des inexactitudes. Par exemple, elles ne prennent pas en compte les signes de ponctuation, tels que les points, les virgules et les traits d'union :

- « casse-tête » est-il composé d'un ou deux tokens ?
- « pomme de terre » les espaces n'indiquent pas réellement une rupture entre les tokens.

Les méthodes de tokenisation reposent généralement sur des règles, des automates finis, des modèles statistiques et des lexiques pour identifier les abréviations ou les mots à multi-token. De nombreux outils sont disponibles, entre autres "NLTK Word Tokenize", "Nlpdotnet Tokenizer", "TextBlob Word Tokenize", "Pattern Word Tokenize". Etc.

5.3. LES STEMMERS

En linguistique, le stemming est le processus permettant de réduire les mots infléchis (ou parfois dérivés) à leur forme de base. L'algorithme de stemming consiste à élaborer une normalisation linguistique dans laquelle les formes variantes

d'un mot sont réduites à une forme commune, par exemple « vais », « vas », « va », « allons » et « allez » sont associés à la forme racine « aller ».

Plusieurs algorithmes ont été proposés pour le développement de stemmers. Ils sont principalement divisés selon leur nature en deux approches à savoir, approche basée sur les règles et approche basée sur les statistiques.

L'approche basée sur les règles repose sur des algorithmes qui sont purement basés sur la connaissance morphologique de la langue. Cette approche est beaucoup plus rapide car il ne nécessite aucune étape de prétraitement. Cependant, elle exige des règles pour couvrir toutes les formes morphologiques. Le premier algorithme de ce type a été proposé par Lovins en 1968 [55], et Porter [56] en 1980 qui a proposé un algorithme pour la suppression de suffixe.

L'approche basée sur les statistiques est indépendante de la langue car les données statistiques (obtenues par un corpus) permettent d'acquérir des connaissances sur la morphologie de la langue. Cette approche ne nécessite aucune expertise linguistique, ainsi qu'elle est plus adaptée aux langues qui utilisent principalement les suffixes par nature. Mais elle prend plus de temps, car il faut effectuer une tâche de prétraitement et la taille du corpus est généralement importante.

5.4. LES ONTOLOGIES ET LE WEB SÉMANTIQUE

5.4.1. LES ONTOLOGIES EN PHILOSOPHIE

En philosophie, l'ontologie est la branche la plus fondamentale de la métaphysique.
Le terme ontologie a été inventé au 17ème siècle en parallèle par les philosophes Rudolf Göcke et Jacob Lorhard.

Étymologiquement, (ont-) provient du participe présent du verbe grec "einai" (être) et, ainsi, le mot latin "Ontologia" (ont + logia) peut être traduit par l'étude de l'existence.

5.4.2. LES ONTOLOGIES EN INFORMATIQUE

Une ontologie est un modèle de données qui représente un ensemble de concepts dans un domaine avec les relations entre ces concepts. Les définitions de ces entités comprennent des informations sur leur signification et leurs contraintes sur leur application. Par conséquent, l'ontologie peut être utilisée pour raisonner sur les objets de son domaine.

L'ontologie contient des informations sur les concepts, les relations, les attributs et les individus. Les concepts sont des groupes abstraits ou des types d'objets. Les relations décrivent les relations entre les objets, elles incluent les

relations de spécialisation et la relation de composition. Les attributs décrivent les concepts auxquels ils appartiennent.

Les individus sont des instanciations de concepts, représentant habituellement des objets du monde réel. Ils contiennent des valeurs (instances d'attributs) et sont connectés à des instances de relation. Les langages ontologiques sont des langages formels utilisés pour coder les spécifications ontologiques.

5.4.3. PEUPLEMENT D'ONTOLOGIE

Le peuplement d'ontologie est le processus d'acquisition de connaissances à partir de données non structurées, semi-structurées et structurées et les transformer en instances ontologiques[4].

5.4.4. L'INFÉRENCE

L'inférence (raisonnement) est une méthode d'acquisition ou d'expansion des connaissances. Elle étend la base de connaissances avec des informations supplémentaires, en utilisant les données, les métadonnées et les règles existantes. Avec le développement de OWL-DL[5], un certain nombre de raisonneurs tels que RACER, FaCT++[6] et Pellet ont été développés. Ces outils utilisent l'information et les restrictions ontologiques pour classer, réaliser et vérifier les données sémantiques. La classification construit la hiérarchie complète des classes. La réalisation trouve des types directs de chaque individu et la vérification valide les données.

Outre ces tâches de raisonnement, on peut développer des règles spécifiques au domaine pour déduire des informations plus complexes. SWRL[7] et les règles de Jena[8]sont deux langages de règles largement utilisés. SWRL est la recommandation du W3C comme langage de règles du Web sémantique. Jena Rules est une puissante alternative à SWRL. L'idée est la même dans les deux langages : lorsque les conditions sur le corps de la règle sont satisfaites, les clauses sur la tête sont exécutées.

5.4.5. LA PLATEFORME « PROTÉGÉ »

Protégé [9] [57] est une plateforme très répandue pour développer des ontologies, avec une communauté active de plus de 100 000 personnes. Protégé est une plateforme indépendante qui permet à l'utilisateur de construire et distribuer des ontologies indépendamment du système d'exploitation. Simplement, Protégé

[4] http://semanticweb.org (last visited on 08/08/2019)
[5] Ontology Web Language Description Logics
[6] http://owl.man.ac.uk/factplusplus/ (last visited on 08/08/2019)
[7] http://www.w3.org/Submission/SWRL/ (last visited on 08/08/2019)
[8] http://jena.sourceforge.net/inference/#rules (last visited on 08/08/2019)
[9] http://protege.stanford.edu/ (last visited on 08/08/2019)

fournit une interface graphique à travers laquelle l'utilisateur est capable de développer une ontologie.

La figure suivante montre les concepts de l'ontologie de la syntaxe arabe en utilisant l'interface graphique de Protégé :

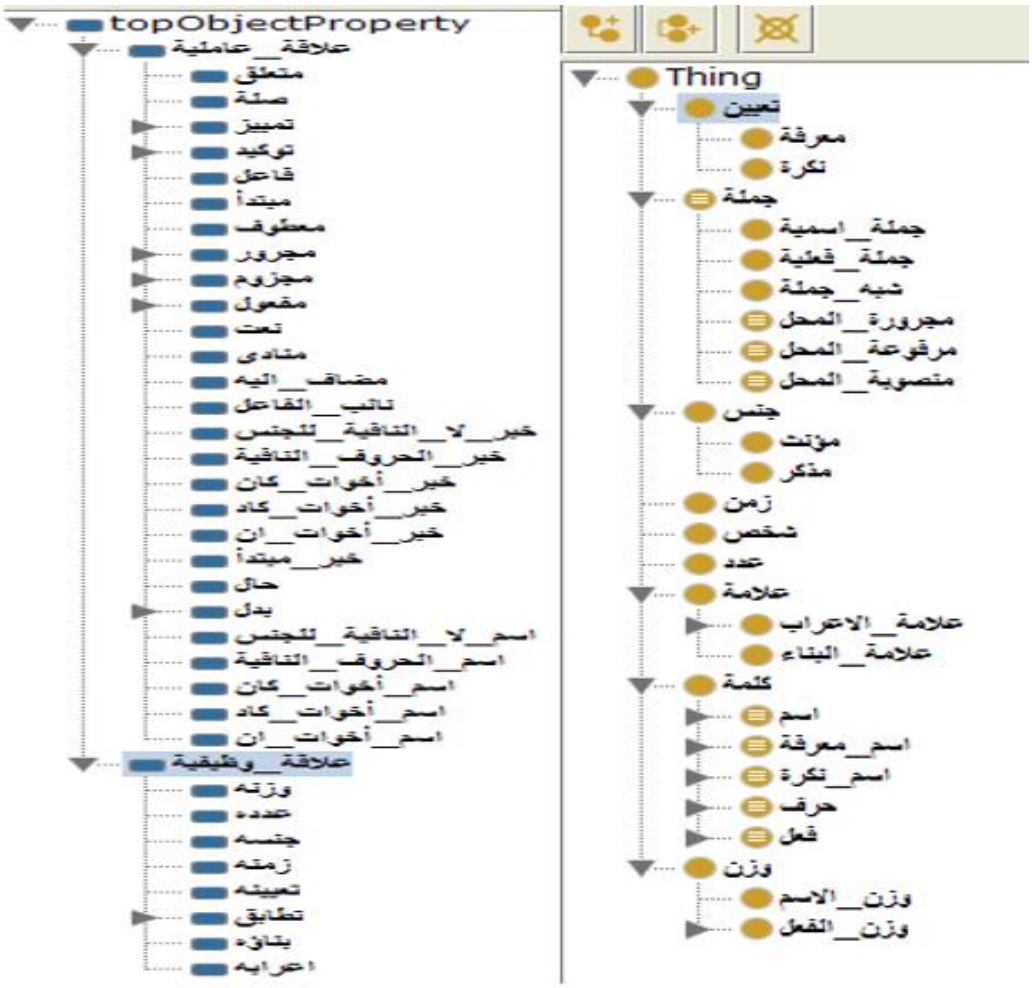

Figure 4 l'outil Protégé

5.4.6. LE WEB SÉMANTIQUE

Le Web sémantique (WS) est défini comme un Web de données. Sa vision est de représenter l'ensemble du Web comme une base de données globalement liée. SW est une extension de l'actuel "World Wide Web", pas un remplacement complet des normes actuelles. Il apporte un certain nombre de nouveaux outils et technologies pour modéliser, annoter, rechercher et intégrer des données.

Le Web sémantique est basé sur deux choses : un ensemble de formats communs pour l'intégration et la combinaison de données provenant de diverses sources, où le Web original est principalement axé sur l'échange de documents. C'est aussi une manière de représenter en utilisant les langues comment les données se rapportent aux objets du monde réel. Cela permet à une personne ou à une

machine, à partir d'une source de données, d'accéder à un ensemble de données qui ne sont pas liés par des sujets. Ainsi, le Web sémantique fournit un cadre commun qui permet aux données d'être partagées et réutilisées par les applications, les entreprises et les communautés.

Tim Berners-Lee et ses collaborateurs ont présenté leur propre vision du Web sémantique par un espace vaste permettant d'échanger des ressources entre les humains et machines en exploitant qualitativement de grands volumes d'informations et de services variés. En effet, le Web actuel est essentiellement syntaxique, en ce sens que son contenu reste difficilement accessible aux traitements automatiques par les machines. Seuls les humains peuvent interpréter ces contenus. Le Web sémantique vise à surmonter cette difficulté. L'idée est que la connaissance soit accessible à la fois aux humains et à la machine, permettant aux programmes de mener des recherches intelligentes en utilisant diverses sources d'information. Ainsi, les données seraient plus facilement accessibles, grâce à la représentation sémantique de leur contenu. Cette représentation sémantique est basée sur des ontologies qui sont des spécifications explicites de conceptualisations. Une conceptualisation rend compte de la signification des termes, dans un domaine donné, elle définit quelles sont les entités (concepts) qui la caractérisent. La spécification explicite signifie que les relations entre les concepts, les propriétés, les fonctions et les axiomes sont explicitement définis.

i. Le but du Web sémantique

Le Web actuel se compose de pages et de documents qui sont connectés entre eux via des hyperliens.

Un utilisateur peut facilement atteindre l'information souhaitée en suivant ces liens. C'est facile pour nous, car le contenu est rendu par le navigateur afin que l'être humain puisse le comprendre et l'interpréter. Cependant, d'un point de vue d'une machine ou d'un agent logiciel, une page Web n'est rien d'autre que du code HTML pur, ce qui ne donne aucune idée de la signification du contenu. Ainsi, les agents automatiques ne peuvent pas parcourir le Web et recueillir des informations aussi facilement que nous. Le web sémantique est proposé pour surmonter ces difficultés.

ii. Langages du web sémantique

Puisque les documents HTML ne sont pas capables de contenir des données sémantiques, des nouvelles normes pour le Web sémantique ont été développées. Nous décrivons quelques-unes des plus importantes, à savoir RDF, RDF-S et OWL, ces trois langages se basent sur XML.

XML : extensible markup language

De nos jours, extensible markup language (XML) est largement utilisé pour représenter l'échange d'informations entre les applications via Internet. XML

fournit un ensemble de sémantiques pour représenter les données et permet au contenu de données d'avoir une structure arbitraire. Il est lisible par l'homme, flexible, facile à utiliser et à créer. Cependant, il ne définit pas la signification des étiquettes. La même balise XML peut avoir différentes significations dans différentes structures de données. D'autre part, différentes balises XML peuvent avoir la même signification. Par exemple, une structure de courrier électronique décrivant XML contient une balise "SendTo" tandis qu'une autre structure de courrier électronique dans XML contient une balise nommée « Destinataire ». Il est évident pour les humains que "SendTo" et « Destinataire » ont la même signification, mais pour les ordinateurs qui ne comprennent pas l'anglais, "SendTo" et « Destinataire » ne signifient pas la même chose.

Pour résoudre le problème ci-dessus, les parties communicantes peuvent soit utiliser le même ensemble de tags XML, soit manipuler manuellement les ambiguïtés sur la signification des balises XML avant que les communications aient lieu. Le tag XML global n'est certainement pas faisable. Un autre langage pour représenter les significations des étiquettes XML est nécessaire de sorte qu'elles puissent être interprétées par les ordinateurs.

RDF : Resource Description Framework

Le World Wide Web Consortium (W3C) a accepté diverses spécifications pour la mise en œuvre du Web sémantique. Ces spécifications sont contenues dans un cadre connu sous le nom de Resource Description Framework (RDF). RDF définit l'URI "Uniform Resource Identifier" pour identifier de façon unique un certain concept ou terme. Pour les URI afin d'éliminer l'ambiguïté RDF ne fait pas référence à Toyota par la chaine de caractères "Toyota". Au lieu de cela, RDF peut utiliser un URI comme : http://www. marques.org/Voitures.owl#Toyota

Une description correspond à un ensemble d'énoncés (ou "statements") au sujet d'une ressource. Un énoncé RDF est aussi appelé triplet car il est composé de trois éléments, sujet-prédicat-objet, où :

1. le sujet représente la ressource décrite, i.e. tout document accessible sur le Web comme les pages HTML, les documents textuels (PDF, Ms Word) ou multimédias (images, vidéo), etc., mais aussi tout objet, abstrait ou non, du monde réel. Les ressources sont nommées en utilisant une URI.
2. le prédicat représente la propriété descriptive, i.e. une caractéristique spécifique, un attribut ou une relation, utilisée pour décrire une ressource.
3. l'objet représente la valeur de cette propriété, soit une valeur littérale, comme un nombre entier ou une chaîne de caractère, soit une autre

ressource accessible par son URI. Par contre, une valeur littérale ne peut en aucun cas être le sujet.

Un triplet peut s'écrire « prédicat (sujet, objet) » ou encore « propriété (sujet, valeur) ». Par exemple, la phrase « Chouaib est né à Casablanca » sera traduite par le triplet :

« né (Chouaib, Casablanca) ».

RDFS : Resource Description Framework Schema

RDF Schema (RDF-S) [58] est un ensemble de ressources RDF qui peuvent être utilisées pour décrire des propriétés d'autres ressources RDF. Contrairement à son nom, RDF-S n'est pas un schéma qui impose des contraintes spécifiques à la structure d'un document, mais fournit des informations sur l'interprétation des déclarations dans un modèle de données RDF.

Exemple :

Tableau 2 Exemple de RDF Schema

Ce document RDF dit que : Toyota est une voiture	**Et grâce à ce document de type RDF-Schema on comprend que la voiture est un véhicule**
<rdf:Description ID="#Toyota "> <rdf:type resource="Voiture" /> </rdf:Description>	<rdf:Description ID="Voiture"> <rdfs:subClassOf rdf:resource="véhicule"/> </rdf:Description>

OWL : Web Ontology Language

Le Web Ontology Language (OWL) [59] est la norme récente pour la spécification ontologique dans le domaine du Web sémantique. Dans la terminologie OWL, les concepts sont appelés classes. Une classe peut être spécifiée comme une sous-classe d'une autre classe, mettant ainsi en œuvre la relation de spécialisation. Les propriétés sont utilisées pour définir le contenu des classes. Propriétés se divisent en deux catégories ; les propriétés d'objet représentent des relations générales entre deux classes, tandis que les propriétés de type de données représentent des attributs. Les instances sont encore appelées individus. OWL s'appuie sur d'autres standards, les types de données XML Schema sont utilisés et la syntaxe RDF / XML est utilisée pour échanger des ontologies OWL.

OWL-Lite : conçu pour représenter des hiérarchies avec des contraintes limitées ; par exemple, OWL-Lite ne permet pas d'exprimer des contraintes de cardinalité autres que 0 ou 1, mais permet l'expression de propriétés de transitivités ou inverses. OWL-Lite permet de définir des concepts d'intersection, mais pas des

concepts d'union. En retour OWL-Lite est toujours décidable et il est facile de mettre en œuvre un moteur d'inférence.

OWL-DL (ainsi appelé en référence à la logique de description) est un surensemble de OWL-Lite qui offre une expressivité maximale tout en maintenant l'exhaustivité et la décidabilité des algorithmes d'inférence. OWL-DL offre la possibilité d'exprimer des concepts d'union, des concepts énumérés, des concepts disjoints et la négation de concepts.

OWL-Full est un surensemble de OWL-DL qui offre la possibilité de récupération de type : un concept peut aussi être un individu ou une propriété et vice-versa. La contrepartie de cette expressivité est la perte de décidabilité : il n'y a aucune garantie qu'un moteur d'inférence fournira une réponse dans un temps fini.

OWL2 est la nouvelle version de OWL. Il propose de nouvelles constructions permettant une plus grande expressivité des contraintes (par exemple la disjonction des propriétés) et facilitant l'écriture des motifs fréquemment rencontrés dans OWL-DL (par exemple un concept d'union de concepts disjointe).

iii. Les langages des requêtes sémantiques

Plusieurs langages de requête ont été développés pour rechercher des documents RDF. RDQL, SeRQL et SPARQL n'en sont que quelques-uns. SPARQL est devenu une recommandation officielle du W3C en 2008, il est actuellement le langage de requête sémantique la plus largement utilisé. Fondamentalement, une requête SPARQL consiste en des conjonctions et des disjonctions de motifs triplets semblables à des triplets de RDF. Malgré sa simplicité, l'utilisation de SPARQL est limitée pour l'utilisateur final. Tout d'abord, la formulation d'une requête demande beaucoup de temps et d'efforts, même pour la requête la plus simple. Deuxièmement, la connaissance du domaine est requise, c'est-à-dire les noms exacts des classes et des propriétés doivent être connus à l'avance. Exemple :

Supposons qu'on a une ontologie qui regroupe tous les concepts et instances liés à la recherche scientifique, voilà la requête SPARQL nécessaire pour extraire tous les articles écrits par l'auteur Moukrim Chouaib :

```
PREFIX
sr:
<http://www.semanticweb.org/administrateur/ontologies/2014/1
1/ScientificResearch#>
SELECT ?uri ?uri
WHERE
     {
       ?uri sr:was_written_by  sr:Chouaib_Moukrim
      }
```

6. Conclusion

Dans ce chapitre, nous avons présenté le contexte général de notre thèse. En effet, nous avons dressé un aperçu sur le traitement automatique des langues (TAL). D'abord, nous avons défini quelques concepts de base. Ensuite, nous avons exposé un bref historique, les niveaux, les applications et les outils du TAL.

Dans le chapitre suivant, nous allons présenter le traitement automatique de la langue arabe (TALA). Nous montrerons aussi quelques ressources et systèmes existants ainsi que les particularités et les complexités de la langue arabe.

Chapitre 2

Les applications de traitement automatique de la langue Arabe

1. Introduction au traitement automatique de la langue arabe

Le traitement automatique de la langue arabe (TALA) a attiré de nombreux chercheurs après des recherches importantes sur la langue anglaise et sur d'autres langues. De nombreux laboratoires ont été créés pour traiter la langue arabe. TALA a récemment fait l'objet d'une attention accrue et plusieurs applications ont été développées, notamment la catégorisation de texte, la détection de spam de pages Web et l'analyse des sentiments. Cependant, développer des outils TALA nécessite des efforts supplémentaires en raison de trois difficultés principales : l'irrégularité de l'ordre des mots dans la construction des phrases, le problème d'agglutination et l'absence des signes diacritiques qui représentent les voyelles dans la plupart des textes.

L'arabe est une langue afro-asiatique qui s'est développée au Moyen-Orient. Plus de 290 millions d'individus parlent l'arabe à travers le monde [60]. La langue arabe a été largement diffusée après l'émergence de l'islam, même si elle existait des siècles avant la religion. En tant que religion universelle, l'Islam a transmis la langue arabe à ses disciples, estimés à près de deux milliards de personnes [61].

Historiquement, l'Arabe est classée en Arabe classique (AC), qui est utilisée comme langue maternelle des peuples arabes depuis 600 ans après JC. Elle est associée à l'Islam et au Coran. Cependant, au fil des siècles, la langue a évolué et a été simplifiée pour créer ce que l'on appelle l'Arabe moderne standard (AMS). La terminologie et les caractéristiques linguistiques de AMS diffèrent de celles de AC, mais la structure des mots et des phrases sont restées. En outre, chaque région possède un dialecte de l'Arabe parlée dans la communauté (entre amis et familles).

De nombreuses caractéristiques rendent la langue arabe distinctive. Tel que, la lecture et l'écriture en arabe se déplacent de droite à gauche; elle se compose de 28 caractères; les majuscules et les minuscules ne sont pas distinguées, comme le Chinois, le Japonais et le Coréen; les nombres sont divisés en pluriel, dual et singulier, avec deux genres—féminin et masculin; elle est composée de plusieurs mots formés à partir de racines, et plusieurs mots-racines sont composés de trois lettres; les phrases commencent par des verbes, suivis par des sujets, et se terminent par des objets pour le prédicat; l'arabe tolère la suppression des pronoms sujets comme l'italien et le chinois.

Les sections suivantes présentent les particularités la langue arabe et un état de l'art sur les travaux qui se sont intéressé au TALA.

2. Les particularités et les complexités de la langue arabe

L'arabe peut être considéré plus complexe que l'anglais ou le français. Il ne possède pas de voyelles ; les signes diacritiques sont placés au-dessus ou au-dessous des lettres. Ces signes diacritiques sont abandonnés dans la plupart des textes sauf le Coran, les Hadiths et les manuels des élèves ; les lecteurs sont censés comprendre les signes diacritiques absents en fonction de leur connaissance de la langue. Cette caractéristique induit à la fois des ambiguïtés structurelles et lexicales dans les textes arabes, car ces divers signes diacritiques peuvent avoir des significations différentes.

2.1. LES PARTICULARITÉS AU NIVEAU MORPHOLOGIQUE :

La tokénisation et l'analyse morphologique reposent principalement sur l'identification des mots et la racine des mots. Néanmoins, la flexion rend difficile la réalisation du stemming. Même si de nombreux efforts ont été déployés à cet égard, certaines améliorations sont encore nécessaires. De plus, l'arabe comme toutes les langues sémitiques (amharique, araméen, maltais et hébreu moderne) se caractérise par l'utilisation de certains schèmes (modèles formateurs de mots) permettant d'obtenir des mots à partir de racines abstraites, représentant des notions sémantiques générales ou des significations précises, ainsi qu'elle est dotée d'une morphologie puissante et d'un ordre des mots flexible. Nous pouvons choisir le mot sur lequel nous voulons insister et le mettre en tête de la phrase.

Les lettres majuscules n'existent pas en arabe, ce qui est une caractéristique majeure de la reconnaissance des noms dans les contextes du TAL, en particulier la reconnaissance d'entité nommée présentée à la section 5.5. L'arabe utilise également des inflexions spécifiques ; généralement, un terme peut être défini comme une combinaison de préfixes (qui peuvent être des articles, des prépositions ou des conjonctions), un lemme et un suffixe (qui sont des objets ou une anaphore personnelle/possessive). Certaines particules peuvent porter des préfixes et suffixes, ce qui complique la segmentation.

2.1.1. LES PRÉFIXES

Les préfixes sont représentés par un morphème correspondant à une seule lettre en début de mot, qui indique, entre autres, la personne de la conjugaison des verbes au présent. Les préfixes ne se combinent pas entre eux.

2.1.2. LES SUFFIXES

Les suffixes en arabe, sont essentiellement utilisés pour des terminaisons des conjugaisons verbales, ainsi que les marques du pluriel et du féminin pour les noms. Ils ne se combinent pas entre eux. La taille des suffixes varie entre 1 et 6 caractères.

2.1.3. LES PROCLITIQUES

Au contraire des préfixes et des suffixes, les proclitiques se combinent entre eux pour donner plus d'informations sur le mot arabe (traits sémantiques, coordination, détermination...). Par exemple, dans le cas des verbes, les proclitiques dépendent exclusivement de l'aspect verbal. Ils prennent donc tous les pronoms et par conséquent ils sont compatibles avec tous les préfixes pris par l'aspect. Dans le cas des noms et des déverbaux, le proclitique dépend du mode et du cas de déclinaison.

À l'écrit, il n'est pas toujours facile de faire la différence entre un proclitique et un caractère appartenant à la racine de certains mots. Par exemple le caractère "س" dans le mot "سرق" « il a volé » est un caractère de la racine, par contre dans le mot "سيخرج" « il va sortir » c'est un proclitique qui marque le futur.

2.1.4. LES ENCLITIQUES

Comme les proclitiques, les enclitiques se combinent entre eux pour donner une post-base composée. Ils s'attachent toujours à la fin du mot pour produire des pronoms suffixes qui s'attachent au verbe, au nom et à la préposition.

2.1.5. EXEMPLE

Le mot "فعرفناهم" (en français : « et nous les avons connus », est le résultat de concaténation du proclitique "ف" « et » : conjonction de coordination, le suffixe "نا" pour le présent masculin pluriel, l'enclitique "هم" (pour le pronom de possession masculin pluriel) et le reste du mot "عرف" indiquant le stem. Le tableau 3 présente le mot "فعرفناهم".

Tableau 3 exemple d'agglutination du mot "فعرفناهم"

enclitique	suffixe	stem	proclitique
هم	نا	عرف	ف
les	nous	avons connus	et

Les chercheurs font de gros efforts pour prétraiter la langue arabe en tenant compte de ses caractéristiques antérieures, qui rendent le prétraitement des textes arabes très différent de celui des autres langues.

De manière générale, l'analyse morphologique pour langue arabe est difficile parce que sa structure morphologique comprend également un système prédominant de clitiques. Ce sont des morphèmes grammaticalement indépendants, mais morphologiquement dépendants d'un autre mot ou d'une autre phrase. Un mot peut être associé à divers proclitiques et enclitiques.

2.2.LES PARTICULARITÉS AU NIVEAU SYNTAXIQUE

En ce qui concerne la syntaxe, la phrase arabe est longue avec une syntaxe complexe. Historiquement, les grammairiens arabes voulaient poser les bases de règles de grammaire empêchant la lecture incorrecte du Coran. L'automatisation du processus qui permet à l'ordinateur d'analyser les phrases arabes est vraiment un problème difficile du point de vue informatique. La grammaire arabe distingue deux types de phrases : verbale et nominale. Les phrases verbales commencent généralement par un verbe et comportent au moins un verbe "فعل" et un sujet "فاعل".

Le sujet ainsi que l'objet peuvent être indiqués par la conjugaison du verbe et ne pas être écrits séparément. Par exemple, le verbe conjugué "راسلتك" « je vous ai écrit » a un sujet et un pronom suffixe d'objet qui y est attaché.

L'anaphore en arabe a accentué l'ambiguïté de la langue, parce que dans certains cas, comme les systèmes de traduction automatique n'identifient pas l'antécédent correct en raison de l'ambiguïté de cet antécédent. Des connaissances externes sont nécessaires pour corriger l'antécédent. De plus, les constituants des phrases en arabe peuvent être permutés sans affecter la structure ni le sens, ce qui ajoute une ambiguïté syntaxique et sémantique supplémentaire et nécessite une analyse plus profonde. Néanmoins, l'accord en arabe est total ou partiel et est sensible aux effets de l'ordre des mots. La langue arabe diffère des autres langues en raison de sa structure complexe et ambiguë que le système informatique doit traiter à chaque niveau linguistique.

2.3.L'AMBIGUÏTÉ DANS LE TALA

Les nombreux niveaux d'ambiguïté posent un défi important aux chercheurs qui développent des systèmes de TAL pour l'arabe. En effet, un seul mot peut présenter plusieurs analyses produites par les analyseurs morphologiques. Le SYSTRAN, qui développe des systèmes de traduction automatique depuis plus de 40 ans, il a été estimé que le nombre moyen d'ambiguïtés d'un token dans la plupart des langues est de 2,3%, alors que dans AMS, il atteint 19,2%. Bien que l'ambiguïté soit principalement due à l'absence de voyelles, les chercheurs de SYSTRAN ont

constaté que l'ambiguïté en arabe est présente à tous les niveaux [62]. L'arabe possède une grammaire et une morphologie complexes, une grande ambiguïté dans le sens des mots et une ambiguïté supplémentaire due au système d'écriture qui omet généralement les signes diacritiques (par exemple, voyelles courtes, doublage de consonnes, marques d'inflexion).

3. Quelques systèmes TALA

3.1. LES SYSTÈMES D'ANALYSE MORPHOLOGIQUE

L'analyse morphologique de l'Arabe depuis longtemps est au centre des recherches sur le TAL en raison de la complexité et la richesse de la morphologie arabe. Il y a certaines prétentions qui sont généralement attendues d'un système d'analyse morphologique pour n'importe quelle langue. Celles-ci comprennent, la couverture de la langue d'intérêt en matière de couverture lexicale (grande échelle) et de couverture des phénomènes morphologiques et orthographiques (robustesse); les formes de surface sont mappées à partir d'un niveau de représentation profond qui résume autant que possible les caractéristiques morphologiques et orthographiques spécifiques à la langue; la réversibilité totale du système afin qu'il puisse être utilisé comme analyseur; l'utilisabilité dans un large éventail d'application de TAL telles que la traduction automatique ou la récupération d'information; et enfin, la disponibilité pour la communauté des chercheurs. Ces questions sont essentielles à la conception de tout système d'analyse morphologique arabe.

De nombreux analyseurs morphologiques ont été construits pour une large gamme d'application de TAL allant de l'RI à la TA dans divers contextes théoriques linguistiques, nous présentons deux analyseurs morphologiques les plus courants dans le traitement automatique de la langue arabe.

3.1.1. "BUCKWALTER ARABIC MORPHOLOGICAL ANALYZER—BAMA"

L'analyseur morphologique arabe Buckwalter (BAMA) utilise une approche concaténative basée sur le lexique, dans laquelle la morpho-tactique et les règles orthographiques sont intégrées directement dans le lexique lui-même au lieu d'être spécifiées en matière de règles générales qui interagissent pour obtenir le résultat [63]. Le système comprend trois composants ou tables distincts : une table pour les stems, une table pour les préfixes et une table pour les suffixes. Il existe des contraintes sur les préfixes et les suffixes qui peuvent se combiner avec un stem pour former un mot arabe légitime.

L'analyseur BAMA est devenu le système préféré, car il offre pour les développeurs qui ne connaissent pas l'Arabe de traiter des textes arabes non structurés grâce à son schéma de translittération bidirectionnelle, de l'écriture arabe à l'écriture latine.

En revanche, le système Buckwalter a été un pionnier car il a mis en œuvre une approche de la morphologie arabe basée sur les stems. Buckwalter a montré

qu'il est plus simple de considérer le stem plutôt que la racine comme l'unité de base du lexique arabe, mais les utilisateurs du BAMA ont également accès à des informations sur la racine. En plus de la segmentation et le stemming, BAMA fournit des désinences casuelles complètes pour les noms et les verbes. Il n'effectue pas seulement l'analyse contextuelle mais fournit toutes les analyses possibles des mots du texte saisi. Le système MADA va encore plus loin en utilisant un module de désambiguïsation qui détermine l'étiquetage POS correcte dans un contexte spécifique [64].

3.1.2. AL-KHALIL MORPHO SYS 2

AlKhalil Morpho Sys est un analyseur morphosyntaxique de mots arabes standard hors contexte. Le système peut traiter des textes non diacritisés, ainsi que des textes partiellement ou totalement diacritisés. Cet analyseur repose sur une approche de la modélisation d'un très grand ensemble de règles morphologiques arabes et sur l'intégration de ressources linguistiques utiles à l'analyse, telles que la base de données des racines, les shèmes "الأوزان" vocalisés associés aux racines et les tables des proclitiques et des enclitiques. En sortie de l'analyse, le système produit un tableau très informatif contenant principalement la diacritisation du stem, sa catégorie grammaticale, ses racines possibles associées aux motifs correspondants, les proclitiques et les enclitiques [65].

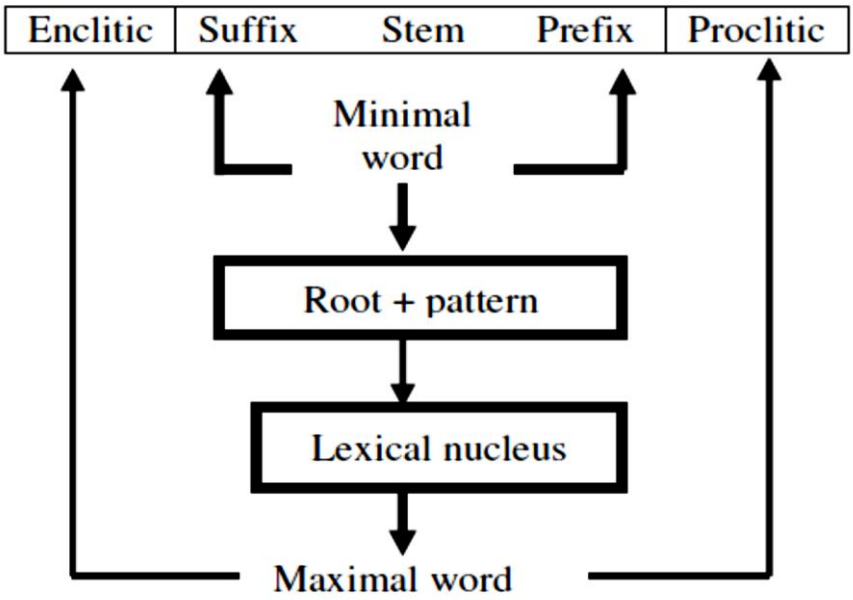

Figure 5 Décomposition d'un mot avec Al-khalil

Cet analyseur est considéré comme le meilleur système morphologique arabe. En fait, Alkhalil a remporté la première position, parmi les 13 systèmes d'analyse morphologique arabe à travers le monde, à un concours organisé par la ligue arabe pour l'éducation la culture et la science (ALECSO).

3.2. LES SYSTÈMES DE CORRECTION ORTHOGRAPHIQUE

La correction orthographique est souvent considérée comme une étape de prétraitement qui traite les erreurs d'orthographe présentes dans les textes. Ces erreurs peuvent réduire l'efficacité des modèles de TAL et peuvent ajouter une marge d'erreur souvent irrécupérable dès la première étape d'un système. Quelquefois, elles peuvent être difficile de les identifier si la forme mal orthographiée est un mot valide, morphologiquement ou sémantiquement incorrect sur le plan contextuel.

La détection et la correction des erreurs d'orthographe sont l'un des problèmes qui ont suscité l'intérêt des chercheurs du TAL dès le début. Dans les années 1964, Damerau [66] a été l'un des premiers chercheurs ayant travaillé sur ce sujet afin de résoudre ces problèmes. Church et Gale [67] ont été les premiers à classer une liste des candidats d'orthographe en fonction des scores de probabilité (en tenant compte des probabilités bi-grammes) sur la base d'un modèle de canal bruyant. Van Delden, Bracewell et Gomez [68] ont utilisé des méthodes d'apprentissage automatique (supervisées et non supervisées). Outre la modélisation en langage n-gramme, la traduction automatique statistique a également été utilisée pour la correction orthographique.

Le problème de correction des erreurs d'orthographe en arabe a été étudié dans un certain nombre d'études comme le travail de Haddad et Yaseen [69] qui ont proposé une approche hybride qui utilise les connaissances morphologiques pour formuler des règles morpho-graphémiques afin de spécifier le processus de reconnaissance et de correction de mots. Hassan, Noeman et Hassan [70] qui ont développé un système indépendant du langage qui utilise des automates à états finis pour proposer des corrections potentielles à une distance d'édition spécifiée du mot mal orthographié. Ces dernières années, la correction orthographique de l'arabe a suscité un regain d'intérêt. Noaman et al. [1] ont utilisé des paires d'erreurs orthographiques et une forme corrigée extraite du QALP "Qatar Arabic Language Bank" pour construire une matrice de confusion des erreurs, puis ont utilisé cette matrice de confusion avec le modèle de canal bruyant pour générer une liste de candidats et sélectionner un candidat approprié pour le mot erroné. La précision globale du système obtenue était de 85%. En revanche, une étude d'Attia et al. [2] ont tenté d'améliorer trois composantes principales : le dictionnaire (ou liste de mots de référence), le modèle d'erreur et le modèle de langage. La façon dont ils ont amélioré le modèle d'erreur consistait à analyser les types des erreurs et à créer un re-classeur basé sur la distance de contrôle qui analysait le niveau de bruit dans différentes sources afin d'améliorer le modèle de langage. En améliorant les trois composants principaux, ils ont atteint un taux de précision de 83%.

3.3. LES SYSTÈMES D'ANALYSE SYNTAXIQUES

De nombreux chercheurs ont travaillait sur les analyseurs syntaxiques, mais nous allons explorer quelques-uns les plus courants pour la langue arabe :

3.3.1. STANFORD PARSER [71]

C'est l'un des analyseurs syntaxiques les plus utilisés de la langue arabe au sein du TALA. Il est écrit en Java et peut-être exécuté via une interface graphique ou à l'aide d'une ligne de commande. Cet analyseur statistique avec le modèle d'apprentissage a connu plusieurs améliorations comme l'intégration d'une représentation du modèle de dépendance. Sa représentation est simple, uniforme et facilement accessible par les utilisateurs non-linguistes. Stanford Parser offre également des outils d'analyse à appliquer en étape de prétraitement comme Stanford Segmenter pour la segmentation des phrases en tokens et Stanford POS-Tagger pour l'étiquetage POS comme le Penn treebank (PTB) [72]. L'analyseur de Stanford et a été adapté à plusieurs langues comme l'anglais [73], le chinois [74], l'allemand [75], le français [76] et l'arabe [77]. Stanford parser arabe a été employé dans divers systèmes tels que, la traduction automatique [78], la question/réponse [79], la modélisation [80]. Les performances de Stanford varient d'une langue à une autre. Pour l'Arabe, il n'a pas généré les meilleurs résultats du fait que Berkeley parser le dépasse de plus que 3,5%. Cette différence est encore plus faible avec d'autres langues [81].

3.3.2. BIKEL PARSER [82]

Cet analyseur offre une amélioration des modèles de Collins [83]. Il vérifie également l'efficacité des étapes de génération appliquées (comme l'élagage de certains nœuds d'arbres d'analyse) et les techniques utilisées (comme la lexicalisation de la grammaire). Il propose l'emploi de nouvelles informations sémantiques pour caractériser les mots outre l'étiquette POS. L'analyseur Bikel est employé pour la génération automatique des analyses préliminaires des phrases du treebank arabe ATB.

3.3.3. ATKS PARSER [84]

C'est un analyseur syntaxique arabe développé par Microsoft dans le laboratoire de technologies de pointe au Caire. Cet analyseur est inclus dans la boîte à outils arabe ("Arabic Toolkit Service"—ATKS). Il convient également de mentionner que l'analyseur ATKS est intégré à plusieurs produits et services Microsoft tels que Windows, Office et Bing et qu'il ne peut être exploité qu'à travers un Web service.

3.4. LES RÉCENTES RESSOURCES ONTOLOGIQUES

3.4.1. WORDNET

WordNet est une grande base de données lexicale d'anglais. Les noms, les verbes, les adjectifs et les adverbes sont regroupés en ensemble de synonymes cognitifs (synsets), chacun exprimant un concept distinct. Les synsets sont liés entre eux par des relations sémantiques et lexicales. Le réseau résultant des mots et des concepts significativement liés peut être navigué avec le navigateur. WordNet est librement et publiquement disponible pour le téléchargement. La structure de WordNet en fait un outil utile pour la linguistique computationnelle et le traitement du langage naturel.

WordNet ressemble superficiellement à un thésaurus, dans le sens qu'il regroupe les mots en fonction de leurs significations. Cependant, il existe des distinctions importantes : WordNet interconnecte non seulement des formes de mots - des chaînes de lettres - mais des sens spécifiques des mots. En conséquence, les mots qui se trouvent à proximité les uns des autres dans le réseau sont sémantiquement désambiguïsés. WordNet étiquette les relations sémantiques entre les mots, tandis que les groupements de mots dans un thésaurus ne suivent aucun motif explicite autre que la similitude de signification.

3.4.2. ARABIC WORDNET

Ce projet comprend la construction d'un WordNet arabe, à la suite du processus de développement de “Princeton WordNet” et d'“Euro WordNet”. Il utilise l'ontologie “Suggested Upper Merged Ontology” comme une interlangue pour relier WordNet arabe aux wordnets développés auparavant.

3.4.3. “SUGGESTED UPPER MERGED ONTOLOGY”—SUMO

SUMO est une ontologie supérieure conçue comme une ontologie de fondation pour une variété de systèmes informatiques de traitement de l'information. SUMO s'occupait à l'origine de concepts de méta-niveau (entités générales qui n'appartiennent pas à un domaine de problème spécifique) et conduirait ainsi naturellement à un schéma de catégorisation des encyclopédies. Il a été considérablement élargi pour inclure une ontologie de niveau intermédiaire et des dizaines d'ontologies de domaine.

3.4.4. THE ARABICONTOLOGY (TRAVAUX DE M JARRAR ET AL.)

Ce projet a débuté en 2010, à l'Université de Birzeit, en Palestine.

ArabicOntology est plus qu'un arabe WordNet. Contrairement à WordNet, ArabicOntology est logiquement et philosophiquement bien fondée, puisqu'elle suit

des principes ontologiques stricts. Mais peut-être utilisé comme un WordNet arabe. Le projet est financé en partie par l'Université Birzeit.

Ils ont identifié l'ensemble des concepts pour chaque mot arabe, et ils ont défini les relations sémantiques entre ces concepts. Ils ont construit manuellement les niveaux supérieurs de l'ontologie à partir des ontologies de niveau supérieur DOLCE et SUMO, et en tenant compte, avec soin, des aspects philosophiques et historiques des concepts arabes.

La figure suivante montre le site web du projet :

Figure 6 Le projet ArabicOntology

4. Conclusion

Nous avons présenté dans ce chapitre les bases du traitement automatique de la langue arabe : origine, particularités morphologique et syntaxique, ambiguïté, etc. Nous avons montré également quelques systèmes et ressources existants.

Dans le chapitre suivant, nous allons présenter un état de l'art sur le domaine de la correction automatique des erreurs syntaxique et les différentes méthodes et approches qu'il utilise.

Chapitre 3

Etat de l'art

1. La correction automatique des erreurs syntaxiques

Le terme « erreur syntaxique » présente des aspects distincts. En effet, il est varié d'une langue à l'autre car chaque langue est constituée de règles de grammaire uniques qui, lorsqu'elles sont violées, peuvent provoquer des erreurs syntaxiques. Les locuteurs non natifs sont influencés par leur langue maternelle et produisent alors des erreurs différentes de celles des locuteurs natifs. Certaines structures linguistiques sont mal utilisées selon le style d'écriture qui peut être formel ou informel.

1.1. LES ERREURS SYNTAXIQUES

Les erreurs syntaxiques ne sont qu'un groupe des erreurs linguistiques. Il existe de nombreuses études sur la classification des erreurs. Certains chercheurs ont classé les erreurs en quatre types à savoir les erreurs d'orthographe, les erreurs de style, les erreurs de grammaire (syntaxe) et les erreurs sémantiques [85]. Z Yuan dans sa thèse de doctorat [86] énonce cinq types d'erreurs à savoir les erreurs lexicales, les erreurs syntaxiques, les erreurs sémantiques, les erreurs de discours et les erreurs pragmatiques. Pour la langue arabe, Qasem Manal [87] (d'après une étude chez les apprenants de langue arabe) a classé les erreurs en six types : les erreurs syntaxiques, les erreurs morphologiques, les erreurs d'orthographe, les erreurs sémantiques et lexicales, les erreurs stylistiques (ou d'utilisation) et les erreurs de ponctuation. Nous nous sommes appuyés sur la dernière classification, qui prend en compte la spécificité de la langue arabe :

- Les erreurs syntaxiques : Toute erreur violant les règles de grammaire est appelée erreur de syntaxe. La grammaire est définie comme un ensemble de règles systématiques à travers lesquelles les mots et les phrases sont assemblés pour donner une signification.
- Les erreurs morphologiques : Les erreurs morphologiques sont celles qui portent sur les signifiants des éléments linguistiques. On distingue ces erreurs en trois catégories fondamentales telles que les erreurs de morphologie lexicale (quand le monème affecté est un lexème) et les erreurs de morphologie grammaticale (quand le monème affecté est un morphème) et les erreurs de syntagmatique, c'est-à-dire celles qui touchent l'accord, le genre, et la position des éléments [88]. Ce sont les désinences ou flexions telles que les marques du genre dans les adjectifs et les marques du temps, du mode, de la personne, du nombre dans les verbes.

- Les erreurs d'orthographe : Elles se définissent par des mots qui n'appartiennent pas à la langue. L'orthographe est un ensemble de règles et d'usage, qui régissent la manière d'écrire les mots ou les sons d'une langue donnée.
- Les erreurs sémantiques et lexicales : Ce sont des erreurs qui ne violent pas les règles de grammaire, mais rendent la phrase insensée ou absurde. Lorsqu'un mot n'est pas détecté comme erreur d'orthographe, et ne rentre pas dans le contexte d'une phrase donnée.
- Les erreurs stylistiques (ou d'utilisation) : Elles ne violent pas la grammaire ou la sémantique d'une langue, mais elles sont en conflit avec la norme sociale et culturelle. Ces erreurs peuvent être dues à l'utilisation de mots peu communs, de structures trop compliquées, de répétition de mots ou de vocabulaire familier.
- Les erreurs de ponctuation : Les signes de ponctuation comme la virgule, le point-virgule, le point final, etc. sont utilisés pour séparer les éléments de phrase. Une ponctuation manquante ou une ponctuation inutile peut modifier le sens de la phrase.

1.2. LA CORRECTION AUTOMATIQUE DES ERREURS

La correction automatique des erreurs est le processus qui vise à détecter et à corriger les erreurs linguistiques dans les textes de manière automatique. Il est souvent divisé en détection et correction. En général la tache de correction automatique distingue quatre phases (figure 7).

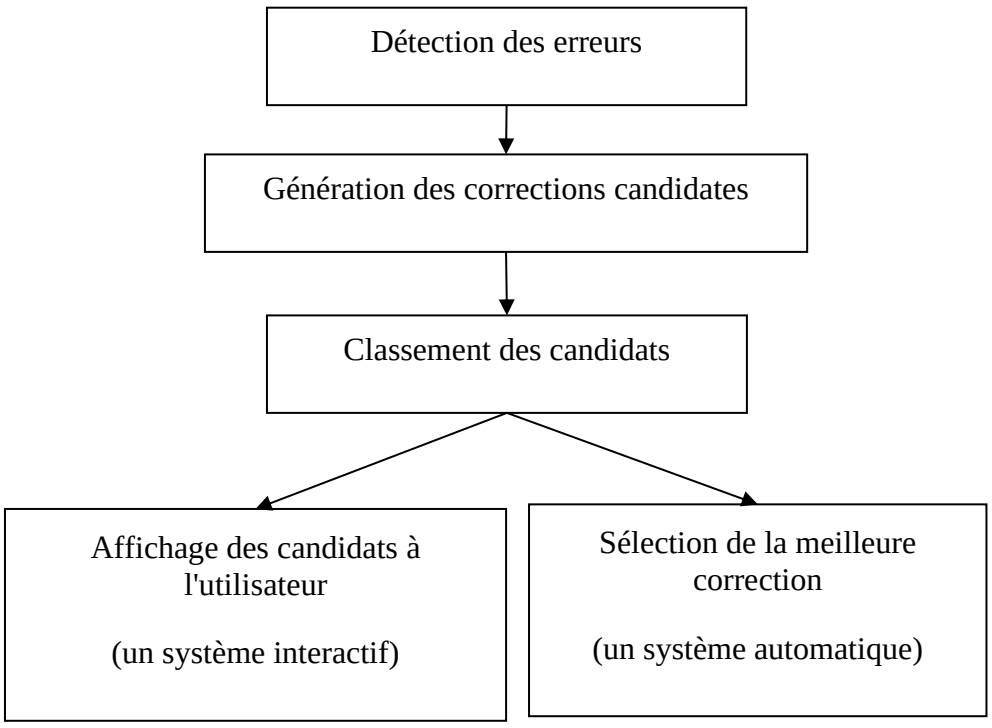

Figure 7 Processus de correction automatique des erreurs.

La détection d'erreurs est le processus de trouver la position et les limites d'un mot ou des mots mal utilisé dans un fragment de texte. Une erreur non détectée ne peut pas être corrigée. Si elle est détectée, l'étape suivante sera la génération de candidats de correction. La troisième étape est le classement des candidats correcteurs à partir des plus probables. Les trois phases du processus peuvent être réduites en une seule partie dans les systèmes modernes, mais elles sont généralement distinctes. Dans les systèmes interactifs, la liste classée des corrections candidates est présentée à l'utilisateur. Dans un système de correction syntaxique purement automatique, l'erreur est remplacée par le meilleur candidat de la correction. Les méthodes de correction syntaxique peuvent varier à chaque phase. Le changement intervenu à la phase précédente a un impact sur les étapes ultérieures et, enfin, sur le résultat de la correction.

1.3. LES DIFFICULTÉS DE CORRECTION DES ERREURS SYNTAXIQUES

La correction automatique des erreurs syntaxiques présente deux difficultés : la nature des langues humaines difficile à coder ces erreurs et le haut niveau de complexité technique. L'ambiguïté de la langue a un impact élevé sur la nature des erreurs linguistiques. Les aspects les plus importants qui rendent la tâche de correction automatique des erreurs syntaxiques difficile sont les suivants :

- Plusieurs corrections pour une seule erreur sont possibles. De nombreuses erreurs peuvent être corrigées de plusieurs manières, par exemple :

القسم في [المعلمون ou المعلمين] [nom/ دخل ou verb/ دخل]
la_classe dans les professeurs [nom/revenu ou verb/sont_entrés]

- Des corrections alternatives sont possibles en raison du manque de correction sémantique nécessaire pour décider quelle correction est la plus précise, en raison de l'ambiguïté d'une langue. Par exemple, le choix de nombreux articles ou déterminants dépend des préférences personnelles de l'auteur. Diverses corrections peuvent nécessiter des modifications dans différentes parties d'une phrase.
- Les erreurs syntaxiques se produisent avec une fréquence basse [89]. La fréquence des erreurs est généralement mesurée par rapport au nombre total des phrases, au nombre total des mots ou au nombre des mots spécifiques à un type d'erreur donné.
- Certaines erreurs dépendent de choix de mots distincts. L'existence d'un mot particulier peut influer sur le choix d'un autre mot à longue distance. Un mot particulier dans une phrase peut dicter le choix du mot beaucoup plus tard, en particulier dans les phrases complexes composées de multiples clauses. La détection de l'erreur peut nécessiter l'analyse d'un contexte large. La

connaissance des relations syntaxiques entre les mots d'un analyseur syntaxique peut être utile pour détecter une telle erreur.

- Plusieurs erreurs dans une phrase peuvent entraîner la survenue d'une erreur dans un contexte utilisé pour la détection d'une autre erreur. Ce phénomène n'indique également que dans le cas où différents composants sont utilisés pour détecter plusieurs erreurs typiques séparément, une méthode qui combine ces corrections est recommandée.

La correction automatique des erreurs syntaxiques pose également un certain nombre de problèmes en tant que tâche de TAL en raison de sa complexité technique élevée. De nombreux systèmes de hautes performances utilisent des composants qui exécutent de nombreuses sous-tâches du TAL, telles que la segmentation des phrases, la tokenisation des mots, l'étiquetage POS, l'analyse syntaxique, la reconnaissance d'entité nommée ou la correction des erreurs d'orthographe. Ces complexités pourraient être une source des problèmes suivants :

- La plupart des outils du TAL traitent des textes d'entrée sans erreur. Comme beaucoup de ces outils sont développés sur des textes corrects, leurs performances sont souvent sous optimales sur des textes comportant des erreurs de langue.
- Plus l'efficacité des composants de chaque système est faible, plus les performances finales sont faibles. Même si les outils exécutant des tâches bien étudiées, telles que l'étiquetage POS, n'obtiennent pas des résultats parfaits. En outre, les limites des composants individuels empilent le traitement en pipeline. Par exemple, les étiquettes POS mal attribuées ont un impact négatif sur les performances de l'analyse syntaxique.
- Les outils basés sur les données statistiques du TAL peuvent masquer certaines erreurs ou les rendre plus difficiles à détecter. Par exemple, si un étiqueteur POS statistique a été formé uniquement sur des données sans erreur, il pourrait affecter une étiquette POS incorrecte à des mots.

2. Les approches de la correction automatique des erreurs syntaxiques

Au cours des dernières décennies, plusieurs approches ont été proposées pour la correction automatique des erreurs syntaxique. La classification générale des méthodes développées les plus réussies est présentée dans la figure 8. Nous pouvons distinguer deux types de correction :

- Les erreurs lexicales (ou non-word errors).
- Les erreurs grammaticales (ou real-word errors)

Les techniques standard de détection des erreurs isolées sont basées sur la recherche dans le dictionnaire ou l'analyse des caractères n-grammes [90]. Comme les erreurs lexicales ne sont pas visées dans cette thèse, ces techniques ne sont pas décrites dans les parties suivantes du chapitre.

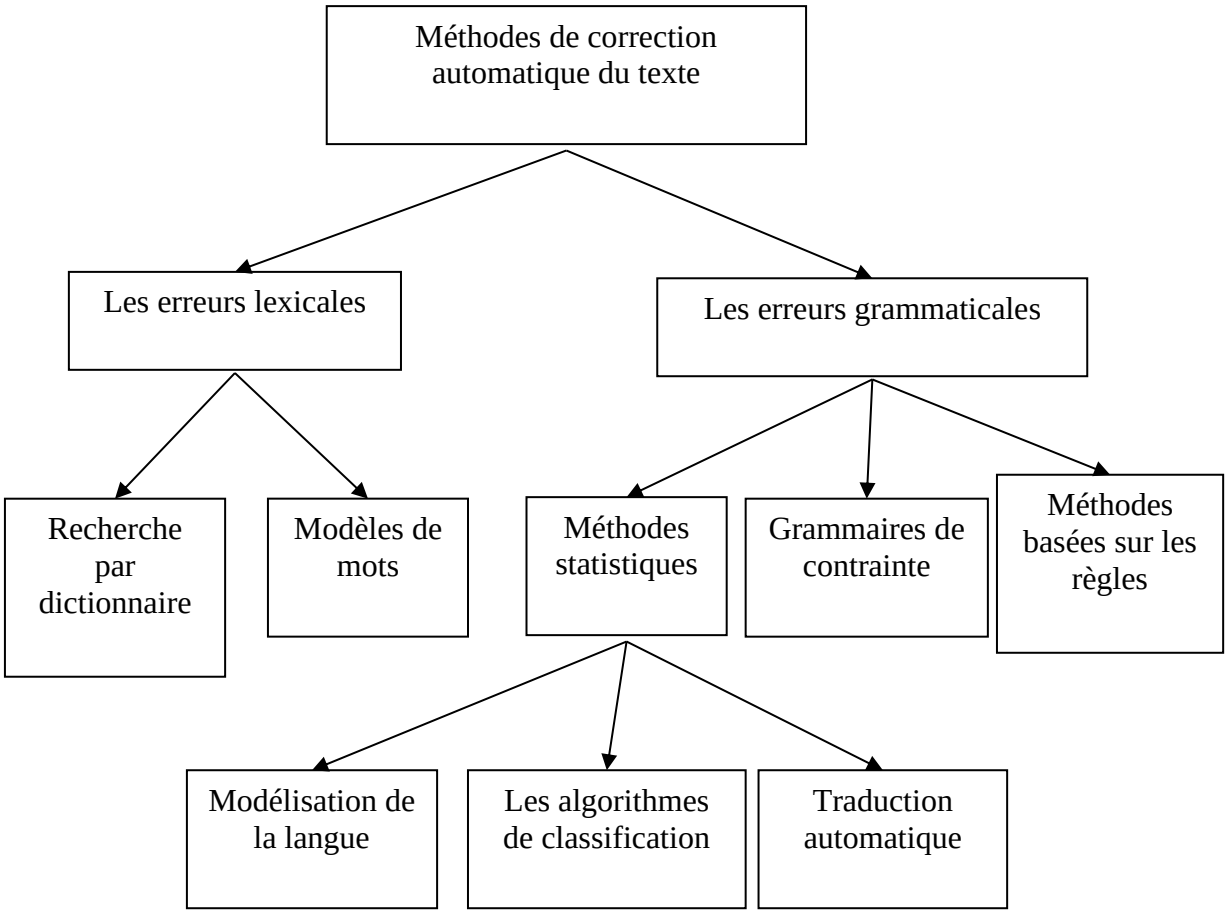

Figure 8 Classification des approches de correction de texte en fonction de type des erreurs

Un correcteur syntaxique est un programme complexe qui nécessite beaucoup de recherches et de ressources linguistiques. De nos jours, les correcteurs syntaxiques, bien qu'ils soient encore loin d'être parfaits, sont meilleurs (pour identifier et corriger les erreurs) et plus faciles à utiliser. En effet, il est difficile de les ignorer.

Les erreurs syntaxiques peuvent être corrigées à l'aide de règles prenant en compte le contexte autour d'une erreur. Les méthodes qui sont basées principalement sur des règles créées manuellement sont appelées méthodes basées sur les règles. Les méthodes basées sur la grammaire reposent sur des grammaires formelles et utilisent des algorithmes d'analyse syntaxique modifiés pour détecter et corriger les erreurs syntaxiques [91]. Les règles peuvent également être acquises de manière transparente à partir de méthodes basées sur des données statistiques. Les méthodes d'apprentissage automatique les plus couramment utilisées pour la correction automatique des erreurs syntaxique sont la modélisation linguistique, les algorithmes de classification et la traduction automatique statistique. Il convient de noter que certaines erreurs lexicales peuvent également être efficacement détectées et corrigées par des méthodes statistiques communément utilisées pour les erreurs syntaxiques si elles se produisent assez fréquemment dans les données d'apprentissage.

Dans la suite de cette section, nous décrivons certaines approches, y compris les méthodes historiques les plus impactantes et un état de l'art sur les solutions qui suivent principalement les approches statistiques et/ou hybrides.

2.1. LES MÉTHODES BASÉES SUR LES RÈGLES

Les premiers outils de correction syntaxique, tels que l'Unix Writer's Workbench développé par les laboratoires Bell [92] en 1982 et GramCheck [93] en 1996 qui utilisaient des règles et des techniques de reconnaissance de patrons[10] (pattern matching) conçues à la main. Le vérificateur syntaxique le plus utilisé de nos jours, intégré à l'éditeur de texte Microsoft Word [94] repose principalement sur une approche basée sur les règles. Les langues traitées sont le chinois, l'anglais, le français, l'allemand, le japonais, le coréen et l'espagnol.

[10] Cette technique peu évoluée est notamment destinée à évaluer les réponses à des questions ouvertes ou semi-ouvertes, où l'apprenant doit rédiger une phrase complète ou un segment de phrase. Selon Winograd (1983), un patron linguistique peut être défini comme une description d'une forme possible d'une langue, par analogie avec un patron au sens propre, qui est un objet dont la forme est identique à la pièce qu'on veut découper dans du tissu ou d'autres matières. Un patron servira à retrouver dans un texte des formes ayant une même construction.

```
<Règle id ="Accord_il" nom="Erreur d'accord : verbe troisième
personne avec 'il' " >

    <Pattern lang="Fr">

      "a| sera| doit| pourrait| fait| a fait|" "il" VBN

    </Pattern>
    <Message>
      'Il' doit être utilisé avec un verbe à la troisième personne,
      par exemple 'il marche'.
    </Message>
    <Exemple type="correcte">
      Il <em>regarde</em> le bâtiment.
    </Exemple>
    <Exemple type="incorrecte">
      Il <em>regardes</em> le bâtiment.
    </Exemple>
</Règle>
```

Figure 9 : Exemple d'outil linguistique pour les règles de correspondance d'erreur

Les systèmes de correction automatique de syntaxes modernes basés sur les règles prennent généralement en charge des fonctionnalités telles que :

- La reconnaissance de patrons avec des expressions régulières.
- Les suggestions de correction alternatives.
- L'analyse du texte d'entrée à plusieurs niveaux linguistiques fournissant les résultats de l'analyse morphosyntaxique, du découpage des phrases nominales ou de l'analyse de dépendance.
- La définition des opérations d'extraction de formes logiques de la phrase, sur les modèles et les règles. Par exemple, la négation, l'union ou l'intersection.
- La génération de suggestions avec synthétiseur de formes infléchies.

Un exemple de détection d'erreurs d'accord est présenté à la figure 9. La règle correspond à la conjugaison du verbe de troisième personne singulier et suggère une correction alternative (Il regarde).

Bien que les règles de correspondance d'erreur soient généralement créées manuellement à partir de modèles d'erreurs fréquemment observés dans des corpus de texte, elles peuvent également être acquises par des méthodes automatiques ou semi-automatiques. Par exemple, le projet dans la communauté des logiciels libres sous le nom de “Language Tool”[11] qui utilise des algorithmes d'apprentissage basés sur la transformation pour acquérir automatiquement des règles symboliques.

[11] https://www.languagetool.org/ , dernier accès le 1 juin 2019.

Depuis les années 90, conformément aux tendances générales du TAL, l'approche basée sur les règles a été remplacée par des méthodes basées sur les statistiques et les méthodes pilotées par les données en raison de la disponibilité croissante de corpus annotés. Toutefois, les composants basés sur des règles peuvent toujours être présents dans les systèmes de correction automatique des erreurs syntaxique en utilisant des approches combinées.

Les principaux avantages de l'approche basée sur les règles décrites par Naber [85] en 2003 dans son correcteur syntaxique et stylistique de l'anglais destiné à la suite de logiciels libres de bureautique OpenOffice :

- Les erreurs pour lesquelles le contexte est clairement défini sont faciles à détecter et à corriger, quelle que soit leur fréquence dans les corpus d'entraînement.
- Les règles peuvent être créées progressivement. Un vérificateur syntaxique basé sur les règles fonctionne immédiatement après la mise en œuvre de la première règle.
- La configuration facile du système : chaque règle a sa propre description et son propre message d'erreur. De plus, chaque règle peut être activée ou désactivée individuellement.

D'autre part, les inconvénients de cette approche sont les suivants :

- L'élaboration des règles manuellement est coûteuse et prend du temps, et doit généralement être effectuée par des linguistes spécialistes.
- La grande dépendance linguistique, car les règles doivent être développées séparément pour chaque langue.
- Difficulté à résoudre les erreurs de classe ouverte, par exemple les erreurs flexionnelles et dérivationnelle dans des langues hautement flexionnelles.

2.2. LES MÉTHODES BASÉES SUR LA GRAMMAIRE FORMELLE

Dans le passé, les chercheurs ont tenté d'établir des règles basées sur le formalisme de la grammaire de structure syntaxique augmentée ("Augmented Phrase Structure Grammar"—APSG) [95] en 1975, avec une approche dite grammaticale. Les systèmes basés sur la grammaire conviennent mieux aux erreurs syntaxiques, car ils traitent explicitement des écarts par rapport à une grammaire encodée manuellement. Une grammaire doit tolérer les erreurs pour analyser une phrase comportant des erreurs syntaxiques. Par conséquent, la grammaire formelle doit être adaptée à des entrées potentiellement erronées. Bien qu'il ne soit pas possible d'analyser tous les types d'erreurs arbitraires et/ou de combinaisons d'erreurs, il est possible d'intégrer dans la grammaire des connaissances sur les erreurs courantes.

Plusieurs techniques ont été proposées pour répondre à cette exigence. Par exemple, dans le cas d'un accord sujet-verbe, une grammaire formelle peut contenir une règle combinant une phrase nominative sujet avec une phrase verbale dans une phrase déclarative, même si le sujet n'est pas en accord avec le verbe principal. Une telle règle—appelée "mal-rule" dans la littérature—contiendrait également une sorte d'indicateur qui indique la présence de cette erreur spécifique [96]. De même, au lieu d'ajouter une nouvelle règle spéciale pour une erreur, il est également possible de « relâcher » les règles en rendant facultatives certaines contraintes d'accord [97]. Dans notre exemple, la grammaire pourrait contenir une règle sujet soumettant la contrainte selon laquelle le sujet d'une phrase déclarative soit en accord avec le verbe principal, mais cette contrainte peut être moins rigide que d'autres contraintes dans le sens qu'elle peut être violée au prix d'élever un drapeau d'erreur. Une troisième méthode permettant d'analyser une entrée erronée consiste à autoriser la génération excessive d'arbres d'analyse grâce à l'utilisation de règles relativement permissives combinées à un mécanisme de classement d'arbres d'analyse [98]. Si aucune analyse complètement bien formée ne peut être trouvée, l'analyse la moins hiérarchisée "less ideal" peut être utilisée pour détecter la présence éventuelle d'une erreur. De plus, une quatrième méthode de corriger les erreurs dans une grammaire formelle consiste à autoriser des analyses partielles permettant au moins d'identifier les morceaux principaux d'une phrase. Des règles spéciales de post-traitement peuvent ensuite examiner les éléments de cette analyse et tenter de trouver les sources de l'échec de la combinaison des éléments dans une analyse de phrase complète [99].

2.3. LES MÉTHODES STATISTIQUES

Les approches statistiques sont devenues de plus en plus importantes dans la communauté TAL au cours des dernières décennies. Les principales raisons de cette tendance sont :

- Les approches statistiques ont tendance à être moins fragiles que les systèmes d'analyse créés manuellement, car elles sont formées sur un grand ensemble de données.
- Une ingénierie de règles coûteuses n'est pas nécessaire (bien que la création du corpus d'entraînement pour les systèmes statistiques ne soit pas économique non plus).
- Expérimenter de nouveaux algorithmes est simple, car il ne nécessite que de recycler le nouveau système sur les mêmes données que celles utilisées lors de l'entraînement du système précédent, permettant ainsi des itérations rapides sur les améliorations des algorithmes.

Les méthodes statistiques sont classées en trois techniques dominantes, à savoir :

2.3.1. LA MODÉLISATION DE LANGUE

Un modèle de langue (ML) statistique (ou probabiliste) est une distribution de probabilité sur une séquence de caractères ou de mots. ML attribue la probabilité $P(t_1, \ldots, t_m)$ à la séquence de mots $t_1, \ldots, t_m$ de longueur m, qui tente de refléter la fréquence et la probabilité de réalisation de la séquence donnée dans la langue. Les probabilités sont estimées sur un grand corpus, qui se compose idéalement de textes sans erreur.

Les modèles de langue statistiques les plus populaires sont les modèles n-grammes. Un n-gramme est une séquence de mots $t_1, \ldots, t_n$ de longueur n. Si n est égal à 1, 2 ou 3, un n-gramme est appelé respectivement uni-gramme, bi-gramme ou tri-gramme. Un modèle de langage n-gramme estime la probabilité p d'une séquence de mots T sur la base des n - 1 mots précédents à l'aide de la propriété de Markov du nième ordre [100] :

$$p(T) \approx \prod_{i=1}^{|T|} p(t_i | t_i, \ldots, t_{i-n+1})$$

L'utilisation d'un modèle de langage en vue de la détection des erreurs syntaxiques permet essentiellement de généraliser le fait que les séquences de mots rares sont plus susceptibles d'être erronées que les plus courantes. Cette généralisation ne peut cependant pas être appliquée naïvement ; une erreur peut-être détectée si un mot très improbable est trouvé dans une phrase (selon le modèle de langue). Par exemple, une phrase telle que « J'ai vu Chouaib hier » qui contient un nom rarement utilisé en français et peut-être jamais été rencontrées au cours de l'entraînement de corpus. Le mot « inconnu » avec sa faible probabilité conduira à un score de modèle de langue inférieur pour les séquences contenant ce nom, bien qu'il n'y ait pas d'erreur dans cette phrase.

Atwell [101] décrit l'une des premières tentatives d'utilisation de modèles de langue pour la correction automatique des erreurs syntaxiques. L'auteur a utilisé un modèle de langage étiqueté pour gérer les transitions d'étiquettes POS peu probables dans le texte d'entré. Il a proposé plusieurs méthodes pour déterminer le moment où une erreur devait être diagnostiquée. Il a expérimenté l'utilisation de probabilités d'erreur créées manuellement qui mesurent la fréquence à laquelle chaque paire de balises survient, trouvant un seuil optimal à des faibles vraisemblances absolues pour les balises POS, ou ajoutant des balises d'erreur aux entrées lexicales qui, lorsqu'elles sont utilisées, indiquent l'occurrence d'une erreur.

L'évaluation des connaissances lexicales "Assessment of Lexical Knowledge—ALeK" développé par Leacock & Chodorow [102] est un autre exemple de système prenant en compte les probabilités de séquences de mots pour la détection des erreurs. ALeK est basé sur des statistiques bi-grammes (paire de mots) recueillies à partir d'un grand corpus d'actualités anglaises étiqueté POS. Ce

n'est pas un système de modèle de langue dans le sens où il attribue des scores à des séquences de mots de longueur arbitraire. Certes, il se concentre sur les paires de mots et leur association statistique. En utilisant deux métriques d'association (information mutuelle par point et rapport de vraisemblance), il peut identifier des paires de mots très improbables. Par exemple, la combinaison de deux noms singuliers « carte| <NOM> mémoire| <NOM> » correspond à une paire de mots très fortement associés, alors que la combinaison d'un déterminant singulier et d'un nom pluriel « une| <DET> mémoires| <NOM> » est très peu (s'il y en a) d'association.

Des recherches récentes de Hdez et Calvo [103] en 2014 portent sur l'examen d'un modèle de langage basé sur des trigrammes syntaxiques et des bi-grammes extraits d'arbres de dépendance générés à partir de Wikipédia.

Les avantages les plus significatifs de l'approche basée sur les ML pour la correction automatique des erreurs syntaxiques sont les suivants :

- Les modèles de langue n'ont besoin que des ressources de texte brut largement disponibles.
- Les modèles de langue peuvent être facilement combinés avec d'autres méthodes.

D'autre part, les inconvénients de cette approche sont les suivants :

- Les n-grammes qui n'apparaissent pas dans les données d'entraînement et par conséquent possèdent une probabilité nulle doivent être traités avec des méthodes de lissage[12] "smoothing methods".
- La distinction entre les n-grammes rares et les n-grammes non grammaticaux doit être modélisée.

2.3.2. LA CLASSIFICATION

La méthode de classification repose sur des méthodes d'apprentissage automatique supervisées. Un classifieur statistique est un modèle qui prend une décision concernant de nouvelles données, basé sur des données au moment de l'entraînement du corpus. La correction des erreurs est perçue comme une tâche de désambiguïsation lexicale ou de sélection de mots. L'ambiguïté entre les candidats de correction est modélisée par des ensembles de confusion prédéfinis (ou ensemble de candidats) contenant des mots souvent confondus.

[12] En statistiques, le lissage est une technique qui consiste à réduire les irrégularités et les désajustements d'une courbe. Cette technique est utilisée en traitement des images afin de créer une fonction d'approximation qui tente de capturer des motifs importants dans les données, tout en laissant à côté ce qui peut être considéré comme une perturbation ou un bruit de mesure.

Par exemple, la correction des erreurs de choix de préposition. Au moment de l'entraînement, un classificateur pour ce type d'erreur considérera chaque préposition observée dans un corpus d'entraînement comme un cas d'apprentissage. Les preuves seront présentées au classificateur sous la forme de « traits » du cas d'entraînement individuel. Le trait le plus important est le choix de la préposition elle-même, qui est souvent appelé le « trait cible », c'est-à-dire la cible que le classificateur doit prédire. D'autres traits que le classificateur doit prendre en compte pour faire ses prédictions peuvent être des mesures telles que « quel est le mot à gauche / à droite de la préposition », « quel est le verbe qui gouverne la préposition », « quels sont les étiquettes POS des mots dans le contexte de la préposition », etc.

La figure 10 illustre la tâche de classification dans un ensemble confus composé de huit prépositions les plus fréquentes {pour, à, sur, en, de, dans, avec, par}. Un mot qui appartient à l'ensemble de confusion et qui a été rencontré dans le texte d'entrée est appelé un mot source (prépositions 'de' et 'en' dans l'exemple). La tâche du classifieur est de décider, pour chaque mot source, lequel du nombre fini d'alternatifs possibles parmi l'ensemble candidat est la plus précise dans le contexte donné. Le contexte est représenté par les mots voisins de la phrase dans laquelle le mot source apparaît et modélisé par les traits du contexte. L'algorithme de classification est entrainé sur un ensemble d'exemples étiquetés extraits d'un corpus des erreurs annoté.

	pour *à* *sur*		*pour* *à* *sur*
Une étude	*de* *dans* *avec* *en* *par*	l'Université Hassan 2	*en* *dans* *avec* *de* *par* 2019 a montré (…)

Figure10 : Exemple d'approche de classification pour la correction de préposition

Les applications des classificateurs à la correction automatique des erreurs syntaxique varient selon trois aspects : la sélection d'ensemble de confusion, la conception des traits de contexte et le choix des algorithmes de classification [104].

i. Les ensembles de confusion

L'approche de classification a été initialement appliquée à la correction des erreurs d'orthographe en fonction du contexte [105]. Dans cette tâche, les ensembles confus consistent généralement en un petit nombre de mots homophones (par exemple, {sans, sang}), ayant une orthographe similaire (par exemple, {vers, vert}) ou partageant certaines fonctions grammaticales (par exemple, {parmi, entre}).

Les erreurs de préposition et d'article sont des exemples des erreurs de classe fermée, qui peuvent être efficacement modélisées par des ensembles de confusion finis. Cependant, l'approche basée sur la classification a également été utilisée pour les erreurs de classe ouverte, tel que les erreurs d'accord entre sujets-verbes [106]. Des ensembles de confusion sont générés à la volée pour chaque mot avec une étiquette POS spécifique basée sur les propriétés linguistiques des erreurs particulières.

ii. Les traits du contexte

L'ensemble des traits de contexte est généralement développé spécifiquement pour chaque type d'erreur. Les traits utilisés pour la correction des erreurs orthographiques sont généralement basés sur les traits n-gram et sac de mot.

Des traits plus sophistiqués, à motivation linguistique, sont généralement conçus pour les erreurs d'articles ou de répétition, ainsi que pour les erreurs de classe ouverte. Toutefois, pour certains types d'erreur, tels que les erreurs de préposition ont obtenu des résultats adéquats avec les ensembles de traits n-grammes uniquement.

Rozovskaya et Roth [107] ont présenté les deux paradigmes, à savoir, le paradigme de sélection et le paradigme de correction. La différence entre les deux est de savoir si le mot source est pris en compte lors de l'entraînement ou non. Dans le paradigme de sélection, le mot source n'est pas utilisé en tant que trait. Par conséquent, les modèles de correction d'erreur sont généralement formés à partir de données natives, puis appliqués à des textes non natifs. Le paradigme de correction, à son tour, utilise les choix de l'auteur original et nécessite des données annotées d'erreur. Récemment, avec la disponibilité de grands corpus annotés, le paradigme de correction est utilisé plus couramment et donne de meilleurs résultats que le paradigme de sélection, qui était auparavant plus populaire.

iii. Les algorithmes de classification

Jusqu'à présent, un nombre important d'algorithmes de classification ont été utilisés dans la correction automatique des erreurs syntaxique. Par exemple : les arbres de décision, le classificateur Naïf Bayes, l'algorithme de Winnow, ou régression linéaire et logistique, etc. Le choix du classificateur, contrairement au choix d'un ensemble de traits, il est généralement indépendant de l'ensemble de confusion et des erreurs typiques ciblées.

Il a été démontré que les méthodes basées sur les algorithmes de classification produisent des résultats les plus intéressants pour la correction d'un certain nombre des erreurs typiques. C'est l'une des approches les plus activement étudiées au cours de la dernière décennie dans le domaine des erreurs syntaxiques. En outre, il y a une large gamme d'améliorations aux modèles de classification est possible. Les avantages les plus importants de cette approche sont les suivants :

- Une large gamme d'algorithmes de classification et de techniques d'apprentissage automatique bien connu peut être appliquée.
- Il est possible de cibler des erreurs typiques spécifiques en concevant divers ensembles de confusion.
- Lorsque les données d'entraînement sont disponibles, la méthode est moins coûteuse et plus générale que l'approche basée sur les règles.
- La méthode est essentiellement indépendante de la langue, à l'exception du besoin de données annotées pour une langue donnée.

Malgré la popularité des méthodes de classification, la construction d'un modèle distinct pour chaque erreur syntaxique est une tâche complexe. Les principales limites sont :

- Seules des erreurs spécifiques de classe fermée peuvent être directement modélisées par des ensembles de confusion finis. La modélisation des erreurs de classe ouverte est plus difficile.
- Une grande quantité de données textuelles annotées sont nécessaires à des fins d'entraînement. La taille et la qualité de l'ensemble de données influencent grandement les performances.
- Un modèle distinct doit généralement être développé pour chaque ensemble de confusion et type d'erreur, ce qui peut inclure une ingénierie de fonctionnalités complexe.
- Le problème d'insertion des mots doit être traité séparément, au-delà du modèle de classification.

2.3.3. LA TRADUCTION AUTOMATIQUE STATISTIQUE

La correction automatique des erreurs syntaxiques peut également être considérée comme une sorte de tâche de traduction automatique. La traduction est effectuée à partir de texte comportant des erreurs interprétées comme langue source en un texte sans erreur, traité comme langue cible.

En traduction automatique statistique basée sur les phrases. Pour une phrase d'entrée S, la phrase correcte suggérée $\hat{T}$ maximise la probabilité conditionnelle par rapport aux corrections possibles [108].

La probabilité est calculée en utilisant un modèle log-linéaire en tant que combinaison pondérée de fonctions $h_i\ (T|S)$:

$$\hat{T} = arg\max_{T} p(T|S)$$

$$\approx arg\max_{T} exp(\sum_{i=1} \lambda_i \log h_i\ (T|S))$$

Les fonctions typiques sont le modèle de traduction apprit à partir d'un corpus parallèle aligné par phrase et le modèle de langue estimé sur des textes sans erreur. Les poids λ_i doivent être appris conformément à la métrique d'évaluation.

Pour la première fois, cette approche a été appliquée à la correction des erreurs syntaxique par Brockett et al. [109] en 2006. Les auteurs utilisent la traduction automatique statistique afin de corriger quelques erreurs syntaxiques pour un ensemble de 14 noms en masse qui posent des problèmes aux apprenants d'anglais comme seconde langue. En ce qui concerne cette tâche très limitée, ils obtiennent les résultats d'environ 62% d'erreurs corrigées et montrent que leur système peut battre le correcteur syntaxique Microsoft Word 2003. Dahlmeier et Ng [110] présentent un décodeur de recherche du faisceau personnalisé qui intègre des classificateurs discriminants pour des catégories d'erreur spécifiques telles que des articles et des prépositions.

Les recherches ont commencé à s'attaquer à aux problèmes de correction automatique des erreurs syntaxique basée sur la traduction automatique statistique dans les années 2010. Ces dernières années, la correction syntaxique a fait l'objet de nombreuses études, et ce, notamment avec les campagnes d'évaluation (une description détaillée sur les campagnes d'évaluation dédiées à la correction automatique des erreurs syntaxiques est présentée dans la section III) dans laquelle les participants ont utilisé le même ensemble de données pour former et évaluer leurs modèles. Elles ont été organisées depuis 2011 par les campagnes d'évaluation "Helping Our Own—HOO" en 2011.

Les systèmes les plus remarquables de correction syntaxique basés sur la traduction automatique statistique sont celles qui ont participé aux campagnes d'évaluation CoNLL [111, 112].

Des travaux récents qui s'appuient également sur les systèmes de correction syntaxique exploitent les méthodes de reclassement par liste n-best ou de nouvelles fonctionnalités, telles que les modèles de traduction automatique neuronaux ("Neural Machine Translation"—NMT) —"feed-forward" [113].

L'utilisation de la traduction automatique statistique présente plusieurs avantages par rapport à d'autres approches de correction automatique des erreurs syntaxiques, à savoir :

- L'absence de restriction à des types spécifiques des erreurs permet de corriger un nombre important des erreurs, y compris les erreurs de classe ouverte.
- Les systèmes de traduction automatique statistiques basés sur les phrases peuvent naturellement corriger les erreurs dans les phrases, pas seulement pour les mots individuels.

- Les données parallèles avec les versions corrigées des phrases sont suffisantes à des fins d'entraînement, c'est-à-dire qu'aucune annotation pour les erreurs typiques n'est requise.
- Les données monolingues, qui sont plus facilement disponibles que les données parallèles, peuvent être facilement intégrées aux systèmes de correction syntaxique sous forme de modèles de langue.
- L'approche de traduction automatique est relativement facile à adapter à d'autres langues.
- Construire un système basé sur la traduction automatique statistique ne nécessite pas de connaissances linguistiques d'experts en correction des erreurs syntaxiques.

Les inconvénients de cette approche sont :

- Les performances des systèmes basés sur la traduction automatique statistique dépendent fortement de la quantité et de la qualité des données d'entraînement.
- Les erreurs qui n'ont pas été observées dans les données d'entraînement ne peuvent pas être corrigées. Par exemple, les erreurs d'orthographe.
- La correction des erreurs dans les dépendances à longue portée d'une phrase peut être difficile pour les systèmes de traduction automatique statistiques basés sur les phrases.
- Le contrôle et la manipulation de correction des erreurs typiques individuelles est difficile lorsque l'entraînement ne se limite pas à eux.

2.4. APPROCHES COMBINÉES

Diverses méthodes semblent mieux adapter à la correction du type spécifique des erreurs. Les systèmes de correction automatique des erreurs syntaxique qui utilisent des composants séparés pour traiter différents types d'erreurs doivent incorporer des techniques de combinaison de différentes approches.

Gamon [114] en 2010 a combiné les modèles de langue et les classificateurs dans une approche de méta-classification. En effet, les modèles de langue fournissent des informations supplémentaires pour filtrer les suggestions parasites des classificateurs spécifiques aux erreurs. Dahlmeier et Ng [115] en 2011 ont étudié l'optimisation des structures alternatives, une méthode permettant de combiner des classificateurs d'une manière d'apprentissage par transfert, dans laquelle les données sans erreur sont utilisées pour l'entraînement et le modèle est ensuite appliqué aux données contenant des erreurs. Gamon [116] en 2011 a expérimenté la modélisation de séquence, une technique qui opère sur la séquence de mots dans une phrase, modélisant la probabilité d'une erreur à un emplacement donné de la phrase en prenant en comptes divers traits du contexte. Rozovskaya et

Roth [117] en 2011 ont montré un autre avantage des modèles statistiques pour la correction d'erreur non-native, à savoir leur capacité d'inclure des informations sur les erreurs typiques. L'intégration de ces connaissances permet à un système de traduction automatique statistique à cibler des groupes spécifiques d'utilisateurs selon la régularité dans leurs modèles d'erreur.

Wagner [118] en 2012 indique que les grammaires statistiques peuvent être utilisées pour la détection d'erreurs. Les grammaires statistiques (ou probabilistes) ne reposent pas sur des règles manuelles, mais utilisent plutôt des règles et des probabilités associées à partir d'un vaste corpus de phrases syntaxiquement annotées, appelé "Tree Bank", tel que le "Penn Tree Bank" [119]. Pour les grammaires probabilistes, aucun mécanisme de relaxation ou règles mal – mal rules - ne peut être réglé manuellement. Néanmoins, ces grammaires tendent à être très admissible, c'est-à-dire qu'elles sont généralement capables d'analyser presque toutes les phrases erronées, mais avec une probabilité d'analyse plus faible que pour une phrase standard et sans erreur. Wagner a montré qu'avec d'autres traits, la probabilité à partir de ces grammaires peuvent être utilisée en tant que signal de détection d'erreurs. Notez que cette orientation de recherche brouille la frontière entre les méthodes basées sur la grammaire et les méthodes statistiques. Nous pouvons dire qu'elle a plus d'affinités pour les techniques statistiques, car les grammaires probabilistes ne permettent pas le contrôle manuel des règles de grammaire.

Ehsan et Faili [120] en 2013 combinent un système de correction automatique des erreurs syntaxiques formé par des phrases artificielles erronées avec un correcteur syntaxique basé sur des règles dans un système interactif pour l'anglais et le farsi en agrégeant les candidats de correction des deux composants. Les auteurs montrent que les deux approches sont complémentaires et que le système hybride offre les meilleures performances.

En 2014 Felice et al. [121] explorent diverses stratégies pour la combinaison de méthodes basées sur les règles avec un système de traduction automatique statistique et des modèles de langue. Leur système hybride, qui s'est classé premier dans les campagnes d'évaluation de CoNLL-2014, met en pipeline un module basé sur les règles avec un système traduction automatique statistique basé sur des phrases produisant des corrections candidates, qui sont ensuite classées par un modèle de langue basé sur le Web et filtrées par le type d'erreur.

Dans les travaux récents, Rozovskaya et Roth [122] en 2016 montrent la complémentarité des approches de classification et de traduction automatique statistique en mettant en pipeline deux systèmes. De nouveaux résultats techniquement encourageants sur l'ensemble de tests CoNLL-2014 sont obtenus.

3. Les campagnes d'évaluation pour correction des erreurs syntaxiques

Au cours des dernières années, un nombre important de progrès remarquable a été réalisés dans le domaine de correction automatique des erreurs syntaxique. Ils sont le fruit de diverses campagnes d'évaluation—"shared tasks". Ce sont des compétitions ouvertes, où les équipes participantes sont encouragées à développer des systèmes de correction automatique des erreurs syntaxique dans le respect des contraintes définies par les organisateurs. Ils fournissent des ensembles de données et des cadres d'évaluation qui permettent une évaluation objective et comparable des méthodes développées. Les campagnes d'évaluation diffèrent les unes des autres, ils se concentrent sur différents types d'erreurs ou en ciblent différents domaines et langues.

Les métriques d'évaluation les plus populaires spécifiques à la correction automatique des erreurs syntaxiques dans les compagnes d'évaluation sont la précision (P), le rappel (R) et le F-score ($F_{0,5}$).

3.1. LES CAMPAGNES D'ÉVALUATION HOO

Les compétitions de la correction automatique des erreurs syntaxique ont été lancées par la compagne d'évaluation ("Helping Our Own"—HOO) [123] organisée en 2011. Le but était de développer des méthodes automatiques qui aident les auteurs au sein de la communauté TAL à la rédaction des articles scientifiques. Des extraits de texte annoté à partir d'articles ayant déjà étés publiés dans les actes de conférence ou d'atelier de l'association de linguistique computationnelle ("Association for Computational Linguistics"—ACL) ont été utilisés comme données d'entraînement et de test. Les performances des six systèmes participants ont été évaluées à l'aide d'une mesure F-score [124] au niveau de détection, de reconnaissance et de correction.

La deuxième édition de la compagne d'évaluation HOO 2012 [125] était axée sur la détection et la correction des erreurs de détermination et de préposition commises par des non-anglophones. 14 équipes ont participé à la tâche d'évaluation. Le système UI [126] a obtenu le F-score le plus élevé pour les sous-tâches de détection et de reconnaissance, le système NU [127] a remporté la sous-tâche de correction. Les deux systèmes utilisaient des classificateurs en pipeline pour traiter les deux types d'erreurs.

3.2. LES CAMPAGNES D'ÉVALUATION CONLL

Les compétitions les plus percutantes pour la correction des erreurs syntaxique ont été les deux éditions des compagnes d'évaluation CoNLL [128] organisées dans le cadre de conférence sur l'apprentissage du langage naturel ("Conference on Natural Language Learning"—CoNLL) en 2013 et 2014.

L'objectif des compagnes d'évaluation était d'évaluer les algorithmes et les systèmes de correction automatique des erreurs syntaxique dans des essais en anglais écrits par des apprenants non-anglophones. Un système participant devait détecter les erreurs syntaxiques qui se produisent dans les textes d'entrée et renvoyer les textes corrigés de manière entièrement automatique. L'édition de la compagne d'évaluation CoNLL-2013 a été consacrée à la correction des erreurs dans les cinq catégories d'erreurs sélectionnées (erreurs d'article, de déterminant, de préposition, de nom, de forme verbale et d'accord sujet-verbe), qui représente un tiers à la moitié de toutes les erreurs dans les ensembles de données fournies. L'édition CoNLL-2014 ciblait les erreurs syntaxiques de tous types survenus dans les essais, regroupées en 28 catégories. L'absence de restriction pour des types d'erreurs spécifiques a introduit un scénario plus naturel pour la tâche de correction automatique, qui a été suivi dans d'autres compétitions, telles que QALB (Qatar Arabic Language Bank) ou AESW (Automated Evaluation of Scientific Writing). Les équipes participantes ont reçu des données d'entraînement avec des corrections syntaxiques annotées manuellement et ont été autorisées à utiliser des données supplémentaires accessibles au public pour le développement. L'ensemble de données d'entraînement commun mis à disposition par les organisateurs était le corpus d'apprentissage de l'université nationale de Singapour ("National University of Singapore Learner Corpus"—NUCLE).

Au cours de CoNLL-2013, la plupart des systèmes participants ont utilisé l'approche de classification basée sur l'apprentissage automatique, y compris le système gagnant UIUC [129]. Treize soumissions de systèmes ont pris part à l'édition CoNLL-2014. Les résultats officiels sont présentés dans le tableau 4. Le meilleur système (CAMB) utilisait une approche hybride qui consiste à canaliser le système de correction d'erreur basé sur les règles et la traduction automatique statistique, complété par un grand modèle de langue basé sur le Web.

Tableau 4 Les résultats officiels de CoNLL-2014

Rang	ID de l'équipe	Approches	Précision	Rappel	$F_{0,5}$
1	CAMB	Rule-based / Language Model / Machine Translation	39.71	30.10	37.33
2	CUUI	Machine Learning	41.78	24.88	36.79
3	AMU	Machine Translation	41.62	21.40	35.01
4	POST	Language Model / Rule-based	34.51	21.73	30.88
5	NTHU	Rule-based / Language Model / Machine Translation	35.08	18.85	29.92
6	RAC	Rule-based / Language Model	33.14	14.99	26.68
7	UMC	Machine Translation	31.27	14.46	25.37
8	PKU	Language Model / Machine Learning	32.21	13.65	25.32
9	NARA	Machine Translation	21.57	29.38	22.78
10	SJTU	Rule-based / Machine Learning	30.11	5.10	15.19
11	UFC	Rule-based	70.00	1.72	7.84
12	IPN	Language Model / Rule-based	11.28	2.85	7.09
13	IITB	Machine Translation / Machine Learning	30.77	1.39	5.90

3.3. AUTRES COMPÉTITIONS

Des concours similaires axés sur la correction des erreurs ont été organisés pour des langues autres que l'anglais. En 2014 et 2015, deux compagnes d'évaluation QALB sur la correction automatique de texte en arabe ont été organisées de manière similaire aux compagnes d'évaluation de CoNLL [130]. Alors que QALB-2014 traitait les erreurs commises par des arabophones, l'édition de l'année suivante proposait une piste supplémentaire axée sur les erreurs dans les essais écrits par des apprenants non-arabophones. Les équipes participantes ont reçu des données sur l'entraînement et le développement et étaient libres d'utiliser d'autres ressources disponibles au public.

Un autre concours de diagnostic des erreurs grammaticales en chinois - Chinese Grammatical Error Diagnosis (CGED)- organisée lors de l'atelier sur les techniques de traitement du langage naturel à des fins pédagogiques (NLP-TEA) depuis 2014.

4. La correction automatique des erreurs syntaxiques pour la langue arabe

La plupart des recherches sur le traitement automatique des langues en arabe sont consacrées à l'analyse morpho-syntaxique, sans se préoccuper des problèmes de la correction des erreurs syntaxiques. En effet, il existe un nombre très limité de recherches (que nous connaissons jusqu'à présent) sur la correction des erreurs syntaxique en arabe.

Le vérificateur de grammaire Arabic GramCheck [4] est implémenté en Prolog (SICStus) pour certaines erreurs grammaticales courantes. Cet outil a pour objectif initial de détecter l'erreur et de montrer à l'utilisateur les règles syntaxiques violées pour les phrases non-grammaticales et éventuellement de proposer des suggestions d'amélioration. Le système est basé sur une analyse syntaxique approfondie et repose sur une approche de relaxation des traits pour la détection de phrases arabes mal formées.

Récemment, un projet en cours d'élaboration d'un outil Web permet de détecter les erreurs de grammaire arabe en utilisant Deep Learning et les réseaux de neurones notamment Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM) et LSTM bidirectionnels [5]. Chaque modèle d'apprentissage approfondi est formé sur un ensemble de données d'entraînement, qui contiendra des phrases étiquetées. Cependant, l'élément central de ce projet est le corpus arabe, qui devrait non seulement être annoté pour les erreurs linguistiques, mais également indiquer les types d'erreurs. Malheureusement, le corpus disponible ne désigne pas les types des erreurs, ce qui représente un défi majeur pour ce projet.

5. Conclusion

Ce chapitre introduit les notions fondamentales de la correction automatique des erreurs syntaxiques. Nous avons présenté plusieurs approches dans ce domaine et souligné leurs forces et leurs faiblesses.

Récemment, les dernières études ont montré les avantages remarquables des approches basées sur les données statistiques qui ont ouvert la voie à des résultats intéressants. Nous avons montré que cette approche nécessite d'énormes données qui ne sont pas disponibles en langue arabe. C'est ce qui explique notre choix d'adopter l'approche basée sur les règles et qui s'est avéré efficace dans plusieurs systèmes.

À notre connaissance, il n’existe pas de travaux qui traitent la correction automatique des erreurs syntaxique de la langue arabe dans son ensemble à l'exception de quelques recherches très limitées. Notre thèse veut apporter deux contributions à ce domaine.

La partie suivante montre nos travaux de recherche dans cette thèse. nous allons nous focaliser dans le chapitre suivant sur la présentation d'une nouvelle approche de la la correction automatique des erreurs syntaxique en langue arabe.

Deuxième partie

Contributions

Chapitre 4 : Une approche novatrice de la correction automatique des erreurs syntaxiques dans les textes arabes.

Chapitre 5 : La correction des erreurs syntaxique de désinence casuelle en arabe

Chapitre 4

Une approche novatrice de la correction automatique des erreurs syntaxiques dans les textes arabes

1. Introduction

De nombreux travaux se concentrent sur le traitement automatique des langues à plusieurs niveaux. À savoir, la morphologie qui définit la structure des mots [65, 131], la syntaxe qui détermine la composition de phrases [132, 133] et la sémantique qui détermine le sens [134, 135]. Plusieurs programmes tels que la traduction automatique, l'extraction d'informations, le résumé automatique de texte, etc. peuvent les exploiter. Cependant, le défaut d'un tel programme réside dans la relation entre les mots constituant une phrase qui peut être parfois syntaxiquement incorrecte et qui peut donc conduire à des résultats incorrects. Cela nécessite impérativement un système de correction automatique efficace.

La plupart des travaux dans ce domaine portent sur le niveau d'orthographe [136] ; il vérifie simplement l'existence de mots dans le dictionnaire mais ne peut pas détecter les erreurs syntaxiques. En ce qui concerne l'arabe, qui est l'une des langues les plus utilisées sur le Web, les recherches sur ces types d'erreurs restent limitées [4]. Ainsi, la difficulté de corriger les erreurs syntaxiques en arabe s'est illustré à plusieurs niveaux : la complexité et la richesse de cette langue ; l'absence de voyelles dans la plupart des textes ; l'irrégularité de l'ordre des mots dans la construction des phrases ; problèmes d'inflexion des mots (mots se terminant en fonction de leurs cas : nominatif, accusatif, génitif, etc.) ; agglutination ; et d'autres problèmes d'analyse morphologique. Tous ces facteurs entravent le traitement automatique des erreurs à plusieurs niveaux.

L'objectif de notre travail est de concevoir une nouvelle approche capable de traiter automatiquement les erreurs syntaxiques de la langue arabe. Le mot « traitement » peut être défini comme toute manipulation algorithmique d'une entrée – c'est-à-dire, des signaux linguistiques à des fins diverses, telles que la catégorisation, la compréhension jusqu'à la production, la traduction, etc. Dans notre cas, ce mot peut viser à transformer les données qui permettent de détecter et corriger des éléments non grammaticaux en générant des phrases à partir de mots. Le mot « automatique » signifie avoir la capacité de détecter ou de corriger des erreurs, sans la participation humaine, d'une manière indépendante, et d'imposer des contraintes sérieuses afin d'effectuer les calculs correspondants.

Les données linguistiques doivent être comprises de manière totalement explicite, cohérente et opérationnelle. Pour cela, l'utilisation de divers types de formalismes et de techniques informatiques doit être appropriée. Le processus pourrait être automatique entièrement ou seulement partiellement ; l'utilisateur peut avoir le choix entre une correction semi-automatique et purement automatique.

2. L'approche syntaxique et le dictionnaire adopté

3.1. L'APPROCHE SYNTAXIQUE ADOPTÉE

Il existe plusieurs formalismes pour représenter l'analyse d'un texte. Par ailleurs, presque toute la littérature traite deux formats de représentation syntaxique, à savoir la représentation par constituant et la représentation par dépendance.

L'approche syntaxique adoptée dans cette étude s'inspire de la grammaire de la dépendance (GD) fondée par Tesnière [33]. Elle est basée sur la logique du prédicat. Cette étude propose un point de vue linguistique de la grammaire arabe traditionnelle qui a été interprétée dans une description logique, qui prendra finalement un formalisme informatique dans une ontologie de domaine.

Pour atteindre cet objectif, il semble nécessaire de traduire les données grammaticales en termes de structure sous la forme d'un quadruplet (GC, R, OP, Ax). Tels que GC désigne la catégorie grammaticale, R un ensemble de relations grammaticales, l'opération OP et Ax un ensemble d'axiomes.

La phrase (S) dans le cadre où nous nous sommes situés est définie comme un réseau syntaxique qui peut être exprimé par la formule suivante :

$$(\forall x, y \in GC) / S = \bigwedge_i^n Ri\ (x, y) \qquad (1)$$

Tels que x, y représentent des mots et R une relation grammaticale.

À titre d'exemple, la relation « Sujet » est établie entre un verbe et un nom, tel que :

$$(\exists x \in Verbe, \forall y \in Nom) / Sujet\ (x, y) \qquad (2)$$

Nous constatons que l'analyse de dépendance permet un traitement automatique facile, facilitant ainsi l'apprentissage supervisé et l'application des algorithmes classiques [137]. En effet, les arbres de dépendance représentent une manière hiérarchique de structurer les informations où chaque mot est lié à un mot-clé dont il dépend.

Contrairement à l'analyse syntaxique basée sur les constituants, où le nombre de propositions qui représente la phrase ne peut pas être prédite à l'avance, chaque analyse générée contient un nombre fixe d'éléments de représentation. Par conséquent, sachant que chaque mot n'a qu'un en-tête, l'analyse de dépendance contient exactement un élément de représentation pour chaque mot.

3.2. LE DICTIONNAIRE UTILISÉ

L'organisation du dictionnaire est une étape essentielle de tout le processus de génération de phrases, nous avons organisé notre dictionnaire sous forme de tables dans la base de données contenant environ 6000 racines.

Nous avons choisi Arramooz Alwaseet [138] qui est un dictionnaire arabe open source. Il est généré à partir d'Ayaspell (spellchecker arabe) ; ses données sont collectées manuellement.

Ce dictionnaire se compose de trois parties :

- Mots vides (Stop words)
- Noms (voir tableau 5)
- Verbes (voir tableau 6)

Le dictionnaire contient plus de 50 000 mots, qui couvrent plus de 10 000 verbes et 40 000 noms, ainsi que des dizaines de particules et d'outils syntaxiques.

Tableau 5 Description de la table « Nouns »

Champs	**Descriptions**
vocalized	vocalized word
unvocalized	unvocalized word
wordtype	word type (Noun of Subject, noun of object, …)
Root	word root
feminable	the word accepts Teh_marbuta
defined	the word is defined or not
gender	the word gender
number	the word is single, dual or plural
Single	the single form of the word
dualable	accept dual suffix
feminine	the feminine form of the word
masculine	the masculine form of the word
masculin_plural	accept masculine plural
feminin_plural	accept feminine plural
broken_plural	the irregular plural if exists
mamnou3_sarf	Doesn't accept tanwin
k_suffix	accept Kaf suffix
…	…

Tableau 6 Description de la table « Verbs »

Champs	Descriptions
vocalized	vocalized word
unvocalized	unvocalized word
root	root of the verb
future type	The future mark, used only for trilateral verbs
triliteral	the verb is triliteral (3 letters) or not
transitive	transitive or not
double_trans	has double transitivity for two objects
think_trans	the verb is transitive to human
unthink_trans	the verb is transitive to unhuman being
reflexive_trans	pronominal verb
past	can be conjugated in past tense
future	can be conjugated in present and future tense
passive	can be conjugated in passive voice
…	…

3.3. L'ONTOLOGIE DE DOMAINE UTILISÉE

Le Web sémantique intègre de nouvelles pratiques dans l'organisation du contenu Web et une nouvelle infrastructure permettant aux agents logiciels d'aider efficacement les utilisateurs d'Internet à accéder aux sources et services d'information. Il s'agit d'arriver à un site Web intelligent, où les informations ne seraient pas simplement stockées, mais comprises par les ordinateurs afin d'apporter des réponses pertinentes à l'utilisateur.

XML permet d'indiquer l'organisation logique du contenu d'un document mais n'assure pas la sémantisation de l'information. L'ontologie consiste à annoter ces informations afin de les doter d'une signification interprétable par l'ordinateur. C'est précisément le rôle des couches RDF et RDF-S dans l'architecture du Web sémantique.

OWL est une extension de RDF basé sur le RDF Schemas. Il définit un vocabulaire riche pour décrire les ontologies. Le langage OWL peut être défini en

trois sous-langages, en fonction du niveau d'expression que nous souhaitons exprimer (comme nous avons expliqué dans la section 5.4.6. Il existe trois niveaux dans OWL : OWL-Lite, OWL-DL et OWL-Full).

Pour atteindre notre objectif, nous avons construit une grammaire arabe basée sur le langage de la théorie des ensembles, qui peut être utilisée pour définir presque tous les objets mathématiques. Nous avons emprunté certains de ces principes théoriques afin de construire la grammaire arabe. Nous avons choisi le langage d'ontologie de domaine (OWL-DL), qui décrit un domaine particulier (la syntaxe arabe) en définissant les classes et les relations (propriétés) en raison de la convergence remarquable entre ce type d'ontologie et la théorie mathématique des ensembles.

L'ontologie de la syntaxe arabe (OSA) [139] est un modèle de données représentatives d'un ensemble de concepts relevant du domaine de la syntaxe arabe, ainsi que des relations entre ces concepts.

Nous pouvons représenter l'OSA par un graphe régi par des axiomes dont les nœuds sont des concepts ou des classes et dont les arcs indiquent leurs propriétés :

$$OSA = \{C, R\} \quad (3)$$

Tels que C représente les concepts et R les relations.

L’objectif principal de l’OSA est de fournir aux agents logiciels une intelligence linguistique artificielle permettant de raisonner sur des objets de structure syntaxique arabe, permettant ainsi aux machines de « comprendre » les constituants de la phrase arabe.

Les concepts de l'OSA représentent des catégories grammaticales ou linguistiques, tandis que leurs relations ontologiques font référence aux divers liens syntaxiques existant entre ces catégories grammaticales. Chaque classe grammaticale constitue un ensemble, en d’autres termes, une catégorie syntaxique bien définie, dont les éléments sont liés à des fonctions grammaticales (Fig. 11).

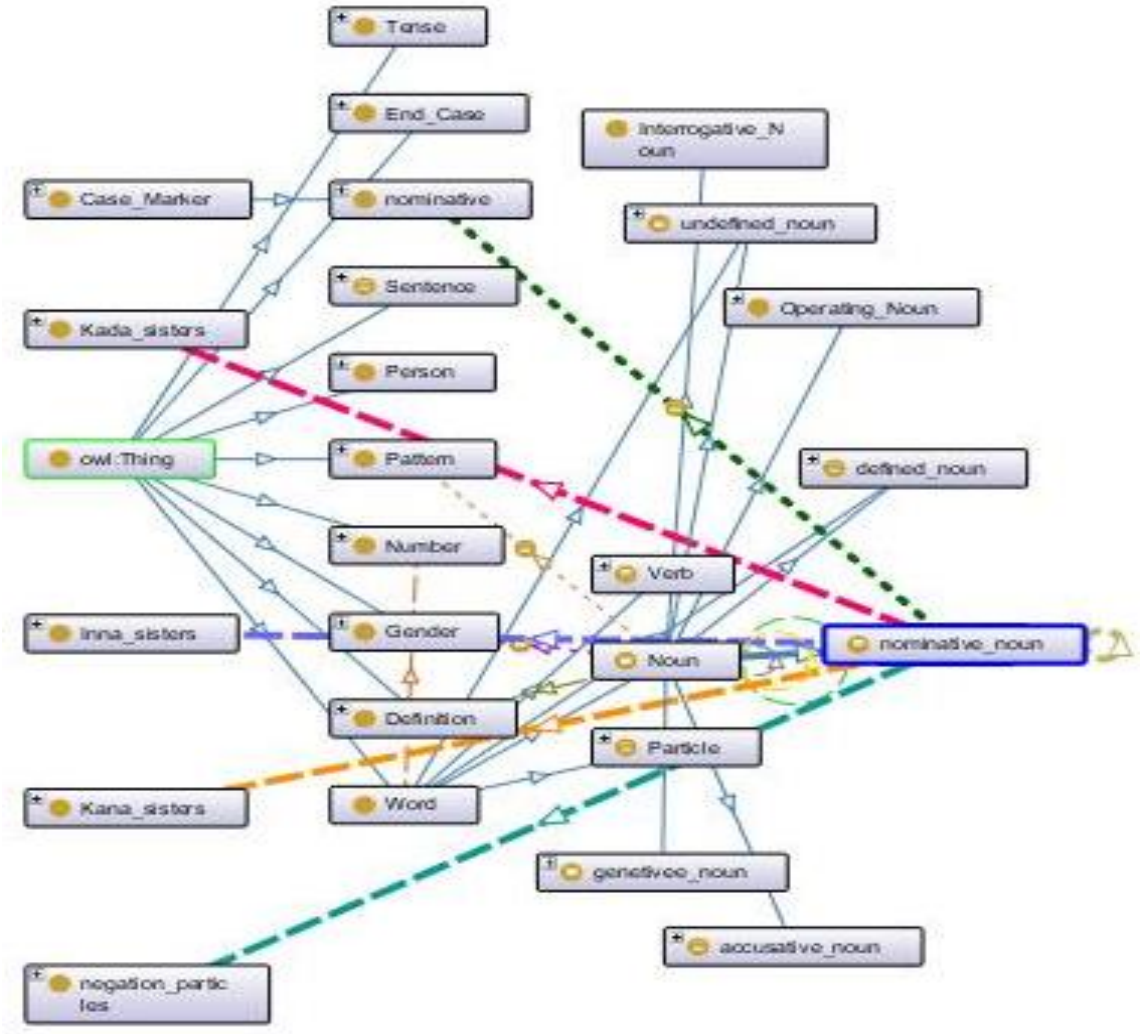

Figure 11 graphe des classes et des propriétés

Les concepts grammaticaux, organisés sous forme d'arborescence hiérarchique, représentent des classes au sens ontologique du terme, alors que les relations grammaticales hiérarchiques représentent des propriétés. Nous pouvons distinguer deux types de relations, la relation de dépendance "علاقة عاملية" reliant les mots et les phrases, ainsi que la relation fonctionnelle "علاقة وظيفية" qui attribue aux mots et aux phrases des traits fonctionnels.

L'ontologie est créée et implémentée à l'aide de l'outil Protégé [57] afin de modifier les règles de grammaire arabe avec OWL2 recommandé par le W3C en ajoutant plusieurs aspects spécifiques à OWL2, tels que les connectives booléennes, les sous-chaînes de propriétés et les restrictions de cardinalité qualifiées, etc.

Notre ontologie est organisée par un ensemble de règles de grammaire arabe afin de fournir des mécanismes permettant de décrire des groupes de ressources similaires (classes) et les relations entre ces ressources (propriétés) (Fig. 12).

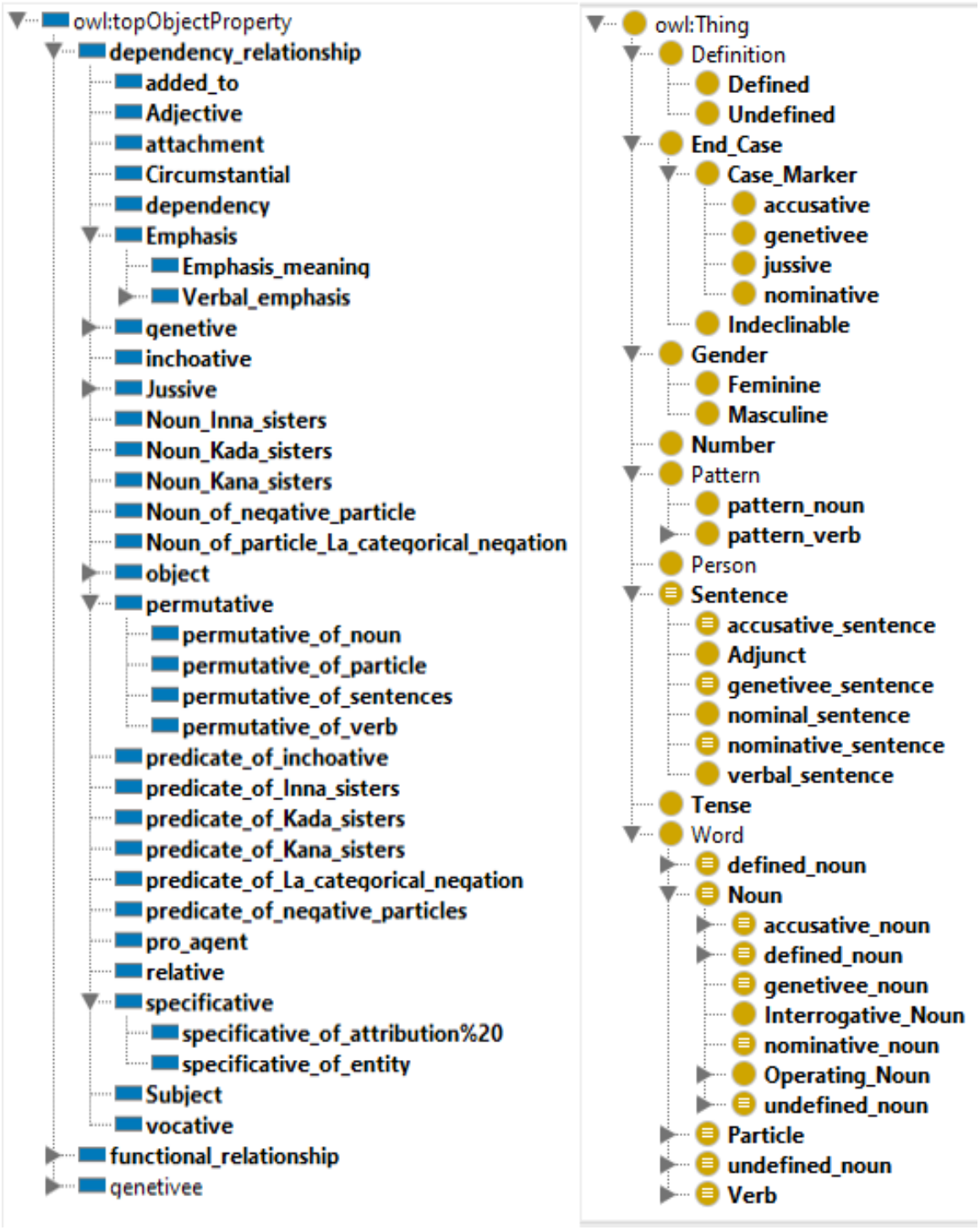

Figure 12 les propriétés et les classes

Un système d'héritage permet à chaque entité ontologique d'hériter des propriétés et des axiomes descriptifs dans lesquels elle est incluse pour définir correctement les classes grammaticales du langage OWL 2, en fonction de la description logique qui nous fournit les moyens appropriés. Ainsi, la classe “Defined_noun” hérite ses traits syntaxiques et fonctionnels de la classe “Noun”.

Par exemple, nous avons défini « nom nominatif - اسم_مرفوع » par l'expression dans la (Fig. 13).

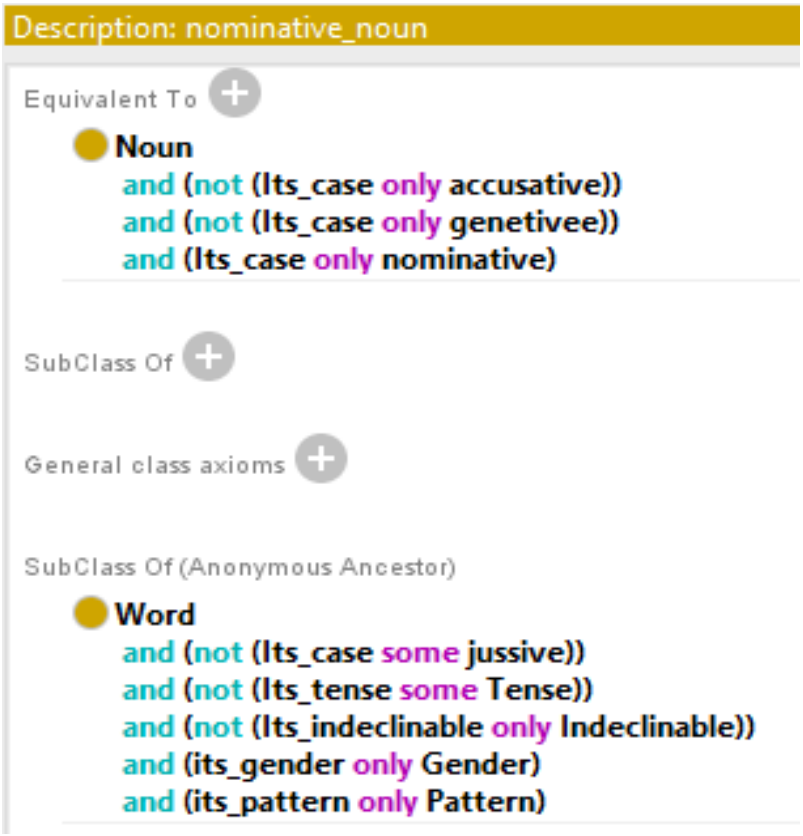

Figure 13 Description du "nominative noun - اسم_مرفوع"

Tels que "AND" représente l'union et "NOT" la négation logique. Le mot-clé "SOME" dans l'exemple (Fig. 13) signifie que la propriété de désinence casuelle prend certaines (∃) de ses valeurs à partir du signe jussif "علامة_الجزم", tandis que "ONLY" signifie que pour toutes (∀) les valeurs attribuées sont prises à partir du signe nominatif "علامة_الرفع".

Pour illustrer clairement ces propriétés logiques. Les tableaux 7-9 montrent respectivement la description logique et informatique dans « Protégé » pour le nom, le verbe et la particule.

Tableau 7 Description symbolique et informatique du « Nom »

Description symbolique	Description informatique en Protégé	Interprétation
$\neg(\exists x \in \text{Noun}\|\text{Its\_case}(x) = jussive\_marker)$ $\rightarrow(\forall x \in \text{Noun}\|\neg(\text{Its\_case}(x) = jussive\_marker))$	not(Its_case some jussive_marker)	Do not accept the jussive case "الجزم"
$\neg(\exists x \in \text{Noun}\|\text{Its\_tense}(x) = Tense)$ $\rightarrow\forall x \in \text{Noun}\|\neg(\text{Its\_tense}(x) = Tense))$	not (Its_tense some Tense)	Do not accept the tense (past, present, future)
$(\forall x \in \text{Noun} \mid (\text{Its\_gender}(x) = Gender))$	(Its_gender only Gender)	Accepts a gender
$(\forall x \in \text{Noun} \mid (\text{Its\_pattern}(x) = Pattern))$	(Its_pattern only Pattern)	Accepts pattern

Tableau 8 Description symbolique et informatique du « Verb »

Description symbolique	Description informatique en Protégé	Interprétation
$\neg(\exists x \in \text{Verb}\|\text{Its\_case}(x) = \text{Genetive\_marker})$ $\rightarrow(\forall x \in \text{Verb}\|\neg(\text{Its\_case}(x) = \text{Genetive\_marker}))$	not (Its_case some Genetive_marker)	Do not accept the genitive case "Genetive_marker"
$\neg(\exists x \in \text{Verb}\|\text{Its\_gender}(x) = Gender)$ $\rightarrow\forall x \in \text{Verb}\|\neg(\text{Its\_gender}(x) = Gender))$	not (Its_gender some Gender)	Do not accept the gender
$(\forall x \in \text{Verb} \mid (\text{Its\_pattern}(x) = Pattern))$	(Its_tense some Tense)	Accepts tense
$(\forall x \in \text{Verb} \mid (\text{Its\_tense}(x) = Tense))$	(Its_pattern only Pattern)	Accepts pattern

Tableau 9 Description symbolique et informatique du « Particle »

Description symbolique	Description informatique en Protégé	Interprétation
$\neg(\exists x \in \text{Particle}\|\text{Its\_pattern}(x) = Pattern)$ $\rightarrow(\forall x \in \text{Particle}\|\neg(\text{Its\_pattern}(x) = Pattern))$	not (Its_pattern some Pattern)	Do not accept the pattern
$\neg(\exists x \in \text{Particle}\|\text{Its\_tense}(x) = Tense)$ $\rightarrow\forall x \in \text{Particle}\|\neg(\text{Its\_tense}(x) = Tense))$	not (Its_tense some Tense)	Do not accept the tense
$\neg(\exists x \in \text{Particle}\|\text{Its\_gender}(x) = Gender)$ $\rightarrow\forall x \in \text{Particle}\|\neg(\text{Its\_gender}(x) = Gender))$	not (Its_gender some Gender)	Do not accept the gender
$(\forall x \in \text{Particle}\| (\text{Its\_indeclinable}(x) = Indeclinable))$	(Its_indeclinable only Indeclinable)	Accepts static case-ending

L'exploitation de l'ontologie OSA est assurée par un système de requêtes défini par SPARQL. De manière similaire aux requêtes SQL, l'utilisateur peut accéder à la base de données OSA via ce langage de requête RDF.

L'exemple suivant pourrait illustrer comment interroger OSA : supposons que nous voulons déterminer les relations syntaxiques dont l'élément gouverné "المعمول" est un « nom nominatif - اسم_مرفوع », tel que :

R(x, اسم_مرفوع) (4)

La figure 14 montre les relations possibles vérifiant (4) et leurs gouverneurs

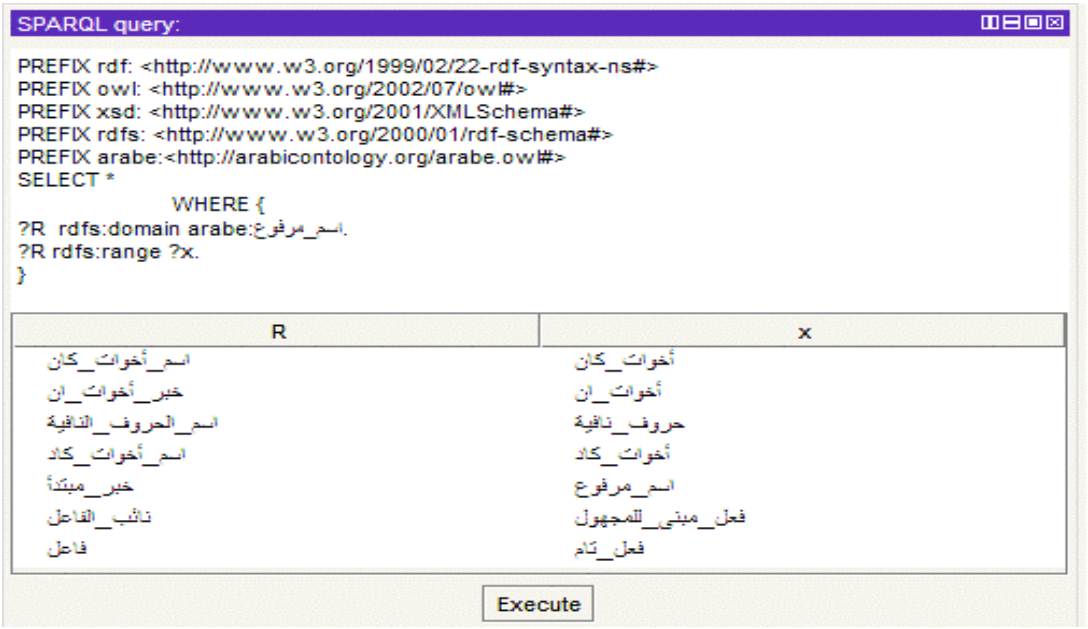

Figure 14 Requête SPARQL de R(x, اسم_مرفوع)

Si nous voulons restreindre le champ du gouverneur “x”, car il s'agit d'une particule “حرف” :

$$R(\text{x,اسم-مرفوع}) \wedge (x)\text{حرف} \qquad (5)$$

Le résultat obtenu est (voir Fig. 15) :

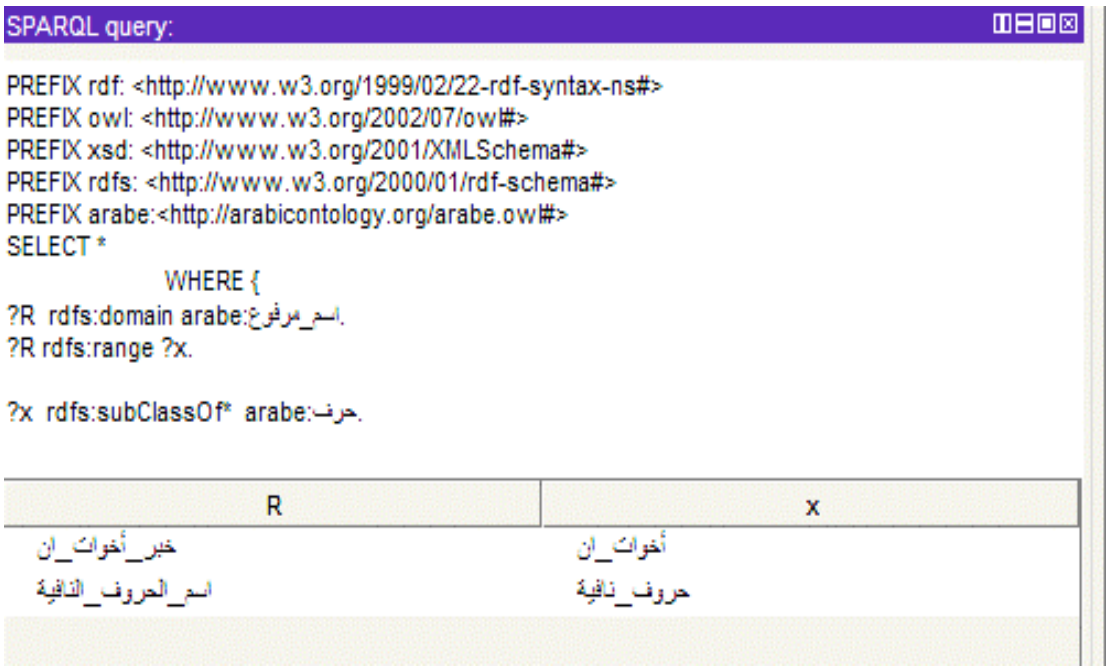

Figure 15 Requête SPARQL de R(x, اسم_مرفوع) ∧ (x) حرف

3. La méthode adoptée

La méthode de correction des erreurs syntaxiques adoptée regroupe les informations de l'analyseur morphologique Al-Khalil [65] et du dictionnaire susmentionné en réunissant des informations sur les racines, les noms, les verbes, etc., ainsi que sur certaines règles morphologiques. Cette méthode est divisée en trois phases (voir figure 16).

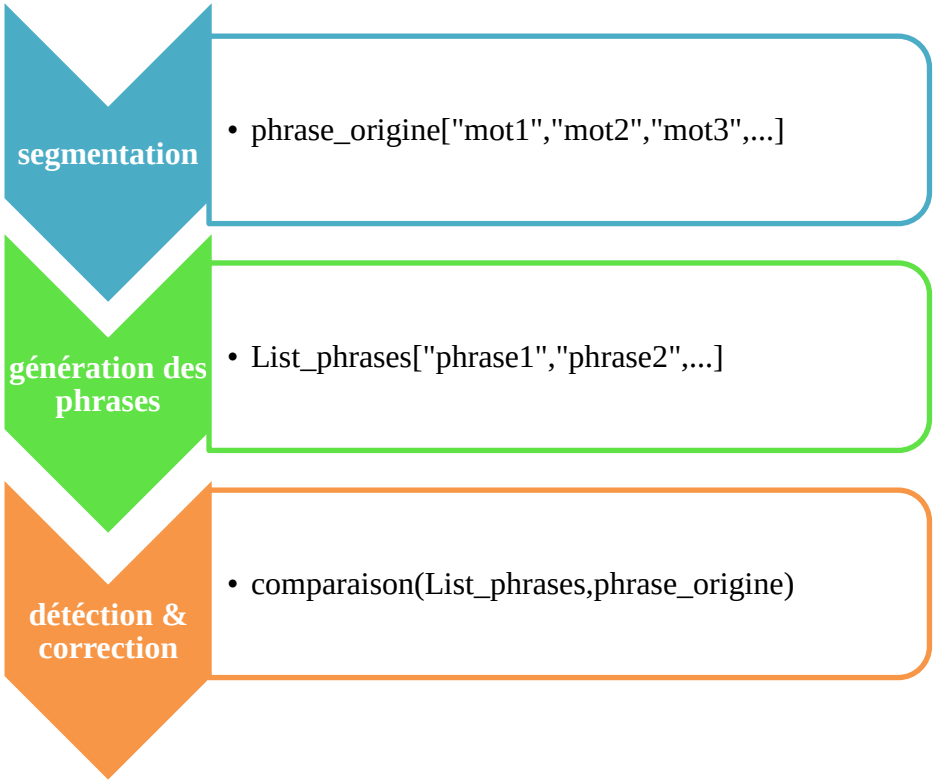

Figure 16 les trois phases de la méthode adoptée

4.1. LA PHASE DE SEGMENTATION

Le problème de segmentation en phrases pour langue arabe est compliqué. En effet, l'arabe n'utilise ni caractères majuscules ni ponctuation régulière, ce qui rend les méthodes classiques de segmentation inappropriées à cette langue. De plus, l'agglutination des mots est une autre particularité de l'Arabe qui rend la segmentation encore plus difficile à réaliser [140].

Nous avons adopté la segmentation en deux étapes : premièrement, une segmentation du texte en phrases, et deuxièmement, une segmentation de phrases en mots. La segmentation du texte en mots est réalisée par la plate-forme "Software

Architecture for Arabian Processing—SAFAR" [141], qui contient un segmenteur de texte arabe basé sur l'exploration contextuelle des signes de ponctuation, des mots connecteurs jouant le rôle de séparateurs de phrases ainsi que ceux de certaines particules, telles que les conjonctions de coordination. Le processeur de phrases est une application qui montre comment diviser un texte en phrases, puis normaliser les phrases et les translittérer.

La segmentation de la phrase en mots est basée sur la détection d'espaces, de signes de ponctuation et de certains caractères spéciaux. La plate-forme SAFAR propose plusieurs méthodes permettant la tokénisation, définie comme le processus de division d'un texte en éléments (mots).

La segmentation de la phrase peut être vue comme une opération dont l'argument est la phrase et le résultat associé est un ensemble de mots distincts {w0, w1… wn} (voir Fig. 17).

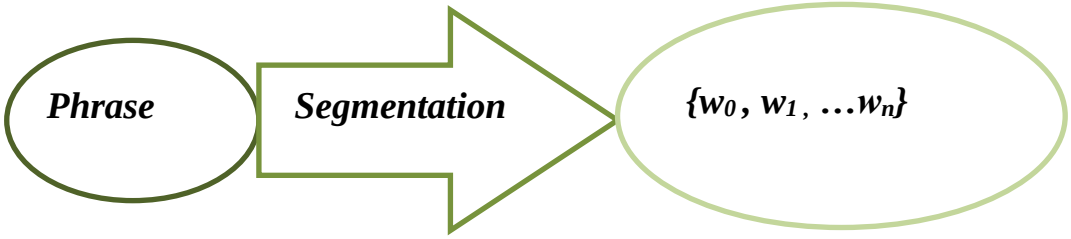

Figure 17 la phase de segmentation

4.2. LA PHASE DE GÉNÉRATION DE PHRASES :

Le processus d'élaboration d'une phrase se fait en deux étapes :

4.2.1. ÉTAPE 1 : LA CATÉGORISATION

La catégorisation associe un ensemble de traits syntaxiques (Nombre, Genre, Personne…) à chaque mot obtenu dans de la phase de segmentation (Fig. 18).

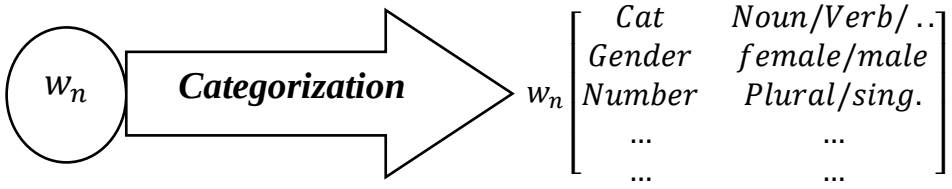

Figure 18 L'étape de catégorisation

La catégorisation nous fournit deux types d'informations ; le premier est relatif aux traits lexicaux (TL) auxquels appartient le mot et le second spécifie les traits fonctionnels (TF) du mot concerné. Sachant que chaque catégorie lexicale est donnée avec des traits distincts ; s'il s'agit d'un verbe, le mot ne peut prendre que

les traits relatifs aux caractéristiques verbales (temps, transitivité, forme grammaticale, etc.), dans le cas où le mot est un nom, le processus de catégorisation associe le mot considéré aux caractéristiques nominales (nombre, genre, etc.).

Ces informations pertinentes forment deux ensembles disjoints

$$CG = TF \cup TL$$

Avec les résultats d'Al-Khalil, notre système ne peut toujours pas identifier les différentes formes de mots arabes. Pour cette raison, nous avons utilisé le dictionnaire pour nous aider à reconnaître les différentes formes d'un mot. Par exemple, pour récupérer le singulier "le garçon - الولد" du pluriel "les garçons - الاولاد", nous pouvons utiliser une requête pour obtenir le singulier dans le dictionnaire (Fig. 19).

```
<noun id='100000'>
 <vocalized>أَوْلاَدٌ</vocalized>
 <unvocalized>أولاد</unvocalized>
 <root>ولد</root>
 <number>جمع تكسير</number>     <!--broken plural-->
 <single>وَلَدٌ</single>
  ...
</noun>
```

Figure 19 Le mot "ولد" dans le dictionnaire

Le nombre de résultats fournis est très important, cependant, il s'avère que l'analyseur morphologique Al-Khalil produit beaucoup des résultats alternatifs pour notre système, c'est pourquoi nous devons les examiner afin d'obtenir les informations nécessaires que nous souhaitons utiliser en supprimant les doublons.

Nous extrayons d'abord la catégorie du mot concerné, à savoir : isVerb (), isNoun (), isParticle ().

- S'il s'agit d'un verbe, nous n'utilisons que les traits syntaxiques suivants : Type, Transitive, Impartial, Prefix, Suffix et Tense.
- S'il s'agit d'un nom, nous n'utilisons que les traits syntaxiques suivants : Type, Gender, Number, Prefix, Suffix et Definiteness.
- S'il s'agit d'une particule, nous utilisons le trait syntaxique : Type.

Le tableau 10 montre un exemple de catégorisation du mot "كتب" « "a écris" / "a été écris" / "livres" » :

Tableau 10 La catégorisation du mot "كتب"

Les catégories	Les traits syntaxiques	Output 1	Output 2
Verbe	*Type*	past active verb	past passive verb
	Transitive	yes	yes
	Prefix	#	#
	Suffix	#	#
Nom	*Type*	Non-derivative noun	Verbal noun
	Gender	masculine	feminine
	Number	plural	singular
	Prefix	#	كَ
	Suffix	#	#
Particule	*Type*	#	#

4.2.2. ÉTAPE 2 : LE FUSIONNEMENT

Après avoir déterminé toutes les informations syntaxiques nécessaires, nous procédons à l'élaboration des phrases.

Les mots sont combinés par l'opération de fusion pour former un ensemble de paires orientées (x, y) conformes aux axiomes qui contrôlent la formation des couples décrits ci-dessus.

Ensuite, l'ensemble de ces couples forme une phrase simple de sorte qu'elle prenne la forme (7) :

$$\bigwedge_1^n R(x, y) \qquad (7)$$

La formation de couples est licenciée par des schémas préétablis décrits par une ontologie de domaine (Fig. 20).

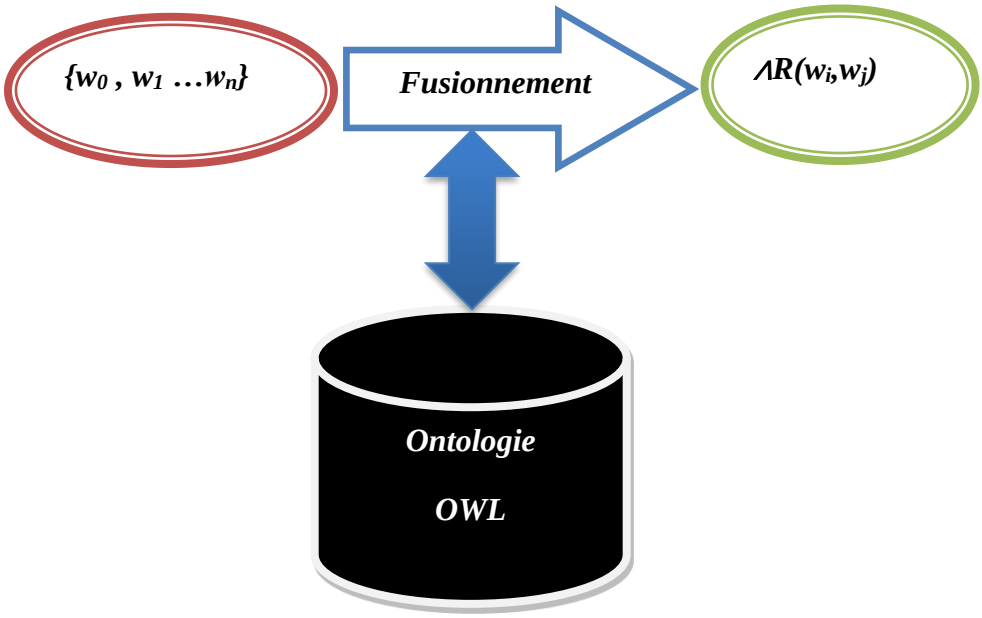

Figure 20 Étape de fusionnement

Exemple

Nous illustrerons ceci par la phrase S :

" أكل أحمدُ التفاحةَ- Ahmed ate the apple "

$$S = Subj\left(\text{أكل،أحمد}\right) \wedge obj\left(\text{أكل،التفاحة}\right) \wedge \text{def}\left(\text{ال،تفاحة}\right)$$

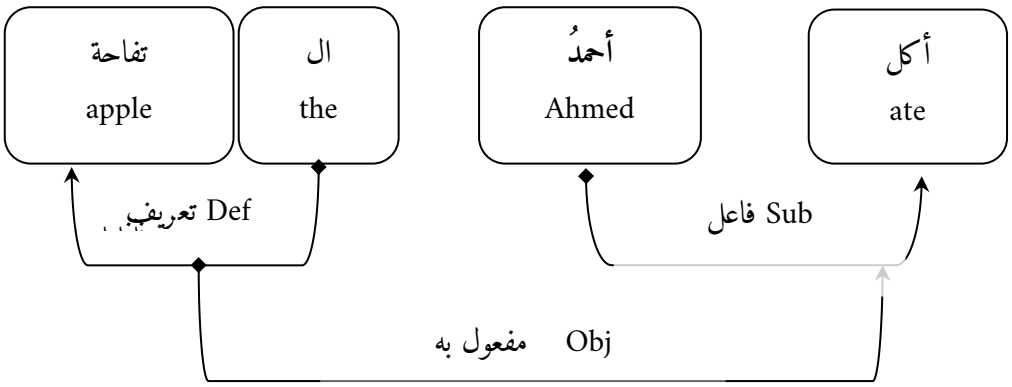

Figure 21 Exemple de fusionnement

Nous pouvons distinguer deux types d'opérations de fusions ; l'un opère sur l'axe linéaire (Fig. 21) pour composer des relations grammaticales, tandis que l'autre opère sur l'axe vertical en spécifiant les aspects fonctionnels des mots (type, temps, transitivité pour les verbes, etc.).

$$\text{خرج} \begin{bmatrix} Cat & Verb \\ Trans & No \\ Tense & Paste \\ \dots & \dots \\ \dots & \dots \end{bmatrix}$$

Figure 22 matrice des informations syntaxiques du mot " خرج"

L'axe fonctionnel sera représenté par une matrice (Fig. 22) contenant les informations syntaxiques associées aux mots.

Après avoir mis à jour la matrice catégorielle par les opérations de catégorisation, les mots peuvent donc se combiner dans l'axe linéaire et prendre leurs positions sous licence telles qu'elles sont établies en grammaire arabe.

Nous avons implémenté notre ontologie par les définitions des relations grammaticales, qui relient les champs grammaticaux ; ces relations grammaticales se caractérisent par un ensemble de propriétés formelles, que nous présenterons brièvement :

a. Les relations grammaticales sont des paires dont les éléments sont soumis à un ordre spécifique. C'est semblable à celui d'une paire de relations mathématiques. De sorte que si l'ordre des deux extrémités de la paire change, la signification de la relation change également, alors nous disons que la paire grammaticale est une relation asymétrique, dans ce cas, les deux paires suivantes ne sont pas égales :

$$\forall x, y \in GC / \quad R(x, y) \longrightarrow \sim R(y, x) \qquad (8)$$

Exemple :

Subject(خرج , الولدُ) ≠ Subject(الولدُ , خرج)

Si vous faites attention à la relation de sujet, vous constaterez qu'elle a une direction spécifique ; il est donc possible de dire que chaque relation a une base de départ et une fin, le point de départ s'appelle "domain" et la fin s'appelle "range" (Fig. 23). Cette loi s'applique à toutes les relations grammaticales sans exception.

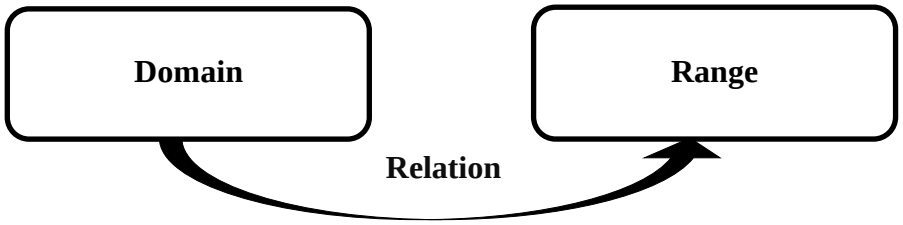

Figure 23 La loi des relations grammaticales dans l'ontologie

b. Les relations grammaticales sont intransitives dans le sens qu'aucun élément de CG est en relation avec un deuxième élément lui-même en relation avec un troisième. La grammaire arabe interdit que le premier élément soit également lié au troisième élément.

$$(\forall x, y, z \in CG)(\forall R) / \sim(R(x, y) \wedge R(y, z) \longrightarrow R(y, z)) \quad (9)$$

$$\sim (\sim(R(x, y) \wedge R(y, z)) \vee R(y, z))$$

$$(R(x, y) \wedge R(y, z)) \wedge \sim R(y, z))$$

Un tel système formel décrit ci-dessus est susceptible de nous fournir des informations sur la façon dont la phrase est élaborée ; la construction d'une phrase nécessite deux types d'informations. D'une part, nous sommes amenés à spécifier les éléments catégoriels indiqués ci-dessus par les lettres x, y, z. Ce type d'information nous est fourni par une base de données lexicales et morphologiques (...). D'autre part, la mise en relation de ces éléments catégoriels nous exige d'énumérer toutes les liaisons possibles reconnues par la grammaire arabe. Ce type d'information nous est fourni par une base de données ontologique.

c. Dans la phrase grammaticale, nous distinguons deux types de relations : la relation principale qui constitue l'essence de la phrase, puis une relation secondaire qui peut être abandonnée sans compromettre le sens général de la phrase.
d. Les relations grammaticales sont irréflexives (ou antiréflexives) car aucun élément catégorique n'est lié à lui-même :

$$(\forall x \in GC)\ (\forall R) /\ \ \sim R(x, x) \quad (10)$$

Exemple :

خرج الولدُ

Subject(الولدُ , الولدُ)

L'ontologie de syntaxe arabe ne définit pas seulement les liens grammaticaux possibles mais impose des contraintes sous la forme d'axiomes. Par exemple, la relation Subject (فاعل) doit être contrôlée par la contrainte suivante :

$$(\exists x \in Verb, \forall y \in Noun) / Subject (x, y) \quad (11)$$

$$\longrightarrow has\_case (y, Nominative)$$

Ce qui postule que tout sujet y de x porte une désinence casuelle nominative (علامة الرفع).

Nous avons adopté une cinquantaine de relations grammaticales ; le tableau 11 illustres certaines de ces relations dans notre ontologie arabe :

Tableau 11 Les relations grammaticales

Relation	Domain	Range	Address
Subject	Nominative noun	Verb/Operating noun	http://arabicontology.org/arabe.owl#فعل
Pro-agent	Nominative noun /Preposition	Verb passive/passive participle	http://arabicontology.org/arabe.owl#نائب_الفاعل
First Object	Accusative noun	Verb/Operating noun	http://arabicontology.org/arabe.owl#مفعول_به
Noun of Kana sisters	Nominative noun	Kana sisters	http://arabicontology.org/arabe.owl#اسم_اخوات_ كان
Predicate of inchoative	Nominative noun	Nominative noun	http://arabicontology.org/arabe.owl#خبر_مبتدأ
Adjective	Noun	Noun	http://arabicontology.org/arabe.owl#نعت
Vocative	Accusative noun	Vocative particle	http://arabicontology.org/arabe.owl#منادى
Its pattern	Noun/Verb	pattern	http://arabicontology.org/arabe.owl#وزنه
Its gender	Noun	Gender	http://arabicontology.org/arabe.owl#جنسه
Its number	Noun	Number	http://arabicontology.org/arabe.owl#عدده
Its tense	Verb	Tense	http://arabicontology.org/arabe.owl#زمنه
Genitive by preposition	Genitive noun	Preposition	http://arabicontology.org/arabe.owl#مجرور_بحرف
Possessive Construction	Genitive noun	Undefined noun	http://arabicontology.org/arabe.owl#مضاف_اليه
Predicate of Inna sisters	Nominative noun	Inna sisters	http://arabicontology.org/arabe.owl#خبر_اخوات_ان
Predicate of Kana sisters	Nominative noun	Kana sisters	http://arabicontology.org/arabe.owl#خبر_اخوات_ كان
Circumstantial	Undefined accusative noun	Verb	http://arabicontology.org/arabe.owl#حال
…	…	…	…

4.3. LA PHASE DE DÉTECTION ET DE CORRECTION DES ERREURS

Cette phase consiste à comparer toutes les phrases syntaxiquement correctes générées par la phase précédente avec la phrase originale ; dans ce cas, nous avons deux possibilités :

- Si le système a trouvé la phrase d'origine dans la liste des phrases générée : dans ce cas, le système passe à la phrase suivante car il considère que la phrase est correcte.
- Si le système n'a pas trouvé la phrase d'origine dans la liste des phrases générée : l'utilisateur peut choisir la manière de correction, c.-à-d. il peut choisir manuellement une phrase parmi les phrases générées à partir des résultats de la phase précédente. Il peut également choisir une correction purement automatique qui propose la phrase correcte la plus probable.

La figure suivante (Fig. 24) montre comment détecter et corriger les erreurs en fonction de la liste des phrases générée :

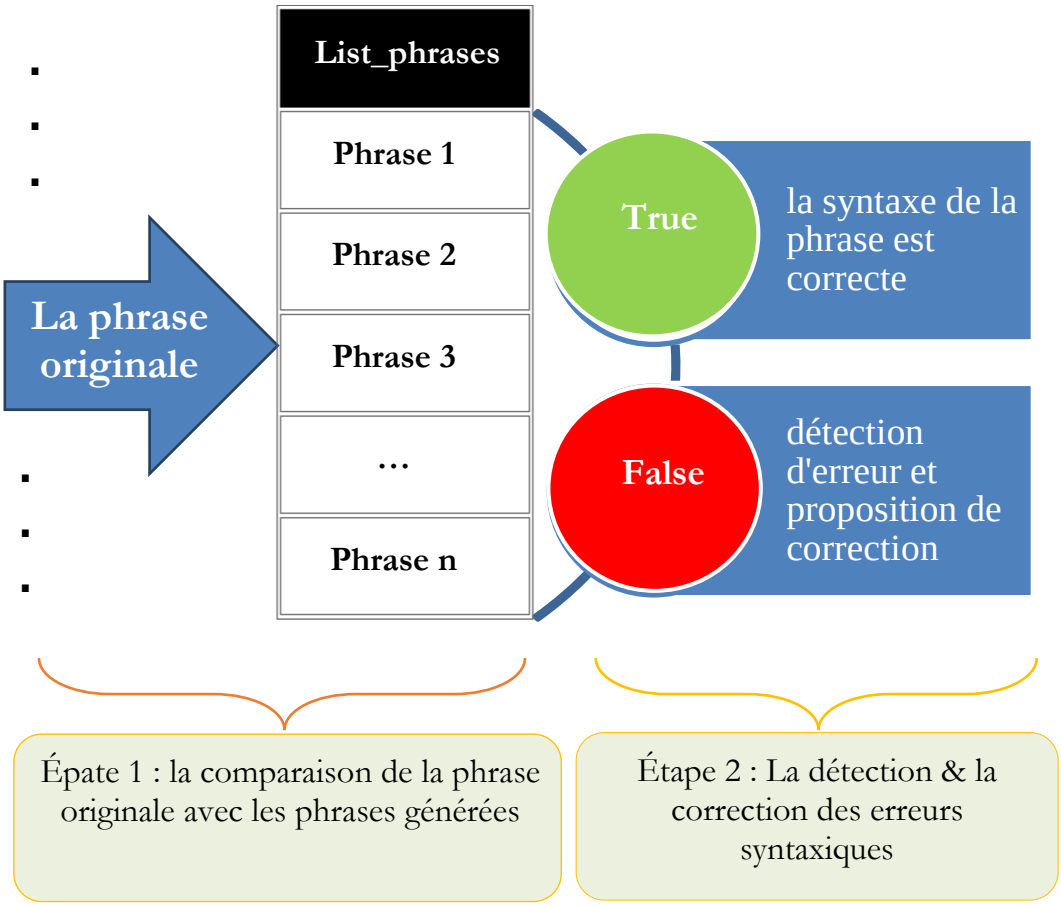

Figure 24 La phase de détection et correction des erreurs

Nous calculons la « distance de Levenshtein » [142] entre la phrase d'origine et les phrases générées afin de permettre une certaine souplesse lors de la comparaison.

La distance de Levenshtein :

La distance de Levenshtein est le nombre minimal d'opérations (prises dans cet ensemble) nécessaires pour transformer A1 en A2. La dérivation corrective

optimale est la suite d'opérations d'éditions utilisées pour calculer la distance de Levenshtein :

$$D\ (A_1\ ;\ A_2) = e_1\ ;\ e_2, \ldots,\ e_n \qquad (12)$$

$$avec\ e_k = (x_i\ ;\ x_j),\ 1 \leq k \leq n \qquad \forall x_i\ ,\ x_j \in \{\sum \cup \{\varepsilon\}\}$$

Un algorithme de programmation dynamique permet de calculer la D (A_1 ; A_2) dans un temps de l'ordre de θ (| A_1 |, | A_2 |), avec | A1 | (Resp. A_2 |) la longueur de A_1 (ou de A_2). Les coûts unitaires peuvent également être attribués à ces opérations comme suit :

$$\omega(x_i\ ,\ x_j) = \begin{cases} 1 & si\ x_i \neq x_j \\ 0 & si\ x_i = x_j \end{cases} \quad \forall x_i\ ,\ x_j \in \{\sum \cup \{\varepsilon\}\} \qquad (13)$$

De ce point de vue, la distance de Levenshtein constitue également le coût minimal de transformation de A_1 en A_2 en fonction des opérations à coûts unitaires.

La fonction "LevenshteinDistance" renvoie un entier, chaque fois qu'il est petit, la phrase proposée doit être à la première place.

4. Exemple

Le but de cet exemple est d'illustrer le lien entre la génération de phrases syntaxiquement correctes et la correction d'erreurs syntaxiques détectées.

Soit la phrase incorrecte suivante :

"رجع المسافرين البعيدون"

« Les voyageurs lointains sont revenus »

Pour corriger les erreurs de syntaxe de cette phrase, nous allons appliquer notre approche en procédant comme suit :

5.1. LA SEGMENTATION

La première étape consiste à segmenter la phrase en mots. Le résultat obtenu est le suivant :

Segmentation (رجع المسافرين البعيدون) = Seg_1(رجع) +Seg_2(المسافرين) + Seg_3(البعيدون)

5.2. LA CATÉGORISATION

Après la segmentation de la phrase en trois unités, cette étape a pour objectif d'associer un ensemble de traits morphosyntaxiques (Nombre, Genre, Personne…) à chaque mot obtenu (Fig. 25).

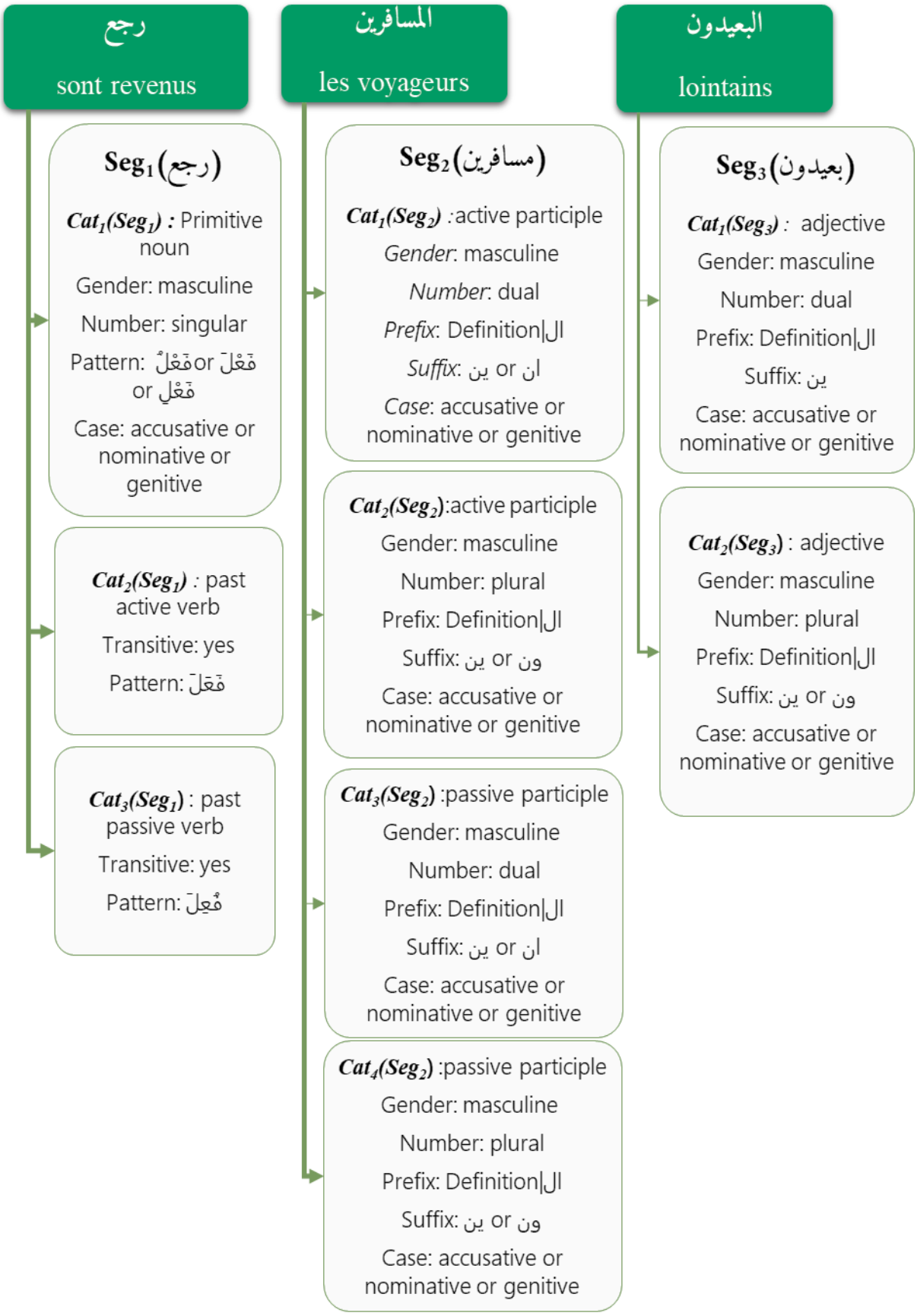

Figure 25 étape de catégorisation

5.3.LE FUSIONNEMENT

Afin de construire une phrase syntaxiquement correcte, il faut rechercher toutes les fusions possibles, nous obtenons alors :

- Mrg_{1_1} =($Cat_1(Seg_1)$ +$Cat_1(Seg_2)$) =**R_{1_1}(Noun$_1$, Noun$_2$)**
- Mrg_{1_2} =($Cat_1(Seg_2)$ +$Cat_1(Seg_3)$) =**R_{1_2}(Noun$_2$, Adj)**
- Mrg_{2_1} =($Cat_2(Seg_1)$ +$Cat_1(Seg_2)$) =**R_{2_1}(Verb, Noun)**
- Mrg_{2_2} =($Cat_1(Seg_1)$ +$Cat_1(Seg_3)$) =**R_{2_2}(Noun, Adj)**

REQUÊTE SPARQL POUR TROUVER LA 1ÈRE RELATION R_{1_1} :

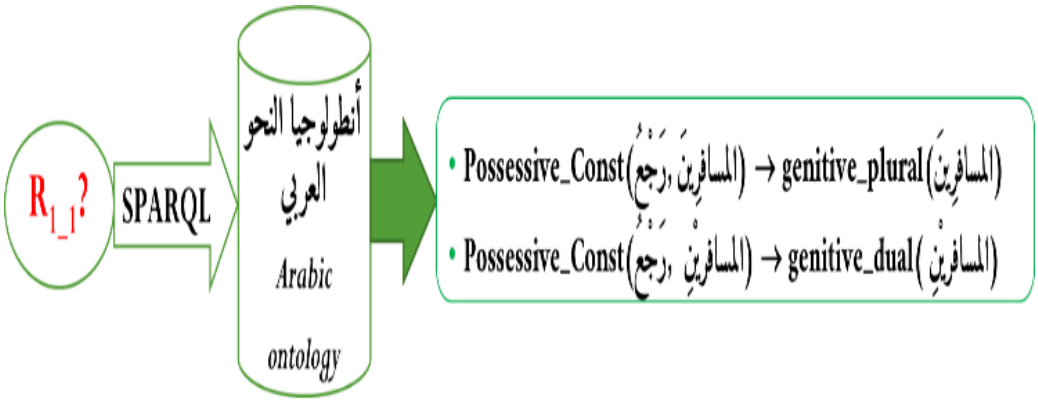

Figure 26 La requête SPARQL pour R_{1_1}

REQUÊTE SPARQL POUR TROUVER LA RELATION R_{1_2} :

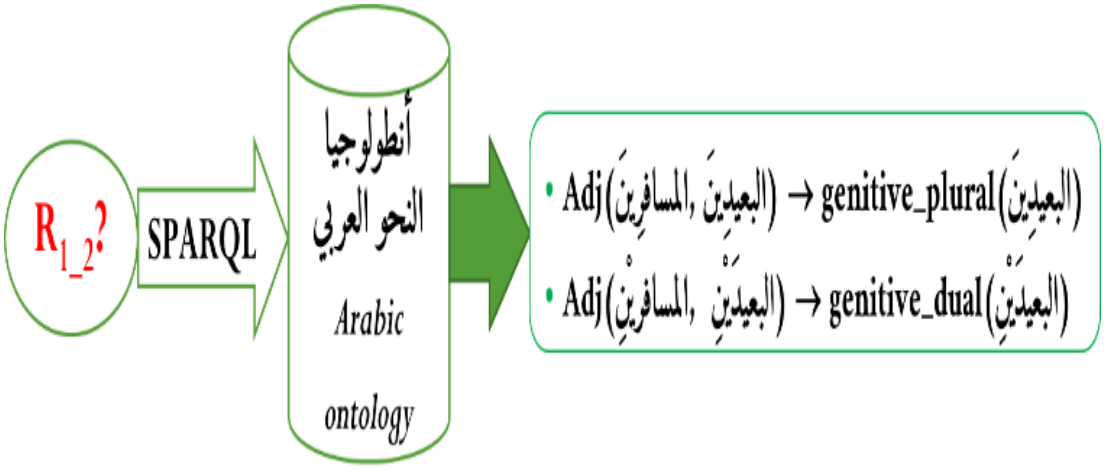

Figure 27 La requête SPARQL pour R_{1_2}

REQUÊTE SPARQL POUR TROUVER LA RELATION R_{2_1} :

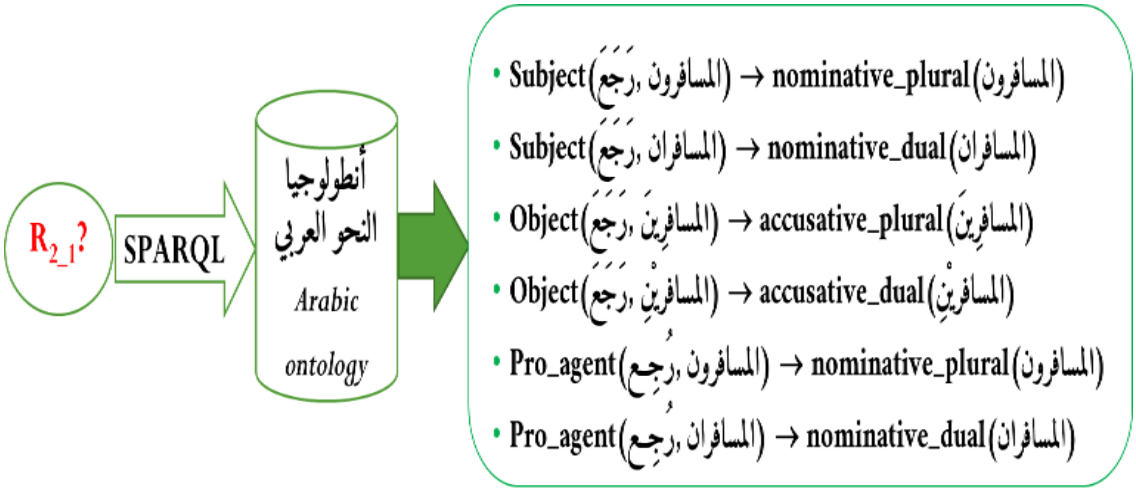

Figure 28 La requête SPARQL pour R_{2_1}

REQUÊTE SPARQL POUR TROUVER LA RELATION R_{2_2} :

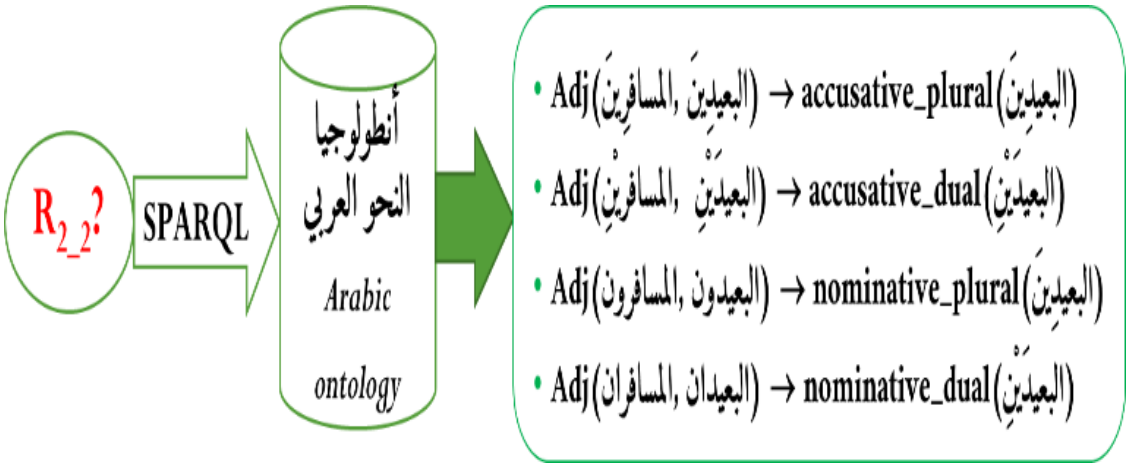

Figure 29 La requête SPARQL pour R_{2_2}

À la suite de cette phase, nous obtenons huit phrases syntaxiquement correctes, à savoir :

- رَجَعَ المسافرِينَ البعيدِينَ
- رَجَعَ المسافرَيْنِ البعيدَيْنِ
- رَجَعَ المسافرون البعيدون
- رجع المسافران البعيدان
- رجع المسافرِينَ البعيدِينَ
- رجع المسافرَيْنِ البعيدَيْنِ
- رُجِع المسافرون البعيدون
- رُجِع المسافران البعيدان

Afin d'améliorer notre système, nous pouvons donner à l'utilisateur le choix de prendre en compte les signes diacritiques ou non, car presque tous les textes arabes sont non-voyellés, à l'exception des livres religieux et de certains manuels scolaires ; le système normalise ces phrases en supprimant les marques diacritiques.

Le résultat devient alors :

- رجع المسافرين البعيدين
- رجع المسافرون البعيدون
- رجع المسافران البعيدان

5.4. DÉTECTION ET CORRECTION DES ERREURS

Nous comparons enfin les trois phrases avec la phrase originale :

```
for (int i = 0; i < phrasesList.size(); i++)
  {
    if( LevenshteinDistance.computeLevenshteinDistance
    (phrasesList.get(i), phrase_original)==0)

         estCorrecte=1;
    else

         phrases_proposeList.add(phrasesList.get(i));

  }
```

Le résultat obtenu :

```
Les phrases proposées sont :

                                رجع المسافرين البعيدين
                                رجع المسافرون البعيدون
                                رجع المسافران البعيدان
```

5. Évaluation et discussions

Dans cette section, nous présentons les résultats des évaluations effectuées sur les phrases arabes. Afin de valider notre approche, nous devons l'évaluer en utilisant des métriques de précision et de rappel sur les informations syntaxiques combinées. Cependant, il n'y a pas de bons corpus contenants des annotations sur plusieurs niveaux de la grammaire arabe, ce qui conduit tout d'abord à l'annotation d'un nouveau corpus de référence contenant 360 phrases. Sur les 360 phrases arabes, il y avait 30 phrases syntaxiquement correctes et 330 phrases non grammaticales, regroupant plusieurs types d'erreurs grammaticales, à savoir :

- 200 erreurs d'accord (genre, nombre, singulier, dual ou pluriel…).
- 100 erreurs de désinence casuelle (nominatif, accusatif ou génitif).
- 30 erreurs d'article défini par "ال".

Les corpus d'erreurs grammaticales ne sont pas faciles à trouver. Certes, ils n'existent pas en langue arabe pour le moment. En effet, nous avons annoté manuellement 330 phrases incorrectes grammaticalement tirées des écrits des apprenants, avec trois classes d'erreurs : erreurs d'accord, erreurs de désinence casuelle et erreurs liées à l'utilisation d'articles définis et indéfinis. Nous avons rassemblé et annoté ce groupe de phrases. Dans cette étude, 30 phrases correctes ont été incluses pour voir si le système est capable de détecter sa validité grammaticale. Ces classes d'erreurs sont présentées avec des exemples à l'annexe A.

L'évaluation de notre système utilise les deux mesures communes de précision et de rappel, ainsi que de F-mesure. Les formules sont les suivantes :

$$Recall = \frac{Number\ of\ errors\ correctly\ detected}{Total\ number\ of\ introduced\ errors} \quad (14)$$

$$Precision = \frac{Number\ of\ errors\ correctly\ detected}{Total\ number\ of\ detections} \quad (15)$$

$$F - measure = 2 * \frac{Precision * Recall}{(Precision + Recall)} \quad (16)$$

Le tableau 12 résume les résultats obtenus :

Tableau 12 Résultats de la détection des erreurs syntaxiques

Syntactic error	Precision	Recall (rappel)	F-measure
Agreement	96,75%	89,5%	92,98%
Case endings	90,42%	85%	87,62%
Definite article "ال"	88,88%	80%	84,20%
Total	92,01%	84,83%	88,27%

La longueur moyenne d'une phrase était de sept mots, et la phrase la plus longue avait onze mots. Notre système comprend environ 200 règles de grammaire.

La complexité de l'approche est généralement proportionnelle à la taille de la phrase et au nombre de sorties de l'analyseur morphologique Al-Khalil pour chaque mot extrait, ainsi qu'au nombre de règles grammaticales utilisées dans la phase de génération de phrases. Il peut être intégré à des applications de TAL de niveau supérieur car il est développé avec le langage de programmation open source Java. De plus, les résultats des phrases correctes peuvent être obtenus via des services Web, des bibliothèques et des sorties XML.

Les résultats de notre approche donnés dans le tableau 11 montrent une précision de 92% et un rappel de 84% ou plus, en moyenne, ce qui est un bon niveau pour ce type de tâche « détection d'erreurs ». Il convient de noter également le haut niveau de précision qui caractérise un niveau de fiabilité très important.

Cette propriété de détection est importante dans ce cas car si le système trouve que la phrase contient une erreur, il passera automatiquement à la phase suivante pour générer les phrases correctes en fonction des mots extraits de la phase précédente. Le rappel pourrait être amélioré en étant plus complet dans les listes d'entités constituées.

Si nous considérons maintenant la métrique de F-mesure (16), qui est un meilleur indicateur de synthèse, nous trouvons que notre méthode fonctionne bien (88,27%). Nous avons également évalué un ensemble de 30 phrases correctes afin de tester le système avec des phrases grammaticalement correctes ; notre système a considéré que 27 phrases sont correctes et 3 sont incorrectes, ce qui donne à notre système un autre avantage et une mise en œuvre réussie en utilisant cette approche.

Nous pouvons voir que certaines relations sont perdues en raison de l'ambiguïté des règles syntaxiques. Nous pouvons donc introduire l'approche basée sur la traduction automatique statistique [113] en fonction de nos informations linguistiques issues de l'ontologie et de l'analyseur morphologique afin de régénérer des phrases correctes.

Il serait intéressant de comparer notre approche avec les autres. Cependant, comme nous l'avons expliqué à la section 4 dans le chapitre III, il n'existe pas de système permettant de corriger les erreurs de syntaxe en arabe, de même qu'il n'existe pas de corpus contenant ces informations à des fins de tests. De plus, il n'est pas possible de faire une comparaison avec des recherches connexes pour d'autres langues, bien que cela soit très difficile car les conditions expérimentales ne sont pas les mêmes.

Les résultats sont satisfaisants car la phase de détection d'erreur syntaxique autorise un rendement élevé de précision et des informations syntaxiques plus correctes tout en conservant une grande quantité d'informations. D'autre part, les résultats obtenus lors de l'évaluation des autres parties, et en particulier de la partie de la correction, qui contient « la phase de génération automatique des phrases correctes », nous permettent d'espérer qu'une évaluation sur un corpus plus grand confirmera encore plus la validité de l'approche proposée.

À moyen terme, il serait intéressant d'évaluer la correction sur un corpus plus large et de valider l'approche sur d'autres informations syntaxiques, à savoir : les relations syntaxiques qui jouent un rôle primordial dans notre système.

6. Conclusion et perspectives

Dans cette contribution, nous avons présenté une nouvelle approche de la détection et de la correction automatique des erreurs syntaxiques dans les textes arabes. Cette approche est basée sur la génération de phrases utilisant le modèle de dépendance, dont les règles et les contraintes sont obtenues grâce à une description logique de la grammaire arabe par l'ontologie. Ce travail repose sur deux hypothèses. Premièrement, qu'il soit possible de générer toutes les phrases possibles et, deuxièmement, qu'il soit possible de comparer la phrase d'origine et les phrases générées. Cette étude en est encore à ses débuts et notre objectif principal était de mettre en œuvre une nouvelle approche de détection et de correction des erreurs syntaxique basée sur la génération automatique de phrases sur un corpus plus important. Les premiers résultats obtenus sont encourageants et nous sommes impatients de poursuivre nos recherches.

Chapitre 5

La correction des erreurs syntaxique de désinence casuelle en arabe

1. Introduction

La correction de texte est une tâche importante pour le traitement automatique des langues (TAL). Certaines applications de TAL nécessitent une analyse syntaxique des textes appréhendés en matière de relations grammaticales et codées sous forme d'annotations fonctionnelles dans des arbres syntagmatiques ou de dépendance, ce qui semble utile pour de nombreuses tâches. Cependant, ces analyseurs commettent des erreurs lors de l'analyse des phrases erronées dans les relations syntaxiques et les étiquetages ; nous pouvons donner un exemple avec la phrase erronée suivante : "دخل المعلمون المجتهدين"

```
دخل/VBD
المعلمون/DTNNS
المجتهدين/DTJJ
-----
Root (Root, دخل)
dobj (دخل , المعلمون)
dep (المعلمون, المجتهدين)
-----
(S(VP(VBD دخل) (NP (DTNNS المعلمون) (DTJJ المجتهدين))))
```

Tel que :

Tableau 13 étiquetage syntaxique de la phrase "دخل المعلمون المجتهدين"

Tag	Signification
DOBJ	Direct Object
DEP	Unclassified dependent
VBD	Verb, past tense
DTNNS	noun, plural with the determiner "Al" (ال)
DTJJ	adjective with the determiner "Al" (ال)

Nous avons remarqué que l'analyseur "Stanford Parser" a analysé la phrase bien qu'elle soit incorrecte. Les relations qu'il a établies entre les mots ne sont pas non plus correctes.

L'adjectif est un nom qui permet de décrire un autre nom dans la langue arabe. En anglais, les adjectifs précèdent le nom en cours de qualification. Cependant, en arabe, les adjectifs viennent après le nom en cours de qualification. L'adjectif en arabe est appelé "نعت" « le qualifiant ». Le nom qui est qualifié par cet adjectif est appelé "منعوتٌ" « le qualifié ».

Les règles suivantes s'appliquent à la formation de l'adjectif qualificatif arabe :

- En langue arabe, le mot employé pour décrire un nom vient après celui-ci.
- L'adjectif suit en genre (féminin/masculin), en définition (défini, indéfini) et en nombre (singulier, duel, pluriel) le qualifié auquel il se rapporte.
- L'adjectif suit le qualifié dans ses désinences casuelles اَلْإِعْرابُ [13]

Si nous voulons appliquer les règles de l'adjectif, il faut premièrement corriger l'erreur lorsque l'analyseur commence à annoter les tokens, certes la phrase correcte est :

"دخل المعلمون المجتهدون"

Tel que :

المجتهدون /DTJJS

Il est à noter que l'annotation DTJJS n'existe pas dans Stanford Parser, où DTJJS est un adjectif et au pluriel avec le déterminant « Al_ ال ».

L'erreur précédente s'appelle une erreur de désinence casuelle qui apparaît généralement sur les terminaisons des mots qui indiquent ses rôles syntaxiques.

Malgré les améliorations apportées aux analyseurs syntaxiques, il reste la correction des erreurs syntaxique parmi ses principaux objectifs, également pour tous les systèmes de TAL, et en particulier pour la langue arabe, qui est la langue pour laquelle les arbres syntaxiques nuisibles sont souvent spécifiés syntaxiquement.

L'objectif de cette étude est d'utiliser les résultats de l'analyseur syntaxique Stanford Parser et de les améliorer afin de corriger les erreurs syntaxiques au niveau de désinence casuelle en se basant sur une description logique des relations grammaticales de l'ontologie OSA.

[13] Un nom en arabe peut avoir quatre cas qui sont appelés en français : nominatif, datif, accusatif et génitif

2. La désinence casuelle syntaxique en langue arabe

Il existe deux types de phrases arabes [143], phrases nominales et phrases verbales :

- La phrase nominale, où le premier mot de la phrase est un nom (e.g. "الرجل مغربي—al-rajol maghribi"—l'homme est marocain).
- La phrase verbale, où le premier mot de la phrase est un verbe (e.g. "ولد الرجل في المغرب—wulida al-rajol fi al-maghrib"—l'homme est né au Maroc).

La désinence casuelle "الاعراب" est le changement qui se produit dans les terminaisons des mots en raison des divers facteurs impliqués dans ses fonctions grammaticales.

La langue arabe a trois types de désinence grammaticale, à savoir : le cas nominatif "المرفوع", le cas accusatif "المنصوب" et le cas génitif "المجرور" [144].

Voici toutes les situations possibles qui indiquent quand utiliser ces catégories :

2.1. LE CAS NOMINATIF

Le cas nominatif est utilisé dans plusieurs situations :

- Le sujet d'une phrase verbale.
- Le sujet et le prédicat d'une phrase nominale.
- Le vocatif (s'adressant directement à quelqu'un).
- Le cas nominatif est également la valeur par défaut pour les mots qui sont seuls.

2.2. LE CAS ACCUSATIF

Le cas accusatif est utilisé dans les cas suivants :

- L'objet d'un verbe transitif.
- Expressions adverbiales de temps, de lieu et de manière indiquant les circonstances dans lesquelles une action se déroule.

- L'objet interne ou structure accusative apparentée. Une façon d'intensifier une action en suivant le verbe avec son nom verbal correspondant et un adjectif le modifiant.
- L'accusatif circonstanciel. C'est une façon de décrire une condition / action qui se déroule en même temps que l'action principale.
- Montre le but d'une action.
- L'accusatif de spécification.
- Le prédicat de "kāna كان" « être » et ses sœurs (il existe 13 de ces petit verbe).

2.3.LE CAS GÉNITIF

Le cas génitif est utilisé dans plusieurs situations, à savoir :

- L'objet d'une préposition
- L'objet d'un adverbe locatif.
- Le deuxième terme d'une construction de substantifs dont le second terme est au génitif "إضافة"

2.4.LES ERREURS DÉSINENCES CASUELLES TRAITÉES

Les désinences casuelles grammaticales entraînent une modification du nom de l'une des trois manières suivantes :

- La voyelle courte de la dernière lettre "حركة" est changé i.e. "مدرسٍ", "مدرسٌ", "المدرسُ" ou "المدرسِ".
- Une lettre entière à la fin du nom est changée i.e. "المدرسون" ou "المدرسين".
- Parfois, une forme différente du nom est utilisée ; il est rarement utilisé et ne s'applique qu'aux pronoms i.e. "كَ" ou "أنتَ".

En langue arabe, les consonnes sont toujours écrites et les voyelles courtes sont facultatives. En conséquence, l'Arabe écrit peut-être totalement, partiellement ou entièrement voyellé. En général, les textes arabes ne sont pas vocalisés, à l'exception des textes religieux, des textes utilisés dans l'éducation des enfants et des poèmes. En arabe moderne, certaines voyelles sont indiquées pour aider les lecteurs à lever l'ambiguïté de certains mots.

Pour cette raison, nous avons convenu dans cette étude de traiter des textes non voyellé et de traiter les erreurs apparentes. À un niveau plus général, les erreurs syntaxiques qui peuvent être liées aux désinences casuelles, nous citons :

2.4.1. LE PLURIEL MASCULIN RÉGULIER "جمع المذكر السالم"

- La lettre finale est "ن", e.g. "مدرسون" "enseignants (de genre masculin)".
- L'avant-dernière lettre : lorsque le mot est nominatif, il s'agit de la lettre "و", e.g. "مدرسون". Quand il est accusatif ou génitif, il devient la lettre "ي", e.g. "مدرسين".

2.4.2. LE NOM DUEL "المثنى"

- La dernière lettre du dual est la lettre "ن" e.g. "رجلان" "deux hommes"
- L'avant-dernière lettre : quand le mot est nominatif, il s'agit de mettre la lettre "ا" e.g ."مدرسان". Quand il est accusatif ou génitif, il faut mettre la lettre "ي" e.g. "مدرسين".

2.4.3. LES SIX NOMS

Il y a six noms masculins singuliers en arabe qui prennent diverses formes :

- "أَخٌ" frère
- "أَبٌ" père
- "حَمٌ" beau-père
- "فَمٌ" bouche
- "هَنٌ" chose
- "ذُوْ" possesseur de

3. La méthodologie adoptée

La méthode de correction syntaxique adoptée est basée sur l'étiquetage de Stanford Parser et ses relations syntaxiques. Nous pouvons diviser cette méthode en quatre étapes (voir la figure 30), que nous détaillerons plus tard :

- La segmentation du texte en phrases.
- Le traitement des phrases avec Stanford Parser pour obtenir l'étiquetage de chaque mot avec les relations syntaxiques entre les mots extraits.
- L'élaboration de relations linguistiques correctes.
- La détection et de correction des erreurs de désinence casuelle.

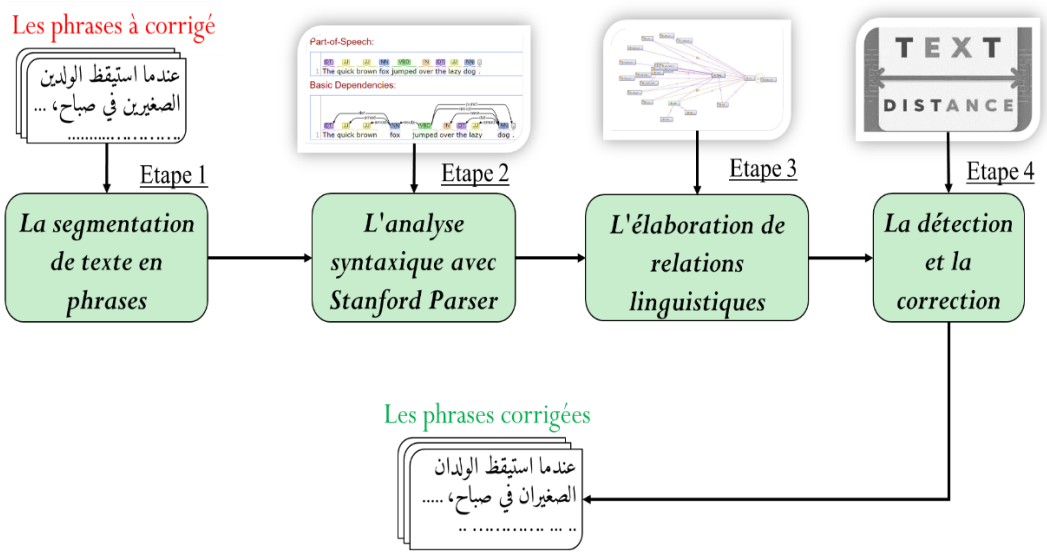

Figure 30 Les étapes de la méthode de correction syntaxique adoptée

3.1. LA SEGMENTATION DU TEXTE EN PHRASES

Le problème de la segmentation du texte en phrases en langue arabe est confus. Nous avons adopté la même méthode que la première approche pour segmenter le texte, qui se résume comme suit :

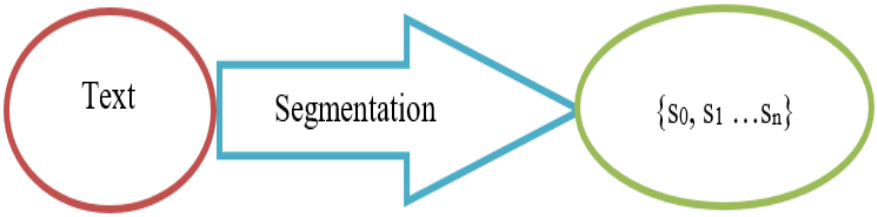

Figure 31 La segmentation du texte en phrases

Tels que : s0, s1, ... sont des phrases

3.2. L'ANALYSE SYNTAXIQUE

L'objectif de l'analyse syntaxique se résume en deux tâches : étiqueter les différentes composantes de la phrase par l'attribution à chaque mot un ensemble approprié des étiquettes morpho-syntaxiques utilisables pour la prochaine étape, et l'enrichissement des ressources linguistiques afin de les utiliser avec l'ontologie OSA qui contient les règles de la syntaxe arabe.

À ce stade, nous insérons les phrases à corrigé dans l'analyseur Stanford [71] et nous dessinons l'étiquette de tous les composants linguistiques de la phrase.

L'ordre des éléments linguistiques :

Cet analyseur organise les mots annotés en utilisant les deux champs “index” et “leafNumber”, ce qui nous permet de connaître l'ordre de chaque élément linguistique dans la phrase. Son rôle apparaît lorsqu'il y a une ambiguïté dans les relations identifiées par Stanford Parsrer.

```
<S index="0">
     <VP index="1">
           <VBD index="2" value="دخل" leafNumber="1">
           <NP index="3">
                 <DTNNS index="4" value="المعلمون" leafNumber="2">
                 <DTJJ index="5" value="المجتهدون" leafNumber="3">
           </NP>
     </VP>
</S>
```

Figure 32 L'ordre des éléments linguistiques

3.3. L'ÉLABORATION DE RELATIONS LINGUISTIQUES CORRECTES

Cette étape consiste à élaborer les relations grammaticales correctes en appliquant les résultats de l'étape précédente afin de corriger les relations incorrectes en utilisant l'ontologie arabe, qui contient une description logique de l'ensemble des règles de la langue arabe.

Nous avons adopté l'approche syntaxique de la grammaire de la dépendance fondée par Lucien Tesnière qui est appliqué dans Stanford Parser. Sur la base de la logique du prédicat, les relations grammaticales se divisent en deux catégories :

3.3.1. CATÉGORIE POUR LES CAS GRAMMATICAUX

Cette catégorie montre les relations de désinence casuelle entre les composants grammaticaux tels que le sujet, l'objet, etc.

Exemple :

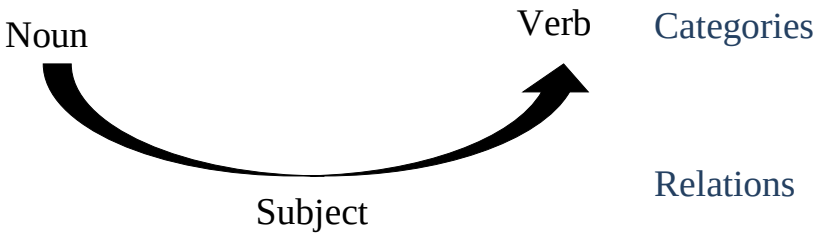

Figure 33 les catégories et ses relations

3.3.2. CATÉGORIE QUI REPRÉSENTE LES CARACTÉRISTIQUES DES MOTS

Les caractéristiques morphologiques telles que la relation de genre, qui attribuent aux mots les valeurs de masculinisation, féminisation, etc. Cette catégorie sera représentée à l'aide d'une matrice de valeurs.

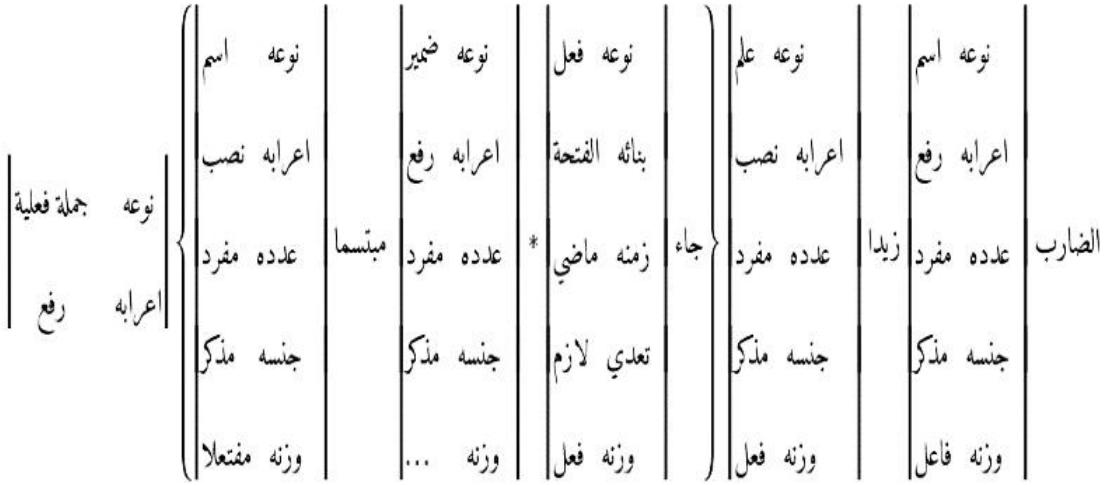

Figure 34 Exemple de matrice de valeurs

3.4. LA DÉTECTION ET LA CORRECTION DES ERREURS DE DÉSINENCE CASUELLE

Cette étape consiste à comparer la phrase correcte et la phrase originale en calculant la distance de « Levenshtein ».

Dans ce cas, nous avons deux possibilités :

- Si le système a constaté que la phrase d'origine et la phrase corrigée dans notre système sont identiques, il passe à la phrase suivante car il considère que la phrase est correcte.
- Si les deux phrases ne sont pas identiques, le système suggère alors la phrase correcte à l'utilisateur.

4. Exemple

Le but de cette section est d'illustrer le lien qui doit être établi entre l'analyse syntaxique et la correction des erreurs de désinence casuelle détectées.

Nous allons considérer l'exemple de phrase incorrecte suivant :

"جاء المعلمين المدعوين"

« Les professeurs invités sont venus »

Pour expliquer le fonctionnement du système dans son ensemble, nous allons appliquer notre approche afin de corriger les erreurs de désinence casuelle dans cette phrase :

4.1. LA SEGMENTATION DU TEXTE EN PHRASES

Dans cet exemple, nous n'avons qu'une phrase, la segmentation du texte est donc égale à la phrase elle-même.

Le résultat obtenu est :

Segmentation (جاء المعلمين المدعوين) = Seg_1 (جاء المعلمين المدعوين)

4.2. L'ANALYSE SYNTAXIQUE

Après la segmentation, nous passons à cette étape, qui vise à obtenir les traits morpho-syntaxiques de chaque mot ainsi que les relations syntaxiques :

```
جاء/VBD
المعلمين/DTNNS
المدعوين/DTJJ
-----
Root (Root, جاء)
dobj (جاء , المعلمين)
dep (المعلمين , المدعوين)
-----
(S (VP (VBD جاء)(NP (DTNNS المعلمين)(DTJJ المدعوين))))
```

4.3. L'ÉLABORATION DE RELATIONS LINGUISTIQUES CORRECTES

Après l'analyse syntaxique, nous devons corriger les éléments linguistiques de ces relations :

La première relation ***dobj*** (جاء, المعلمين) est une relation de sujet qui existe dans notre ontologie dont l'espace relationnel doit être un nom nominatif (ici pluriel) et l'étendue de cette relation doit être un verbe, cependant "المعلمين" est un nom pluriel accusatif avec la terminaison "ين" alors nous devons la changer par "ون" et ce morceau devient "المعلمون".

La deuxième relation ***dep*** (المعلمين, المدعوين) est une relation adjective entre un qualifiant "المدعوين" et un qualifié "المعلمين". Comme nous l'avons expliqué dans l'introduction, dans notre ontologie le qualifiant suit le qualifié dans sa désinence casuelle, puisque nous avons changé dans la première relation la terminaison de "المعلمون", nous devons également la remplacer par "المدعوين", alors il devient "المدعوون".

Comme résultat de cette étape, nous obtenons une nouvelle phrase correcte à l'aide de notre ontologie :

"جاء المعلمون المدعوون"

4.4. LA CORRECTION DES ERREURS DE DÉSINENCE CASUELLE

Enfin, nous comparons la phrase obtenue et la phrase d'origine en utilisant la distance de Levenshtein, qui détecte les erreurs de manière indirecte, afin qu'il propose la phrase correcte.

5. Evaluation et discussions

Dans cette section, nous présentons les résultats des évaluations effectuées sur les phrases arabes.

Il existe deux approches principales utiles à l'évaluation du TAL : la boîte noire " black-box" et la boîte en verre " glass-box " [145]. Dans l'évaluation de la boîte noire, les données de test sont choisies uniquement en fonction des relations spécifiées entre entrée et sortie, sans tenir compte de la structure interne du système testé. Dans l'évaluation en boîte en verre, l'évaluateur a accès aux différents rouages du système.

Les deux approches sont utiles à l'évaluation du TAL. Toutefois, l'évaluation de la boîte en verre implique parfois d'envisager la structure ou de diriger les connaissances d'un programme, sans l'exécuter sur des données de test.

Dans notre travail, nous avons choisi l'évaluation de boîte noire car elle conserve son sens initial, c'est-à-dire l'évaluation du résultat du système « phrases correctes » pour une entrée donnée « phrases incorrectes ». Dans ce contexte, l'évaluation peut ne pas être en mesure de localiser la source de l'erreur car elle les détecte de manière indirecte, mais elle fournira une indication du bon ou du mauvais fonctionnement du système.

Afin de valider notre approche, nous devons évaluer un ensemble de 100 phrases arabes, qui regroupe plusieurs types d'erreurs syntaxique de désinence casuelle, à savoir :

- 5 erreurs des six noms.
- 30 erreurs d'accord entre l'accusatif circonstanciel et le sujet.
- 35 erreurs d'accord entre l'adjectif et le nom.
- 30 erreurs d'accord entre le permutatif et l'antécédent "المبدل و المبدل عنه".

Pour que le lecteur puisse les comprendre facilement et que les linguistes les trouvent simples à évaluer, la majorité de ces phrases étaient courtes et simples.

Tableau 14 résultats de la correction des erreurs de désinence casuelle de ce travail

Erreurs de désinence casuelle	Détectées	Non détectées	Total
Désaccord entre l'adjectif et le nom	34	1	35
Désaccord entre l'accusatif circonstanciel et le sujet	28	2	30
Désaccord entre le permutatif et l'antécédent	26	4	30
Les six noms	5	0	5
Total (en pourcentage)	94.28%	5.72%	100

Tableau 15 résultats de la correction des erreurs de désinence casuelle de la première approche

Erreurs de désinence casuelle	Détectées	Non détectées	Total
Désaccord entre l'adjectif et le nom	32	3	35
Désaccord entre l'accusatif circonstanciel et le sujet	23	7	30
Désaccord entre le permutatif et l'antécédent	26	4	30
Les six noms	4	1	5
Total (en pourcentage)	85%	15%	100

Un résumé des résultats de l'évaluation est présenté dans les tableaux 14 et 15. La première colonne montre les différentes erreurs syntaxiques de désinence casuelle dans les phrases d'entrée. Ces résultats sont indiqués dans les colonnes 2 et 3. Le résultat est considéré comme correct si le système donne une phrase syntaxiquement correcte.

Il serait intéressant de comparer ce travail avec d'autres. Cependant, il n'y a aucun système disponible de correction des erreurs syntaxique de désinence casuelle en arabe, à l'exception de notre première approche dans le chapitre IV de la même équipe qui a développé le présent travail. De plus, il n'est pas possible de faire une comparaison avec d'autres langues.

Les résultats de notre approche dans la rangée inférieure montrent une détection totale de 94,28%, ce qui est un bon niveau pour ce type de tâche. En conclusion, le haut niveau de précision qui caractérise un niveau de fiabilité très important est particulièrement remarquable. Cela permet de corriger les erreurs avec une précision globalement très positive.

6. Conclusion et perspectives

Nous avons présenté dans cette étude une méthode de détection et de correction automatique des erreurs syntaxique et plus précisément des erreurs de désinence casuelle dans les textes arabes. Ce travail est basé sur Stanford Parser et l'ontologie de la grammaire arabe OSA ; le traitement de ses résultats se fait avec les règles et les contraintes obtenues par une description logique de la grammaire arabe dans l'ontologie.

Nous espérons que les résultats présentés seront utiles au développement des analyseurs syntaxiques arabes pour toutes les erreurs. Nous souhaitons également proposer une amélioration à Stanford Parser afin d'analyser et éventuellement de corriger les erreurs syntaxiques.

Conclusion générale

Le problème des erreurs syntaxiques est d'une grande importance dans le TAL. En effet, il se manifeste dans la majorité des textes écrits et constitue une source de diminution des performances dans la plupart des applications de TAL. Nos travaux de recherche détaillés dans cette thèse participent à l'automatisation de la correction de ces erreurs en langue arabe. À cette fin, la problématique du traitement automatique de ce type d'erreurs nous amène à obtenir une vision plus formelle sur les connaissances linguistiques computationnelles qui peuvent être traduites par un ensemble de points formalisé en rapport avec la syntaxe.

L'objectif principal de cette thèse était de proposer des méthodes de correction syntaxique dans les textes arabes. Dans un premier temps, nous avons présenté un état de l'art de différentes approches de correction des erreurs syntaxiques proposées dans différentes langues telles que l'anglais et le français. Ensuite nous avons illustré les travaux de recherche développés pour la langue arabe, ils sont en nombre très limité.

Dans un deuxième temps, nous avons proposé deux méthodes de détection et de correction automatique des erreurs syntaxiques pour les textes arabes. La première méthode ; sujet de notre première contribution se base sur la génération automatique des phrases à partir d'une description logique des règles grammaticales dans l'ontologie de la langue arabe. La deuxième méthode concerne la correction des erreurs syntaxiques de désinence casuelle basée sur l'analyseur syntaxique «Stanford Parser» et les règles grammaticales de la langue arabe dans l'ontologie. C'est l'objet de notre deuxième contribution.

Ce travail a apporté des résultats encourageants. En effet, nous avons effectué des expérimentations afin d’évaluer l’efficacité de nos approches. En ce qui concerne l’évaluation de la premiere contribution, nous l’avons testé manuellement sur 360 phrases. Les résultats sont très satisfaisants et montrent une précision de 92 % et un rappel de 84 %. Quant à l’évaluation de la deuxieme contribution, nous avons testé un ensemble de 100 phrases arabes qui regroupe plusieurs types d’erreurs de désinence casuelle. Les résultats avec l'utilisation de l’évaluation en boîte noire montrent une détection totale d'environ 94 %. A travers ces évaluations, nous avons montré sur le plan théorique que la réalisation de cette étude a permis de mettre en évidence les techniques de linguistiques computationnelles. En outre, sur le plan pratique, notre thèse peut donner lieu à une meilleure compréhension des textes capables d'améliorer les applications de TAL.

Toutefois nous sommes impatients de poursuivre nos travaux de recherche sur un système de correction syntaxique multilingue qui constitue une solution à ce problème. Nous souhaiterions également évaluer les approches présentées sur un corpus plus large, afin de confirmer les résultats obtenus. L'Arabe ne possède toujours pas de corpus des erreurs syntaxiques annoté. Certes, la constitution d'un tel corpus annoté rendrait possible l'utilisation de l'approche de traduction automatique statistique [113] pour les données d'entraînement.

Références

[1] Noaman, H.M., Sarhan, S.S., Rashwan, M. (2016). Automatic Arabic spelling errors detection and correction based on confusion matrix-noisy channel hybrid system. Egypt. Comput. Sci. J. (40), 54–64.

[2] Attia, M., Pecina, P., Samih, Y., Shaalan, K., Genabith, J. (2012). Improved spelling error detection and correction for Arabic. In Proceedings of the COLING 2012 Posters, Bombay, India, 103–112.

[3] Attia M., Pecina P., Samih Y., Shaalan K. Van Genabith, J. (2015). Arabic spelling error detection and correction. Natural Language Engineering, doi:10.1017/S1351324915000030

[4] Shaalan, K. F. (2005). Arabic GramCheck: a grammar checker for Arabic. Softw. Pract. Exp. (35), 643–665.

[5] Nora Madi et al. (2018). A Proposed Arabic Grammatical Error Detection Tool Based on Deep Learning. ACLING 352–355 DOI: 10.1016/j.procs.2018.10.482

[6] Lewis M. P., Simons G. F., Fenn, C. D. (2013). Ethnologue: Languages of the World Ethnologue: Languages of the World. Retrieved from http://www.ethnologue.com

[7] bin Ahmad, K. Z. (2019). Examining the effectiveness of Neuro-Linguistic Programming (NLP) techniques in improving Emotional Intelligence (EI) scores. Journal of Research in Emerging Markets, 1(1), 1-9. https://doi.org/10.30585/jrems.v1i1.313

[8] Booth, Andrew D. (1946). Report on Visit to American Laboratories. Rockefeller Foundation Archives.

[9] Weaver W. (1955). Translation. In Machine Translation of Languages: Fourteen Essays, ed. A. D. Booth and W. N. Locke, 15–23. Cambridge, MA: MIT Press.

[10] Chomsky N. (1957). Syntactic structures. The Hague, Mouton.

[11] Hutchins, John. (1996). ALPAC: The (In)famous Report. MT News International (14), 9-12.

[12] Miller, G., and Chomsky, N. (1963). Finitary models of language users. In R. Lute, R. Bush, and E. Galanter (eds.), Handbook of Mathematical Psychology, (2), New York, John Wiley.

[13] Fillmore, Charles J. (1968). The Case for Case. In Emmon Bach and Robert Harms. Universals in Linguistic Theory. New York: Holt - Rinehart – Winston (1), 88.

[14] Quillian, M. R. (1968). Semantic memory. In M. L. Minsky (Ed.), Semantic information processing. 227-259. Cambridge, MA: MIT Press.

[15] Roger Schank. (1972). Conceptual dependency: A theory of natural language understanding. Cognitive Psychology. (3), 552–631.

[16] Woods, William A. (1969). Augmented transition networks for natural language analysis. Rep. CS-1, Comput. Lab., Harvard U., Cambridge, Mass.

[17] Wilks Y. (1975). A preferential pattern-seeking semantics for natural language inference. Artificial Intelligence. (6)1, 53-74.

[18] Kay M. (1979). Functional Grammar, BLS 25, Linguistic Society of America.

[19] Weizenbaum, J. (1966). ELIZA — A computer program for the study of natural language communication between man and machine. Communications of the Association for Computing Machinery. (9), 36–45.

[20] Winograd T. (1972). Understanding natural language. Cognitive Psychology. 3(1), 1-191.

[21] Colby, K. (1973). Simulation of belief systems. In R. Schank & K. Colby (Eds.), Computer Models of Thought and Language. 251–286. San Francisco: Freeman.

[22] Woods, W., Kaplan, R. (1977). Lunar rocks in natural English: Explorations in natural language question answering. Linguistic Structures Processing. In Fundamental Studies in Computer Science. (5), 521-569,.

[23] Barbara J Grosz. (1979). Focusing and description in natural language dialogues. Technical report, DTIC Document.

[24] Mann W. C. and Thompson S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. Text, 8(3), 243–281.

[25] Hobbs, J. R. and Rosenschein, S. J. (1977). Making Computational Sense of Montague's Intensional Logic. Artificial Intelligence, 9(3), 287–306.

[26] Reichman, R. (1985). Getting Computers to Talk Like You and Me. Discourse Context, Focus, and Semantics (An ATN Model). The MIT Press, Cambridge, MA.

[27] McKeown, K. (1980). Generating Relevant Explanations: Natural Language Responses to Questions about Database Structure. In Proceedings of the First Annual Conference on Artificial Intelligence. Stanford (August). 306-309.

[28] McDonald, D. (1980). Natural Language Production as a Process of Decision-Making under Constraints. Ph.D. Thesis, MIT, Department of Electrical Engineering and Computer Science.

[29] Gauvain J., Adda G., Lamel L., Lefèvre F., Schwenk H. (2005). Transcription de la parole conversationnelle. TAL.

[30] Chomsky N. (1975). The logical structure of linguistic theory. New York: Plenum.

[31] Hockett C.F. (1995). A Manual of Phonology. Waverly Press and Indiana University Publications in Anthropology and Linguistics. Baltimore.

[32] Covington M. (2001). A fundamental algorithm for dependency parsing. Proceedings of the 39th annual ACM southeast conference. Citeseer. 95–102

[33] Tesnière L. (1959). Éléments de syntaxe structurale, Klincksieck. Paris.

[34] Hudson, R. (2016). Dependency Grammar. In A. Hippisley & G. Stump (Eds.), The Cambridge Handbook of Morphology (Cambridge Handbooks in Language and Linguistics. 657-682. Cambridge: Cambridge University Press. doi:10.1017/9781139814720.023

[35] Warren W. (1955). Translation. Machine Translation of Languages: Fourteen Essays.

[36] Popovic M. (2012). Class error rates for evaluation of machine translation output, in: Proceedings of the Seventh Workshop on Statistical Machine Translation, Association for Computational Linguistics. (1640), 71–75.

[37] Sergei N. (1989). Knowledge based machine translation. Machine Translation, 4(1), 5–24.

[38] Costa-Jussa M. R., Farrús M., Mariño J. B., Fonollosa J. A. (2012). Study and comparison of rule-based and statistical catalan-spanish machine translation systems, Computing and Informatics 31 (2), 245– 270.

[39] Philipp Koehn and Christof Monz. (2005). Shared task: Statistical machine translation between european languages. In Proceedings of the ACL Workshop on Building and Using Parallel Texts.

[40] Costa-Jussa M. R., Fonollosa J. A. (2015). Latest trends in hybrid machine translation and its applications, Computer Speech & Language 32 (1), 3–10.

[41] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. (2014). Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473.

[42] Vertan C. (2004). Language resources for the Semantic Web: perspectives for machine translation, in: Proceedings of the Second International Workshop on Language Resources for TranslationWork, Research and Training, ACL. 37–41.

[43] Ingwersen P. (1996). Cognitive perspectives of information retrieval interaction: Elements of a cognitive IR theory. Journal of Documentation. (52)1, 3–50. doi: 10.1108/eb026960

[44] Ellis D. (1992). The physical and cognitive paradigms in information retrieval research. Journal of Documentation. (48) 1, 45–64. doi: 10.1108/eb026889

[45] Wimalasuriya DC., Dejing Dou. (2010). Ontology-based information extraction: An introduction and a survey of current approaches. Journal of Information Science.(36), 306–23.

[46] Seth Grimes. (2011). Text/Content Analytics: User Perspectives on Solutions.

[47] Laura Chiticariu, Yunyao Li, Frederick R. Reiss. (2013). Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems. Proceedings EMNLP. 827-832.

[48] Lopez V., Uren V., Sabou M. and Motta E. (2011). Is Question Answering fit for the Semantic Web?: a Survey. Universität Bielefeld, Germany.

[49] Green B.F., Wolf A. K., Chomsky C. and Laughery K. (1961). BASEBALL: An automatic question answerer. Proceedings Western Joint Computer Conference, (19), 207-216. McGraw-Hill.

[50] Massih Amini, Patrick Gallinari. (2003). Apprentissage Numérique pour le Résumé de Texte. In Workshop of ATALA on Automatic Summarization: solutions and perspectives.

[51] Edmondson, H. P. (1969). New Methods in Automatic Extracting. Journal of the ACM. (2) 16, 264-285.

[52] Massih-Reza Amini, Cyril Goutte. (2010). A co-classification approach to learning from multilingual corpora. In Springer, ed. Machine Learning Journal. 105–121.

[53] Othman, B., Haggag, M. & Belal, M. (2014). A taxonomy for text summarization. Information Science and Technology,. (3)1, 43-50.

[54] Mohamed Atef Mosa, Arshad Syed Anwar, Alaa Hamouda (2019). A survey of multiple types of text summarization with their satellite contents based on swarm

intelligence optimization algorithms (163), 518-532. https://doi.org/10.1016/j.knosys.2018.09.008.

[55] Lovins, J.B. (1968). Development of a stemming algorithm. Mechanical translation and Computational Linguistics. (11), 22-31.

[56] Porter M. (1980). An Algorithm for Suffix Stripping Program. 14(3), 130-137.

[57] Horridge M., Knublauch H., Rector A., Stevens R., Wroe C. (2011). A practical guide to building OWL ontologies using Protégé 4 and CO-ODE tools. Ed. 1.3. The University of Manchester. hmowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf (21/11/2018)

[58] Brickley, D., and Guha, R. V. (1999). Resource description framework (rdf) model and syntax specification. W3c recommendation, World Wide Web Consortium.

[59] The World Wide Web Consortium (2004). OWL Web Ontology Language Guide: W3C Recommendation.

[60] Aljasser and M. S. Vitevitch (2018). A Web-based interface to calculate phonotactic probability for words and nonwords in Modern Standard Arabic. (50), 313-322.

[61] Khan M.I. (2018). Revival of Bio Medical Research in The Muslim World. International Journal of Human and Health Sciences. (02), 05-07.

[62] Farghaly A., Shaalan K. (2009). Arabic natural language processing: Challenges and solutions, ACM Trans. Asian Lang. Inf. Process., (8)4, 14.

[63] Tim Buckwalter. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0,. Linguistic Data Consortium, University of Pennsylvania.LDC Catalog No.:LDC2002L49.

[64] Habash N., Owen R. (2005). Arabic tokenization, part of speech tagging and morphological disambiguation in one fell swoop. In Proceedings of the Association for Computational Linguistics (ACL'05).

[65] Boudchiche M., Mazroui A., Ould Abdallahi Ould Bebah, Lakhouaja M., Boudlal A. ((2017)). AlKhalil Morpho Sys 2: a robust Arabic morpho-syntactic analyzer J. King Saud Univ. – Comput. Inf. Sci., (29), 141-146, 10.1016/j.jksuci.2016.05.002

[66] Damerau, F. J. (1964). A technique for computer detection and correction of spelling errors. Communications of the ACM. 7(3), 171–176.

[67] Church, K. W., and Gale, W. A. (1991). Probability scoring for spelling correction. Statistics and Computing. (1), 93–103.

[68] Van Delden, S., Bracewell, D. B., and Gomez, F. (2004). Supervised and unsupervised automatic spelling correction algorithms. In Proceedings of the IEEE International Conference on Web Services. 530–535.

[69] Haddad, B., and Yaseen, M. (2007). Detection and correction of non-words in Arabic: a hybrid approach. International Journal of Computer Processing of Oriental Languages. (20), 237–257.

[70] Hassan, A., Noeman, S., and Hassan, H. (2008). Language independent text correction using finite state automata. In IJCNLP, Hyderabad, India. 913–918.

[71] Chen D. and Manning C. (2014). A Fast and Accurate Dependency Parser using Neural Networks. Association for Computational Linguistics. 740–750.

[72] Taylor, A., Marcus, M., Santorini, B. (2003). The Penn Treebank: An Overview. In: Abeillé. 5-22.

[73] Schuster, S., Manning, C. D. (2016). Enhanced English Universal Dependencies: An Improved Representation for Natural Language Understanding Tasks. In LREC.

[74] Chang, P. , Tseng, H., Jurafsky, D. , Manning, C. D. (2009) . Discriminative Reordering with Chinese Grammatical Relations Features. In Proceedings of the Third Workshop on Syntax and Structure in Statistical Translation.

[75] Rafferty, A., and Manning C. (2008). Parsing Three German Treebanks: Lexicalized and Unlexicalized Baselines. In ACL Workshop on Parsing German.

[76] Green, S., De Marneffe, M. C. Bauer, J., Manning, C. D. (2011). Multiword Expression Identification with Tree Substitution Grammars: A Parsing tour de forcewith French. In EMNLP.

[77] Green, S., Manning, C. (2010). Better Arabic Parsing: Baselines, Evaluations, and Analysis. In Proceedings of the International Conference on Computational Linguistics (COLING). Beijing, China. 349-402.

[78] Shquier, M. M. A., Al-Howiti, K. M. (2015). Fully Automated Arabic to English Machine Translation System: Transfer-based approach of AE-TBMT.

[79] Ahmed, W., AntoBabu, P. (2016). Question Analysis for Arabic Question Answering Systems. ArXiv, abs/1701.02925.

[80] Arman, N., & Jabbarin, S. (2015). Generating Use Case Models from Arabic User Requirements in a Semiautomated Approach Using a Natural Language Processing Tool. J. Intelligent Systems. (24), 277-286.

[81] Green, S., Manning, C. (2010). Better Arabic Parsing: Baselines, Evaluations, and Analysis. In Proceedings of the International Conference on Computational Linguistics (COLING). Beijing, China. 349-402.

[82] Bikel, D. (2004). Intricacies of Collins‟ Parsing Model. Computational Linguistics, (30)4, 479-511.

[83] Collins, M. (1997) . Three generative, lexicalized models for statistical parsing. In The Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics, San Francisco. Morgan Kaufmann.

[84] Microsoft. Arabic toolkit service (ATKS). https://www.microsoft.com/en-us/research/project/arabic-toolkit-service-atks/. Dernier accès 10 Août 2019

[85] Daniel Naber. (2003). A rule-based style and grammar checker.

[86] Zheng Yuan. (2017). Grammatical error correction in non-native English. Technical Report. University of Cambridge, Computer Laboratory.

[87] منال نبيل قاسم السعدي اليافعي, الأخطاء التركيبية لدى متعلمي اللغة العربية, جامعة قطر كلية الآداب والعلوم, 2016.

[88] FEUILLARD C. (2006). Le fonctionnalisme d'André Martinet, Pour une linguistique des langues (sous la direction de Henriette Walter et Colette Feuillard), Paris: Presses universitaires de France.

[89] Hashemiah Mohammad Almusawi. (2019). Determinants of spelling proficiency in hearing and deaf graduate students: The presentation of medial glottal stop, Ampersand. (6). https://doi.org/10.1016/j.amper.2019.100050

[90] Kukich Karen. (1992). Techniques for automatically correcting words in text. ACM Computing Surveys (CSUR). (24)4, 377–439. Print.

[91] Leacock, C., Chodorow, M., Gamon, M., and Tetreault, J. (2010). Automated grammatical error detection for language learners. Synthesis lectures on human language technologies. 3(1), 1134.

[92] MACDONALD, N., FRASE, L., GINGRICH, P. et KEENAN, S. (1982). The writer's workbench Computer aids for text analysis. Educational psychologist. 17(3), 172–179.

[93] Ramirez Bustamante. Flora Declerck Thierry. Sanchez Leon Fernando. (2000). Towards a Theory of Textual Errors. In Proceedings of the 3rd International Workshop on Controlled Language Applications (CLAW'00). Seattle, USA. 29-30.

[94] Heidorn George E. (2000). Intelligent Writing Assistance. In Dale, Robert, Moisl, Hermann et Somers, Harold (Eds.). Handbook of Natural Language Processing. 181–207.

[95] Heidorn, George E. (1975). Augmented Phrase Structure Grammars. In Schank, Roger et Nash-Webber, Bonnie (Eds.). Theoretical Issues in Natural Language Processing: An Interdisciplinary Workshop in Computational Lingustics, Psychology, Linguistics, Artifical Intelligence. Cambridge, Mass. (1–5).

[96] Bender, E. M., Flickinger, D., Oepen, S., Walsh, A., and Baldwin, T. (2004). Arboretum: Using a precision grammar for grammar checking in call. In In Proceedings of the InSTIL/ICALL Symposium: NLP and Speech Technologies in Advanced Language Learning Systems. (83-86).

[97] Douglas, S. and Dale, R. (1992). Towards robust PATR. In Proceedings of the 14th conference on Computational linguistics. (2), 468-474.

[98] Dini, L. and Malnati, G. (1993). Weak constraints and preference rules. Studies in Machine Translation and Natural Language Processing. (75-90).

[99] Holan, T., Kubon, V., and Platek, M. (1997). A prototype of a grammar checker for czech. In ANLP. (147-154).

[100] Markov, A. A. (1960). The theory of algorithms. Am. Math. Soc. Transl. (15)114.

[101] Atwell, E. S. and Elliot, S. (1987). Dealing with ill-formed english text. The Computational Analysis of English: A Corpus-Based Approach. (120-138).

[102] Leacock, C., Chodorow, M. (2003). Automated grammatical error detection. In M. D. Shermis and J. C. Burstein (Eds.), Automated essay scoring: A cross-disciplinary perspective. Mahwah, NJ: Lawrence Erlbaum Associates. (195–207).

[103] Hdez, S. D. and Calvo, H. (2014). Conll-2014 shared task: Grammatical error correction with a syntactic n-gram language model from a big corpora. In Proceedings of the Eighteenth Con-ference on Computational Natural Language Learning: Shared Task. Baltimore, Maryland. Association for Computational Linguistics. (53-59).

[104] Tetreault, J., & Chodorow, M. (2008). The ups and downs of preposition error detection. Paper presented at the COLING, Manchester, UK.

[105] Golding, A.R. & Roth, D. Machine Learning (1999) (34), 107. https://doi.org/10.1023/A:1007545901558

[106] Rozovskaya, A. and Roth, D. (2014). Building a state-of-the-art grammatical error correction system.

[107] Rozovskaya, A. and Roth, D. (2010). Training paradigms for correcting errors in grammar and usage. In North American Chapter of the Association for Computational Linguistics.

[108] Koehn, P. (2010). Statistical Machine Translation . Cambridge University Press, New York, NY, USA, 1st edition.

[109] Brockett, C., Dolan, W. B., and Gamon, M. (2006). Correcting ESL errors using phrasal SMT techniques. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, USA. (249-256).

[110] Dahlmeier, D. and Ng, H. T. (2012). A beam-search decoder for grammatical error correction. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Pro-cessing and Computational Natural Language Learning , EMNLP-CoNLL '12. Association for Computational Linguistics. (568-578).

[111] Yuan, Z. and Felice, M. (2013). Constrained grammatical error correction using statistical machine translation. In Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task. Association for Computational Linguistics. Soa, Bulgaria. (52-61).

[112] Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., and Bryant, C. (2014). The conll-2014 shared task on grammatical error correction. In Proceedings of the Eighteenth Con-ference on Computational Natural Language Learning: Shared Task. Association for Computational Linguistics. Baltimore, Maryland. (1-14).

[113] Chollampatt, S., Hoang, D. T., and Ng, H. T. (2016). Adapting grammatical error correction based on the native language of writers with neural network joint models. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. Austin, Texas. (1901-1911).

[114] Gamon, M. (2010). Using mostly native data to correct errors in learners' writing: A meta-classifier approach. In Proceedings of NAACL 2010. Association for Computational Linguistics.

[115] Dahlmeier, D., & Ng, T. H. (2011). Grammatical error correction with alternating structure optimization. Paper presented at the Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT-2011).

[116] Gamon, M. (2011). High-order sequence modeling for language learner error detection. Paper presented at the Sixth Workshop on Innovative Use of NLP for Building Educational Applications.

[117] Rozovskaya, A., & Roth, D. (2011). Algorithm selection and model adaptation for ESL correction tasks. Paper presented at the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT-2011).

[118] Wagner, J. (2012). Detecting grammatical errors with treebank-induced, probabilistic parsers. Dublin City University, Dublin, Ireland.

[119] Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Tree Bank. Computational Linguistics. (19), 313–330.

[120] Ehsan, N. and Faili, H. (2013). Grammatical and context-sensitive error correction using a statistical machine translation framework. Software: Practice and Experience , 43(2), (187-206).

[121] Felice, M., Yuan, Z., Andersen, Ø. E., Yannakoudakis, H., and Kochmar, E. (2014). Grammatical error correction using hybrid systems and type filtering. In Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task. Association for Computational Linguistics. Baltimore, Maryland. (15-24).

[122] Rozovskaya, A. and Roth, D. (2016). Grammatical error correction: Machine translation and classifiers. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin, Germany. (1), (2205-2215).

[123] Dale, R. and Kilgarri, A. (2011). Helping our own: The HOO 2011 pilot shared task. In Proceedings of the 13th European Workshop on Natural Language Generation. Association for Computational Linguistics. (242-249).

[124] Rijsbergen, C. J. V. (1979). Information Retrieval. Butterworth-Heinemann, Newton, MA, USA, 2nd edition.

[125] Dale, R., Anisimo, I., and Narroway, G. (2012). Hoo 2012: A report on the preposition and determiner error correction shared task. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP. Association for Computational Linguistics. (54-62).

[126] Rozovskaya, A., Sammons, M., and Roth, D. (2012). The ui system in the hoo 2012 shared task on error correction.

[127] Dahlmeier, D., Ng, H. T., and Ng, E. J. F. (2012). Nus at the hoo 2012 shared task. In Proceedings of the Seventh Workshop on Building Educational Applications Using NLP. Association for Computational Linguistics. (216-224).

[128] Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., and Bryant, C. (2014). The conll-2014 shared task on grammatical error correction. In Proceedings of the Eighteenth Con-ference on Computational Natural Language Learning: Shared Task. Association for Computational Linguistics. Baltimore, Maryland. (1-14).

[129] Rozovskaya, A., Chang, K.-W., Sammons, M., and Roth, D. (2013). The university of illinois system in the conll-2013 shared task. In Proceedings of the Seventeenth Conference on Com-putational Natural Language Learning: Shared Task. Association for Computational Linguistics. Soa, Bulgaria. (13-19).

[130] Mohit, B., Rozovskaya, A., Habash, N., Zaghouani, W., and Obeid, O. (2014). The first qalb shared task on automatic text correction for arabic. In Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP). Association for Computational Linguistics. Doha, Qatar. (39-47).

[131] Nora Madi et al. (2018). A Proposed Arabic Grammatical Error Detection Tool Based on Deep Learning. ACLING 352–355 DOI: 10.1016/j.procs.2018.10.482

[132] Al-Sughaiyer, I. Al-Kharashi, I. (2004). Arabic Morphological Analysis Techniques: A Comprehensive Survey.Journal of the American Society for Information Science and Technology.

[133] Socher, R. Christopher, D. (2010). Better Arabic parsing: baselines, evaluations, and analysis. COLING '10 Proceedings of the 23rd International Conference on Computational Linguistics.

[134] Klein, D. Christopher, D. (2003). Fast exact inference with a factored model for natural language parsing. in Suzanna Becker. (15), (3–10). MIT Press.

[135] Elkateb, S.Black, W. Vossen Piek, Farwell, D. Rodríguez, H. Pease, A. Alkhalifa M. (2006). Arabic WordNet and the challenges of Arabic. In Proceedings of the Arabic NLP/MT Conference, London, UK.

[136] Ferré, S. (2017). Sparklis: an expressive query builder for sparql endpoints with guidance in natural language. Semantic Web. 8(3), (405–418).

[137] Kubler, S. Mcdonald, R. Nivre, J. (2009). Dependency parsing. Synthesis Lectures on Human Language Technologies. (1), (1–127).

[138] Mustafa, I., Al-Ziyaat, Abdul Qaadir, A., H. & Al-Najjaar, M. (1960). Al-Waseet Dictionary, the Academy of the Arabic Language in Cairo.

[139] Almalki, T. (2015).أنطولوجيا النحو العربي نحو توصيف منطقي لساني للنحو العربي القديم. دار النابغة للنشر والتوزيع. Tanta. Egypt.

[140] Hadrich Belguith, L., Aloulou, C. & Ben Hamadou, A. (2008). MASPAR : De la segmentation à l'analyse syntaxique de textes arabes. Revue Information Interaction Intelligence I3. (7)2.

[141] Souteh, Y. & Bouzoubaa, K. (2011). SAFAR platform and its morphological layer. Eleventh Conference on Language Engineering ESOLEC'2011. Cairo. Egypt.

[142] Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertionsand reversals SOL Phys Dokl. (707-710).

[143] Ryding K. (2005). Reference Grammar of Modern Standard Arabic. Cambridge University Press. Cambridge, UK.

[144] Al-Muhtaseb H., Mellish C. (1998). Some Differences Between Arabic and English: A Step Towards an Arabic Upper Model. The 6th International Conference on Multilingual Computing. Cambridge, UK.

[145] Illingworth V. (1990). Dictionary of computing. London, Oxford University Press.

Annexe A. Les classes d'erreurs et exemples

Tableau 16 Les classes d'erreurs et exemples

Error class	Error type	Example	The grammatical rule
Agreement class	The subject or the pro-agent	▪ نجح [المجتهدين ← المجتهدون] في الامتحان الموحد ▪ The diligent succeeded in the standardized exam	The subject is a nominative noun
	The predicate of “Inna” and its sisters	▪ إن الطالبين الواقفين [مجتهدين ← مجتهدان] ▪ <Indeed> The two students standing are hardworking	the predicate of “Inna” is nominative (مرفوع)
	The exception	▪ عاد الفائزون إلا [سعدٍ ← سعدًا] ▪ The winners returned all except Saadah	If the exceptive sentence is affirmative and complete, the excepted object takes the accusative case, النصب
	The subject of the nominal sentence “Inna” and its sisters	▪ إن [المهاجمون ← المهاجمين] فشلوا في خطتهم ▪ <Indeed> The attackers failed in their plan	the subject of “Inna” is accusative (منصوب)
	The adjective and its qualified	▪ مررت برجلٍ [الفاضلِ ← فاضلٍ] ▪ I passed by a virtuous man	The adjective “النعت” follows in case ending the qualified “المنعوت” to which it refers
	The demonstrative pronoun	▪ [هذه ← هذا] الماء [الصافية ← الصافي] ▪ This pure water	The demonstrative pronoun follows the noun (feminine, masculine, singular, plural, etc.)
	The number phrase	▪ له [ثلاث ← ثلاثة] بنين و [ثلاثة ← ثلاث] بناتٍ ▪ He has three sons and three daughters	The number has the opposite gender of the noun
	Conjugation of the verb with its subject	▪ [تنهض ← ينهض] التعليم بالمجتمعات ▪ Education is rising up the societies	The verb must agree with its subject in both number and gender
	The adjective and its qualified	▪ انتقل [الطالبين ← الطلاب] الناجحون إلى الجامعات ▪ Successful students moved to universities	The adjective “النعت” follows in number the qualified “المنعوت” to which it refers
Case ending class	The case endings of the predicate “Kana” and its sisters	▪ أصبح المعروف [منكر ← منكرا] ▪ the right became wrong	The predicate of “Kana” must always be accusative
	The case endings of the circumstantial	▪ جاء الولد الخائف [مسرع ← مسرعا] ▪ The frightened boy came quickly	The circumstantial must always be accusative
	The case endings of the object	▪ عندما وجدنا الطفلان [جالسان ← جالسين] قرب الحديقة ▪ When we found the two children sitting near the park	The first object must always be accusative
	Case endings of the genitive and its adjective	▪ يرى جوهر الحقائق [بعينان ← بعينين] [ثاقبتان ← ثاقبتين] ▪ He sees the essence of the facts with piercing eyes	The genitive noun by a preposition must always be genitive and the adjective follow it.
	Case endings of the possessor and its adjective	▪ وقفت بجانب [السيارتان ← السيارتين] [الجميلتان ← الجميلتين] ▪ I stood beside the two beautiful cars	The genitive noun by the possession must always be genitive and the adjective follow it.
	Deletion of "Nun" in the case of the present nominative verb	▪ كانوا [يعلموا ← يعلمون] أنّ الصوم ليس مجرّد امتناعٍ عن الطعام والشراب ▪ They knew that fasting was not just abstinence from food and drink	The "Nun - ن" cannot be deleted with the nominative verb.
Definite article	Particle of negation "غير"	▪ أضف لمعلوماتك [الغير ← غير] الكافية ▪ Add to your insufficient information	Particle of negation "غير" is used without definite article.
	Definition of the possessed	▪ يشهد [العصرنا ← عصرنا] الحاضر تطورا كبيرا ▪ Our present era is witnessing great development	The noun possessed must always be indefinite.

Résumé de la thèse :

Le Traitement Automatique de Langues (TAL) est un domaine de recherche en plein essor en informatique et en sciences cognitives utilisant de nombreuses méthodes expérimentales. La syntaxe est l'une des propriétés les plus importantes de la langue. Elle contient un ensemble de règles structurelles par lesquelles les unités linguistiques se combinent en phrases qui sont partagées entre les locuteurs natifs afin de permettre une communication fluide. La correction automatique des erreurs syntaxiques est parmi les applications de TAL, elle vise à corriger les erreurs syntaxiques dans une phrase source donnée en se basant sur des modèles informatiques et linguistiques.

La présente thèse de doctorat traite le problème des erreurs syntaxiques dans la langue arabe. Pour réaliser cet objectif nous avons proposé deux solutions :

La première est une nouvelle approche basée sur la génération automatique de phrases correctes. Tout d'abord, nous extrayons les mots de la phrase concernée. Ensuite, à partir de ces mots et grâce à une description logique des règles de la grammaire arabe dans l'ontologie nous générons toutes les phrases possibles. Nous comparons ensuite la phrase d'origine avec les phrases (correctes syntaxiquement) générées pour détecter d'éventuelles erreurs. Enfin, dans la phase de correction, si le système ne trouve aucune phrase qui ressemble à la phrase d'origine, les phrases alternatives correctes seront automatiquement proposées. Des tests réussis ont été effectués à l'aide d'un corpus de phrases arabes. Le système mis en œuvre a atteint un taux de précision d'environ 92% et un taux de rappel d'environ 84%. En observant les résultats obtenus, nous concluons que cette approche est prometteuse.

La deuxième solution traite la correction des erreurs syntaxiques, particulièrement de désinence casuelle, en utilisant l'analyseur syntaxique «Stanford Parser » ainsi que l'ontologie de la grammaire Arabe qui contient les règles de la langue arabe. En premier lieu, nous segmentons le texte en phrases. En second lieu, nous extrayons les traits morpho-syntaxiques de chaque mot avec les relations syntaxiques provenant du parseur Stanford. Ensuite, nous traitons les relations obtenues avec l'ontologie. En dernier lieu, nous comparons la phrase d'origine avec la phrase corrigée afin de détecter l'erreur.

Mots-Clés: La langue Arabe, les erreurs syntaxiques, le traitement automatique des langues, les ontologies, l'analyse syntaxique, Stanford Parser.

www.univh2c.ma

19, Rue Tarik Ibnou Ziad, B.P,9167, Mers Sultan Casablanca- Maroc
Tél. +212 5 22.43.30.30/31 I Fax: +212 5 22 27 61 50

E-mail: presidence@univcasa.ma

Avenue Hassan II B.P. 150, Mohammedia, Maroc
Tél : +212 5 23 31 46 35/36 Fax : +212 5 31 46 34
E-mail : presidence@univh2m.ac.ma