Arabic Language Processing using Python NLTK

Dr. Sadik Bessou

University Ferhat Abbas, Sétif-1-

bessou.s@univ-setif.dz

Abstract

In this paper, we present a useful toolkit: NLTK (Natural Language ToolKit), a suite of libraries and programs in python language for symbolic and statistical natural language processing. We explain the main functionalities such tokenization, stemming, tagging, parsing, and semantic reasoning applied on Arabic texts. These functionalities are very useful for Natural language processing developers and even for linguists.

1. Introduction

Natural language processing (NLP) is about developing applications and services that are able to understand human languages.

In this paper, we present practical examples of NLP using the NLTK (Natural language Toolkit) with python.

Millions of gigabytes every day are generated by blogs, social websites, and web pages. There are many companies gathering all of this data to better understand users and their passions and make appropriate changes.

These data could show that the people of Brazil are happy with product A, while the people of the US are happier with product B. With NLP, this knowledge can be found instantly (i.e. a real-time result). For example, search engines are a type of NLP that give the appropriate results to the right people at the right time.

However, search engines are not the only implementation of NLP. There are many awesome implementations out there [3].

2. NLP Implementations

These are some successful implementations of natural language processing:

- **Search engines** like Google, Yahoo, etc. Google's search engine understands that you are a tech person, so it shows you results related to that.
- **Social website feeds** like your Facebook news feed. The news feed algorithm understands your interests using NLP and shows you related ads and posts more likely than other posts.
- **Speech engines** like Apple Siri.
- **Spam filters** like Google spam filters. It is not just about your usual spam filtering; now, spam filters understand what is inside the email content and see if it is spam or not [3].

3. NLP Libraries

There are many open source NLP libraries. These are some of them:

- Natural language toolkit (NLTK) ;
- Apache OpenNLP ;
- Stanford NLP suite ;
- Gate NLP library.

4. Working with NLTK

Natural language toolkit (NLTK) is a leading platform for building Python programs to work with human language data [1]. It is the most popular library for NLP. It was written in Python and has a big community behind it.

NLTK also is very easy to learn, actually, it is the easiest NLP library that we can use.

NLTK is a suite of open source Python modules, data sets, and tutorials supporting research and development in NLP [3].

4.1. Download NLTK

Figure 1. presents the script to download NLTK.

```
1 import nltk
2 nltk.download()
```

Figure 1. Download NLTK.

When we download NLTK, we can see a window showing the different packages: collections, corpora, models. If the status of an element is “not installed” or “partial”, we can click on it and click on download.

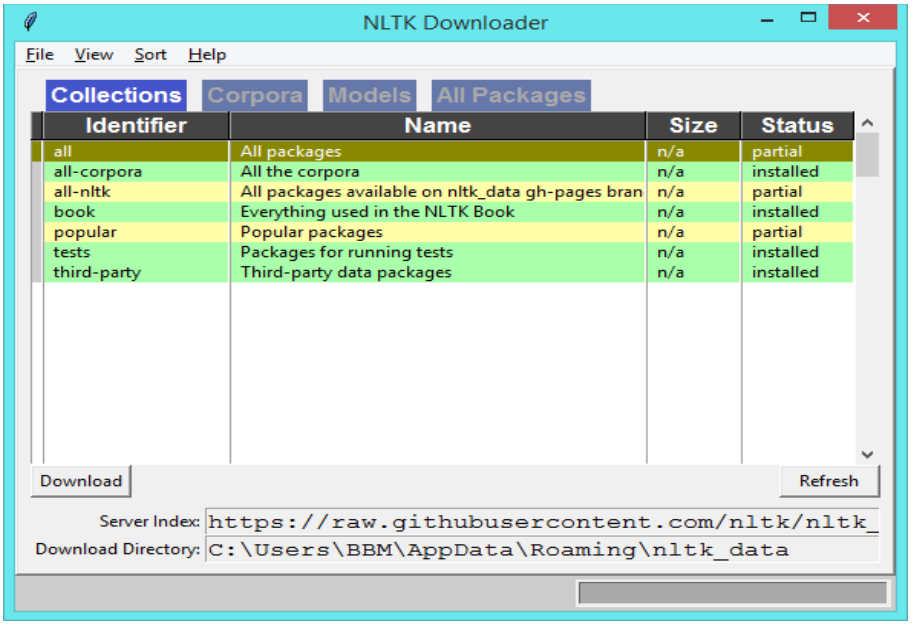


Figure 2. NLTK Downloader.

Table 1. presents the different tasks that could be done by the toolkit such string processing, machine learning, chunking, parsing, part-of-speech tagging and the use the different evaluation metrics.

Table 1. Language processing tasks and corresponding NLTK modules with examples of functionality [2].

Language processing task	**NLTK modules**	**Functionality**
Accessing corpora	corpus	standardized interfaces to corpora and lexicons
String processing	tokenize, stem	tokenizers, sentence tokenizers, stemmers
Collocation discovery	collocations	t-test, chi-squared, point-wise mutual information
Part-of-speech tagging	Tag	n-gram, backoff, Brill, HMM, TnT
Machine learning	classify, cluster, tbl	decision tree, maximum entropy, naive Bayes, EM, k-means
Chunking	chunk	regular expression, n-gram, named-entity
Parsing	parse, ccg	chart, feature-based, unification, probabilistic, dependency
Semantic interpretation	sem, inference	lambda calculus, first-order logic, model checking
Evaluation metrics	Metrics	precision, recall, agreement coefficients
Probability and estimation	Probability	frequency distributions, smoothed probability distributions
Applications	app, chat	graphical concordancer, parsers, WordNet browser, chatbots
Linguistic fieldwork	Toolbox	manipulate data in SIL Toolbox format

4.2. Preprocessing

Figure 3. shows an example of text preprocessing by removing the duplicate punctuation.

```
In [30]:
import nltk
import re
from nltk import sent_tokenize
def remove_duplicate_punctuation(s):
    return(re.sub(r'(\.|\?|!|؟|،|"|»|«|;)\1+', r'\1 ', s))
msa_sents =[]
with open('C:/ms.txt', 'r', encoding='utf-8') as in_file:
    text = remove_duplicate_punctuation(in_file.read())
    lines = [p for p in text.split('\n') if p]
    for line in lines:
        msa_sents += sent_tokenize(line)
msa_sents[:3]

Out[30]: ['\ufeff انا الان في اخر سنة دراسية وبعدها سألتحق بالجامعة',
 'اختار التخصص الذي يحدد مكانتي في المجتمع مستقبلاً.',
 'منذ ايام المتوسطة كنت احب مادة الرياضيات']
```

Figure 3. Text preprocessing.

4.3. Tokenization

Figure 4. shows a text that will be treated in the coming paragraphs. To tokenize a text we have to import the modules: “sent_tokenize” for sentences tokenization and “word_tokenize” for words tokenization.

```
from nltk.tokenize import sent_tokenize, word_tokenize
eg = " ألتحق بالجامعة. سأختار التخصص الذي يحدد مكانتي في المجتمع مستقبلا ما رأيكم؟ منذ ايام المتوسطة كنت احب مادة الرياضيات "
```

Figure 4. Importing tokenization modules.

4.3.1. Sentence tokenization

To get a text tokenized into sentences we use the instruction “sent_tokenize(text)” as shown in figure 5.

```
print(sent_tokenize(eg))
```

```
['انا الان في اخر سنة دراسية وبعدها سألتحق بالجامعة.سأختار التخصص الذي يحدد مكانتي في المجتمع مستقبلاً.', 'ما رأيكم؟ منذ ايام المتوسطة كنت احب مادة الرياضيات']
```

Figure 5. Splitting text into sentences.

4.3.2. Word tokenization

To get a text tokenized into words we use the instruction “word_tokenize(text)” as shown in figure 6.

```
print(word_tokenize(eg))
```

```
['انا', 'الان', 'في', 'اخر', 'سنة', 'دراسية', 'وبعدها', 'سألتحق', 'بالجامعة.سأختار', 'التخصص', 'الذي', 'يحدد', 'مكانتي', 'في', 'المجتمع', 'مستقبلاً', '.', 'ما', 'رأيكم؟', 'منذ', 'ايام', 'المتوسطة', 'كنت', 'احب', 'مادة', 'الرياضيات', '']
```

Figure 6. Splitting text into words.

4.4.Stop words

4.4.1. List of Arabic stop words

Stop words are the most common words in a language. Figure 7. shows the list of Arabic stop words.

```
from nltk.corpus import stopwords
sw=set(stopwords.words("Arabic"))
print(sw)
```

```
{'سوى', 'إنما', 'انه', 'ذات', 'يكن', 'هاته', 'أكثر', 'بهما', 'بمن', 'لاسيما', 'ها', 'ليسا', 'لن', 'إلى', 'اللتين', 'بهم', 'لذ'
ن', 'اللائي', 'لها', 'هذا', 'آها', 'كلاهما', 'بعض', 'بكما', 'فإذا', 'بك', 'منها', 'وإذ', 'فيم', 'لكي', 'أنا', 'لي', 'إذا', 'ك
لما', 'أنتما', 'ليسوا', 'آه', 'علي', 'ليت', 'إيه', 'حيث', 'لك', 'ذي', 'ذواتي', 'منه', 'اللذين', 'اللواتي', 'كأنما', 'هناك',
'وهو', 'من', 'عن', 'ألا', 'كأي', 'أوه', 'بس', 'هاتان', 'عدا', 'يا', 'هذي', 'ولو', 'بكم', 'أنى', 'بعد', 'هما', 'لكن', 'لنا',
'هذ', 'لوما', 'إي', 'لهما', 'ماذا', 'هكذا', 'كليهما', 'عسى', 'هم', 'كلتا', 'بيد', 'فمن', 'لسنا', 'كم', 'لستم', 'هذه', 'غير
, 'قد', 'إليك', 'والذي', 'عل', 'ليست', 'كيت', 'الذي', 'ذلكم', 'أينما', 'ذلكن', 'إنه', 'بلى', 'بماذا', 'في', 'هنالك', 'كليكما
ا', 'أولئك', 'تلكم', 'إليكن', 'ذينك', 'لعل', 'لدى', 'لولا', 'إذما', 'هذين', 'فيه', 'ولا', 'ومن', 'أين', 'بخ', 'لو', 'هذان', '
تي', 'إن', 'تلك', 'إذ', 'عما', 'فيما', 'ليستا', 'من', 'عليك', 'تين', 'هما', 'أنتن', 'بل', 'كأن', 'كلا', 'لهم', 'عند', 'ذين',
'لا', 'أف', 'منذ', 'وما', 'تلكما', 'له', 'خلا', 'لهن', 'ذاك', 'هيا', 'فلا', 'هيهات', 'نحن', 'آي', 'حيثما', 'أنت', 'ذوا', 'كما'
, 'ينا', 'لكم', 'ذانك', 'لكما', 'كي', 'ثم', 'هو', 'لما', 'هاك', 'ذا', 'بها', 'أنتم', 'دون', 'اللتيا', 'حاشا', 'إليكم', 'ليس
', 'أن', 'هاهنا', 'اللتان', 'نعم', 'وإن', 'ذو', 'لستن', 'إنا', 'هنا', 'مع', 'مه', 'ذه', 'ذان', 'بي', 'لكيلا', 'إلا', 'لستما',
'والذين', 'حبذا', 'لم', 'هؤلاء', 'ثمة', 'فتان', 'ذلك', 'لسن', 'الذين', 'ممن', 'هل', 'التي', 'إما', 'ذلكما', 'كيفما', 'هلا', '
فيها', 'أما', 'اللاتي', 'اللذان', 'كيف', 'ريث', 'أقل', 'ما', 'وإذا', 'مي', 'مهما', 'لست', 'بهن', 'هيد', 'عليه', 'إذن', 'متى
, 'فإن', 'ولكن', 'أو', 'إليكما', 'به', 'كذا', 'نحو', 'لكنما', 'أيها', 'كذلك', 'بما', 'بين', 'هاتين', 'هاتي', 'حين', 'ذواتا
', 'تينك', 'أولاء', 'حتى', 'سوف', 'أم', 'كأين', 'أي', 'كل'}
```

Figure 7. Arabic stop words.

4.4.2. Removing Stop words from a sentence

In several cases of NLP applications, we need to remove stop words, because they carry less important meaning than other words. Figure 8. shows an example of stop words filtering from the previous text.

```
words=word_tokenize(eg)
filtred=[]
for w in words:
    if w not in sw:
        filtred.append(w)
print(filtred)
```

```
['انا', 'الان', 'اخر', 'سنة', 'دراسية', 'وبعدها', 'سألتحق', 'بالجامعة', '.', 'سأختار', 'التخصص', 'بعدد', 'مكانتي', 'المجتمع'
', 'مستقبلا', 'رأيكم؟', 'ايام', 'المتوسطة', 'كنت', 'احب', 'مادة', 'الرياضيات
']
```

Figure 8. Stop words removing.

4.5. Stemming

Stemming works on words without knowing their context, which is why it has lower accuracy and is faster than lemmatization [3]. Stemming reduces the inflected words to theirs stems. Figure 9. presents a script that generates the stems of the previous text.

```
from nltk.stem.isri import ISRIStemmer
sentence="ان البحوث الحديثة تهتم اهتماما كبيرا باللسانيات الحاسوبية "
stemmed_words=[]
words=word_tokenize(sentence)
st=ISRIStemmer()
for word in words:
    stemmed_words.append(st.stem(word))
stemmed_words
```

```
['ان', 'بحث', 'حدث', 'هتم', 'هما', 'كبر', 'لسن', 'حسب']
```

Figure 9. Text stemming.

In addition, it is possible to remove diacritics, connectives, to normalize the text as mentioned in figure 10.

```
from nltk.stem.isri import ISRIStemmer
stemmer = ISRIStemmer()
stemmed_words = []
eg2=" أستاذ المُكتَبُ المعلمون والدرس"
words = word_tokenize(eg2)

for word in words:
    word = stemmer.norm(word, num=1)  # remove diacritics representing Arabic short vowels
    word = stemmer.pre32(word)        # remove length 3 and length 2 prefixes in this order
    word = stemmer.waw(word)          # remove connective 'و'
    word = stemmer.norm(word, num=2)  # normalize initial hamza to alif
    stemmed_words.append(word)
stemmed_sentence = " ".join(stemmed_words)
stemmed_sentence
```

```
'استاذ مكتب معلمون درس'
```

Figure 10. Text normalization.

4.6. POS tagging

A Part-Of-Speech Tagger (POS Tagger) scans text and assigns parts of speech to each word, such as noun, verb, adjective, etc. Figure 11. shows an example of POS tagging of the sentence (هذه اللقاءات مفيدة جدا)

```
text = "هذه اللقاءات مفيدة جدا"
words= word_tokenize(text)
tokens=nltk.pos_tag(words)
tokens
```

```
[('هذه', 'JJ'), ('اللقاءات', 'NNP'), ('مفيدة', 'NNP'), ('جدا', 'NN')]
```

Figure 11. Text tagging.

4.7. Wordnent

WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations [4]. Figure 12. shows the script displaying the synsets of the word (lion) in Arabic language.

```
from nltk.corpus import wordnet as wn
list = wn.synset('lion.n.01').lemma_names('arb')
for x in list:
    print (x)

```

```
أسد
لـيْث
مـلِك
مَـلِك
```

Figure 12. Arabic synsets of the word (lion) using Wordnet.

If we replace (lion) by (men) we get the result in Figure 13.

```
الأيْـدِي_الـعـامِـلَـة
الـقُـوَّة_الـبَـشَـرِيَّـة
أيْـدِي
قُـوَّة_عـامِـلَـة
```

Figure 13. Arabic synsets of the word (men) using Wordnet.

5. Conclusion

In this paper, we presented some functionalities of NLTK in python. All presented functions are applied on Arabic language. We have chosen the most important functionalities in any NLP application like tokenization, stemming, stop words removing, POS tagging, and the use of Wordnet.

References

[1] https://www.nltk.org/

[2] Natural Language Processing with Python, by Steven Bird, Ewan Klein *and*

Edward Loper, 2014.

[3] https://dzone.com/articles/nlp-tutorial-using-python-nltk-simple-examples

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

Le Traitement Automatique de la Langue Arabe : Un retour d'expérience de l'université de Tlemcen

Mohammed El Amine ABDERRAHIM, Mohammed. Alaeddine ABDERRAHIM,

Ali Benabdallah, Fatima Zahra BERRAKEM

Université de Tlemcen,
Laboratoire de Traitement Automatique de la Langue Arabe (LTALA)
BP 230 Chetouane, Tlemcen, Algérie
med.amine.abderrahim@gmail.com; abderrahim.alaa@yahoo.fr;
benabdallah.a13@gmail.com; fatima.berrakem@gmail.com

Résumé

Les travaux de recherche que nous sommes entrains d'entreprendre peuvent se résumer dans une question fondamental en TALN qui concerne la représentation du contenu: il s'agit dans nos travaux d'extraire des connaissances contenues dans un texte et de les modéliser.

Les principaux points traités dans notre travail de recherche concernent : la modélisation linguistique propre à la langue arabe, la recherche d'information de contenus en arabe (recherche sémantique, reformulation de la requête), l'annotation des corpus arabe, la construction semi-automatique d'ontologies à partir de textes arabes, la segmentation thématique des textes arabe pour la recherche d'information, et la construction de ressources pour le TALN arabe (base de données linguistique, base de données lexicale).

Les résultats obtenus dans le cadre de nos travaux sur le TALN arabe, problématique centrale dans le cadre du Web sémantique sont encourageants, ils nous ont montré qu'il existe une forte complémentarité entre les outils développés et que de nombreux problèmes restent ouverts.

Nous espérons finalement par le modeste travail réalisé, apporter une contribution significative au projet TALN arabe et qu'il permettra avec les travaux futurs, d'aboutir à la création concrète d'une spécialité Master option TALN arabe dans l'université de Tlemcen.

Mots clés : TALN Arabe; Recherche d'Information Arabe; Reformulation de la requête; Réinjection de la pertinence automatique; WordNet Arabe; Construction des ontologie ;

Keywords : Arabic NLP; Arabic Information Retrieval; Query Reformulation; Relevance Feedback ; Arabic WordNet ; Ontology construction ;

الكلمات المفتاحية:

المعالجة الآلية للغة الطبيعية، البحث عن المعلومات، إعادة صياغة الطلبيـــات، حقن الملائمة، ووردنات عربي، بناء الأنطولوجيات،

1. Introduction

Le Traitement Automatique des Langues Naturelles (TALN) est un domaine à la frontière de la linguistique et l'informatique, il a pour objectif de développer des logiciels capables de traiter de façon automatique des données linguistiques exprimées dans une langue naturelle donnée et pour une application bien définie. Cet objectif passe nécessairement par l'explicitation des règles de la langue puis les représenter dans un formalisme calculable et enfin les implémenter à l'aide des programmes informatiques.

Parmi les applications les plus connues du TALN, nous pouvons citer :

- la traduction automatique ;
- la correction orthographique ;
- la recherche d'information et la fouille de textes ;

- le résumé automatique ;
- la génération automatique de textes ;
- la synthèse de la parole ;
- la reconnaissance vocale ;
- la reconnaissance de l'écriture manuscrite ;

Le travail de recherche que nous présentons dans cette synthèse constitue une contribution modeste à l'effort qui vise à doter la langue arabe des outils performants de traitement automatique. L'enjeu est double :

- d'un point de vue culturel, les langues qui ne seront pas informatisées, risquent d'être exclues des médias modernes de production et de diffusion de l'information;
- d'un point de vue économique, le marché des applications pour le TALN arabe connaît une forte croissance et les gains en productivité dans de nombreux secteurs ne sont pas à démontrer.

On distingue deux aspects différents pour le TALN écrite : l'analyse et la génération. L'analyse se compose d'une suite de traitements (on parle souvent de niveaux d'analyses : morphologique, syntaxique, sémantique et pragmatique), elle consiste à construire une représentation formelle du texte en entrée, cette représentation doit être facile à manipuler par la machine. Par ailleurs la génération consiste à générer des textes à partir d'une représentation interne. Il s'agit de la fonction inverse de celle de l'analyse, mais elle n'est pas forcément obtenue en inversant le processus. La génération de textes apparaît dans des applications comme la traduction automatique de texte, le résumé automatique de texte, la génération des comptes rendus boursiers ou météorologiques, etc.

Dans le cadre de nos travaux de recherche actuels, nous nous positionnons dans le cadre de l'analyse, nous nous intéressons plus

particulièrement à un aspect fondamental du TALN arabe écrite à savoir : la reconnaissance et l'extraction des unités de sens qui composent un texte.

Le traitement automatique des langues se heurte à deux difficultés :

- L'ambiguïté de la langue : elle concerne les différents types d'ambiguïté propres à chaque niveau d'analyse. On parle souvent d'ambiguïté morphologique, d'ambiguïté syntaxique, d'ambiguïté sémantique, et d'ambiguïté pragmatique.

- La complexité des connaissances qui doivent être mises en œuvre à tous les niveaux d'analyse.

En plus de ces difficultés, nous pouvons résumer les problèmes liés au TALN arabe, dont nous devons prendre en compte, dans les points suivants :

- la voyellation multiple,
- la structure complexe du mot graphique arabe (phénomènes d'agglutinations qui caractérisent la langue.
- L'ordre des mots est relativement libre dans une phrase arabe (Verbe + Sujet + Complément ; Verbe + Complément + Sujet ; Complément + Verbe + Sujet).
- Le traitement des cas particuliers : traitement de la 'hamza'', traitement de la

  'Shedda', traitement de l'altération de la forme du mot.
- Le traitement des racines analogues ne donnant pas lieu à des dérivations analogues.
- Le traitement de la racine d'un mot issu d'une racine anormale.
- Le traitement des mots homographes (une même chaîne de caractère qui suivant le contexte recouvre deux notions différentes) Exemple : le mot في : verbe impératif ou préposition.

Reconnaître et extraire les unités de sens constitue donc un enjeu fondamental dans le cadre des applications pour le TALN arabe. En effet, la qualité de ces applications dépend largement de la qualité de reconnaissance de ces unités. Il s'agit donc, dans nos travaux, de développer des modèles et par conséquent des outils pour la reconnaissance et l'extraction des unités de sens pour la langue arabe écrite, facilement réutilisable par les applications de TALN arabe et non lié à un domaine particulier.

Il faut noter que la reconnaissance des unités de sens ne constitue pas une fin en elle-même, mais plutôt un préalable nécessaire à diverses applications.

Nos travaux s'inscrivent dans le cadre général d'un projet scientifique développé au Laboratoire de Traitement Automatique de la Langue Arabe (LTALA) à l'université de Tlemcen.

Le LTALA s'est constitué autour d'un projet principal qui est la mise au point de logiciels pour le TALN arabe. Dans cette optique, les activités portent essentiellement sur deux axes :

- d'une part, la modélisation linguistique propre à la langue arabe,
- d'autre part, la conception et la réalisation de logiciels pour divers domaines d'applications.

Dans ce contexte, on doit disposer d'un ensemble d'outils permettant de faciliter la mise au point des analyseurs et des dictionnaires. Ces outils peuvent nous servir pour :

- la mise au point des grammaires,
- la construction et la mise à jour des dictionnaires,
- la validation des données linguistiques.

Dans ce qui suit nous allons décrire les apports de nos travaux de recherches par rapport à ces axes de recherche.

2. Modélisation linguistique propre à la langue arabe

L'objectif de ce travail de recherche est de construire un modèle de base pour le traitement automatique de l'arabe. Ce modèle repose sur une modélisation des unités linguistiques signifiantes. Pour la validation nous avons réalisé concrètement un analyseur morphologique de l'arabe écrit voyellé ou non. Cet objectif comprend donc deux phases :

- l'élaboration d'un modèle linguistique pour le traitement (on parle souvent de modélisation linguistique),
- la validation du modèle construit par la réalisation et l'expérimentation d'un analyseur morphologique pour les textes arabe. L'analyseur ainsi construit sera utilisé et réutilisable pour de nombreuses applications de l'équipe. La polyvalence et le caractère réutilisable de cet analyseur justifié l'effort nécessaire à fournir pour son développement.

La modélisation linguistique consiste à classer les mots de la langue selon deux aspects, le premier est purement syntaxique en revanche le second est lexical. Chaque classe représente une catégorie grammaticale ou une classe syntaxique. Un ensemble de variables grammaticales est associé à chaque classe syntaxique. Ces variables représentent les traits linguistiques associés à ces classes. La démarche de construction de notre modèle est relativement inspirée des travaux de modélisation réalisés pour le français. En effet, sur la base de l'organisation que nous avons adoptée, nous avons proposé dans [1] un modèle en classes (classe dans le paradigme objet) d'une forme graphique arabe pour le TALN arabe (voir figure 1).

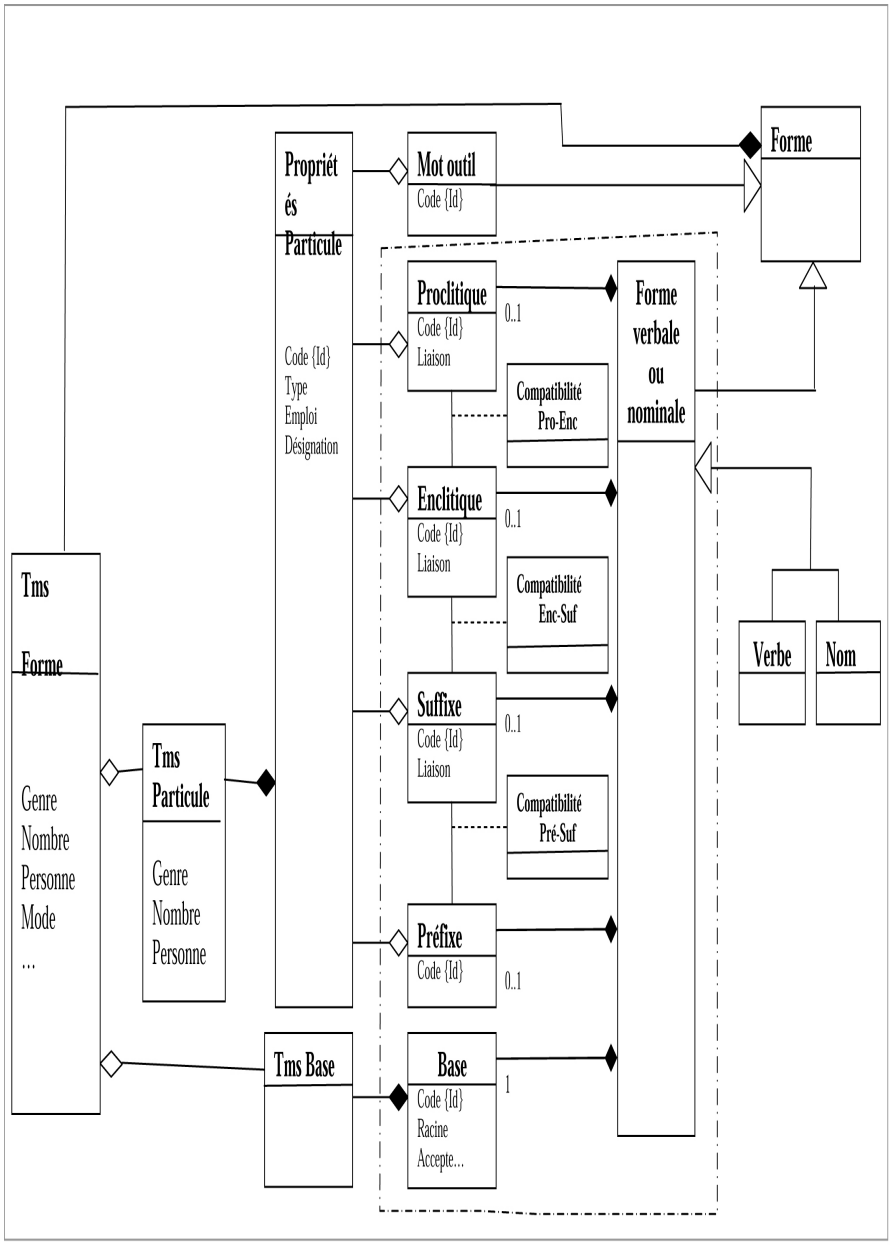


Figure 1 *Modèle d'une forme graphique (en UML) pour le traitement automatique de l'arabe voyellé ou non* [1]

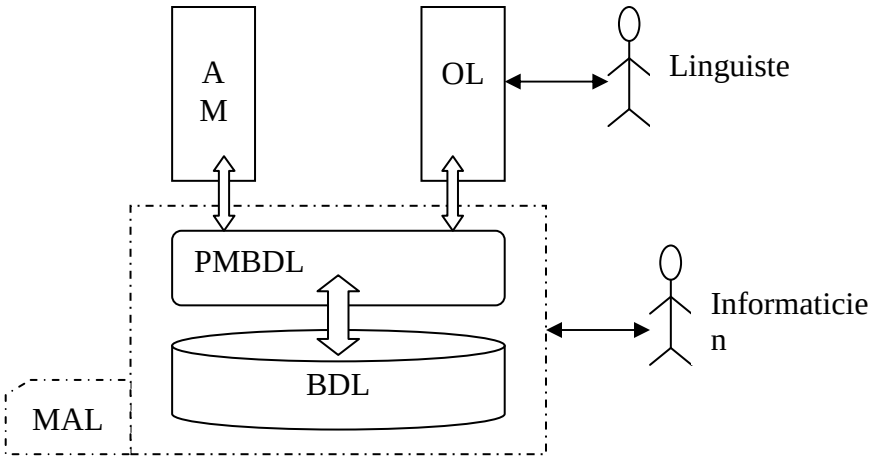
7

L'Analyseur Morphologique (AM) développé est basé sur « MALA » ; un framework que nous avons proposé dans [5], il nous a servi comme une plateforme pour le développement des applications pour le TALN arabe. MALA repose sur deux composantes (couches) principales (voir figure 2) :

- une Base de Données Linguistique (BDL[1]) intégrant toutes les données linguistiques propres à la langue arabe. Conceptuellement la BDL est représentée par un modèle en classes (classe dans le paradigme objet). Par ailleurs son implémentation est réalisée par un ensemble de table dans le modèle relationnel, parmi ces tables on trouve entre autres la table des bases nominales, des bases verbales, des mots outils, des clitiques, des affixes, des compatibilités entre clitiques…
- Un ensemble de Primitives ou de méthodes de base pour la Manipulation de la BDL (PMBDL).

Outre l'AM nous avons réalisé un outil (OL) destiné aux linguistes, il permet de faire la mise à jour de la BDL d'une manière très simple.

[1] La réalisation de cette base à fait l'objet d'un Projet CNEPRU agrée pour trois ans à partir de l'année 2010, dont le titre est: « Construction d'une base de données linguistique pour le traitement automatique de la langue arabe »



AM	**A**nalyseur **M**orphologique
OL	**O**utils pour **L**inguiste
PMBDL	**P**rimitives de base pour la **M**anipulation de la **BDL**
BDL	**B**ase de **D**onnées **L**inguistique
MALA	Framework pour TALN arabe

Figure 2 *Architecture générale du framework MALA[5]*

Le développement de MALA présente de nombreux avantages comme par exemple :

- La séparation (entre les données linguistiques et les programmes qui les manipulent),

- la réutilisation (plateforme commune pour toutes les applications de TALN arabe),

- la normalisation des développements (permettre de construire toutes les applications avec les mêmes normes, technologies,…),

- l'extension,

- et la facilité de la maintenance.

Comparé notre analyseur à d'autres est une tâche très difficile et reste toujours subjective du moment où il n'existe pas de standard en terme de critères pour pouvoir faire cette confrontation. Chaque analyseur possède sa propre sortie et cible bien une application spécifique. Notre analyseur diffère des analyseurs existant pour la langue arabe par le fait qu'il :

- ne cible pas une application spécifique,
- ne réalise pas de prétraitement du texte en entrée,
- peut être utilisé en analyse comme en génération,
- peut être utilisé pour les textes voyellé ou non,
- repose sur un modèle sure et cohérent.

En plus de ces avantages, la réalisation de cet analyseur est simplifiée (seulement un algorithme de segmentation des formes et un autre pour la validation des segments) par le fait qu'il repose sur un framework qui implémente le modèle linguistique développé.

3. La recherche d'information de contenus en arabe

Un Système de Recherche d'Information (SRI) repose sur les trois fonctions suivantes : stocker, organiser (indexer) et rechercher des données (en réponse à des requêtes utilisateurs). Il fait appel à trois types de connaissances:

- les connaissances sur les documents : ils regroupent les informations sur le contenu et le contenant ;
- les connaissances sur les utilisateurs ;
- et les connaissances sur le domaine d'application : ils permettent d'organiser les différents termes utilisés, on retrouve par exemple les dictionnaires, les thesaurus…

Dans le cadre de la recherche d'information pour les textes en langue arabe, la récupération de mots clé est jugée insuffisante, car les termes utilisés dans la requête peuvent présenter par rapport aux documents de la base, des différences sur plusieurs plans, par exemple :

- des variations morphologiques comme dans « مدرسة » et « مدرستان », « خيل » et « خيول » ;
- des variations lexicales (on utilise pour le même sens des mots différents) comme dans le cas dans « فرس » et « خيل » ;
- des variations sémantiques comme dans le cas de « الحجر : مرادف الصخر » et « الحجر : أنثى الخيل ».

L'utilisation des ontologies peut constituer une solution pour résoudre le problème des variations lexicales et sémantiques, ce qui a pour conséquence l'amélioration des résultats de la recherche. Par ailleurs l'utilisation d'un analyseur morphologique peut suffire pour résoudre le problème des variations morphologiques.

Les ontologies peuvent être utilisées à différents niveaux dans un SRI. Elles peuvent participer au processus d'indexation des documents et requêtes, nous parlons alors d'indexation sémantique ou conceptuelle. Elles peuvent également contribuer à faire l'appariement entre les documents et la requête. Enfin les ontologies peuvent aider à la formulation du besoin de l'utilisateur qui peut être formulé sous forme de requête le plus souvent en langage libre (langage proche du langage naturel). Il faut noter toutefois que la formulation de la requête est un processus très important car de sa qualité dépend la qualité des documents restitués par le SRI.

Dans ce contexte et dans le cadre des SRI pour les textes Arabes, nous avons procédé à l'évaluation des performances de l'indexation conceptuelle. A cet effet l'ontologie lexicale WordNet arabe a été utilisée. La figure 3 montre l'architecture proposée.

Les expérimentations réalisées sur un corpus de texte Arabe sont décrit dans [2,3], ils nous ont permet de mesurer l'apport de cette approche de reformulation de requête dans un SRI Arabe.

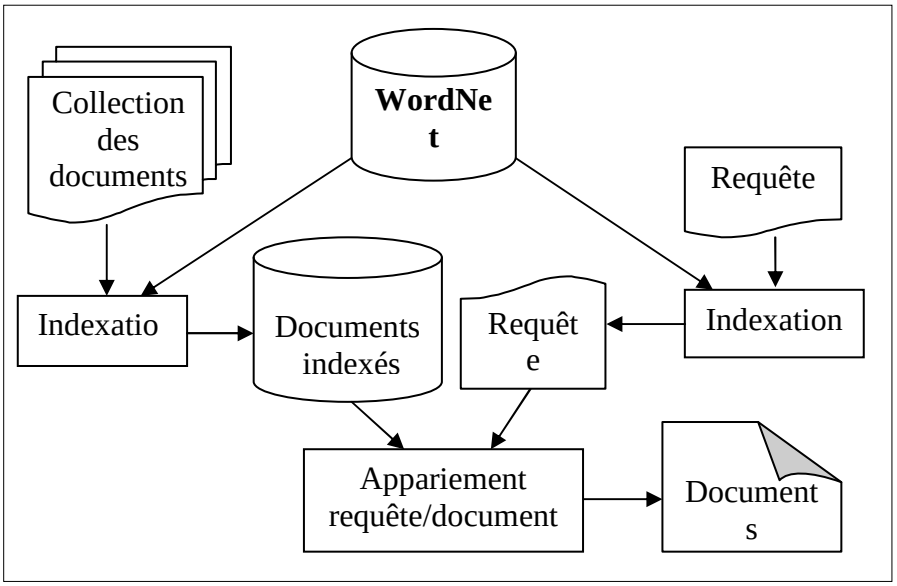


Figure 3 *Architecture du SRI basée sur une indexation conceptuelle des documents et des requêtes [2,3]*

Nous avons aussi examiné l'approche de reformulation de la requête dans un SRI pour les textes arabe. Dans cette optique, il existe plusieurs approches, en effet, nous distinguons :

- La reformulation par l'utilisation d'une représentation du domaine de recherche.
- La reformulation par l'utilisation des relations sémantiques de bases terminologiques.
- La reformulation par l'utilisation d'un espace d'information structuré et construit automatiquement. L'interrogation se fait par navigation (query by navigation).
- La reformulation par l'utilisation des points de vue. Les points de vue représentent des besoins élémentaires en information par exemple : causalité, définition, citation, thème,…

L'examen de ces approches, permet de dégager trois grandes démarches pour la reformulation de la requête (voir figure 4) :

- L'utilisation des ressources externes : consiste à utiliser les ressources externes comme les ontologies ou les thésaurus pour trouver des termes similaires à la requête initiale.

- L'analyse globale : cette approche consiste à analyser tout l'ensemble des documents de la collection pour extraire les termes pertinents à ajouter à la requête initiale. Deux techniques existent: le thesaurus de similarité (similarity thesaurus) et le thésaurus statistique (statistical thesaurus).

- L'analyse locale : les documents retournés en réponse à une requête sont analysés pour extraire des termes pertinents qui serviront à étendre la requête. Deux techniques sont alors proposées dans la littérature :

- La classification locale (local clustering) : consiste à construire une matrice d'association qui quantifie les relations de corrélation entre les termes issus de l'ensemble des documents retournés en réponse à la requête initiale. Selon la méthode de construction des relations de corrélation on distingue trois types de clusters : association clusters, metric clusters et scalar clusters. Nous développons dans la suite de cet article cette technique et nous implémentons le premier type de culster (association clusters).

- L'analyse du contexte local : consiste à utiliser les concepts à la place de mot-clés pour représenter les documents.

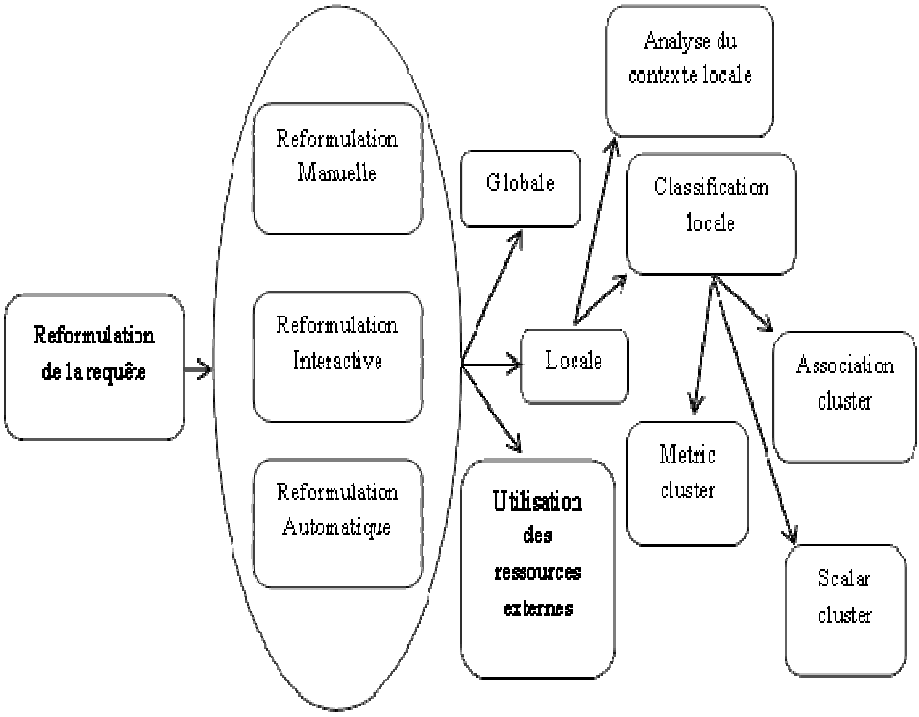


Figure 4 *Les approches pour la reformulation de la requête [4,6]*

Les contributions que nous avons proposées dans cet axe de recherche sont articulées autour de l'évaluation de l'apport réel de ces approches dans un SRI arabe afin de déterminer la meilleure approche à intégrer dans un SRI arabe. En effet, dans [4, 6], plusieurs architectures ont été étudiées:

- **Reformulation de la requête par utilisation d'une ressource externe avec le moteur de recherche Google** (voir figure 5).

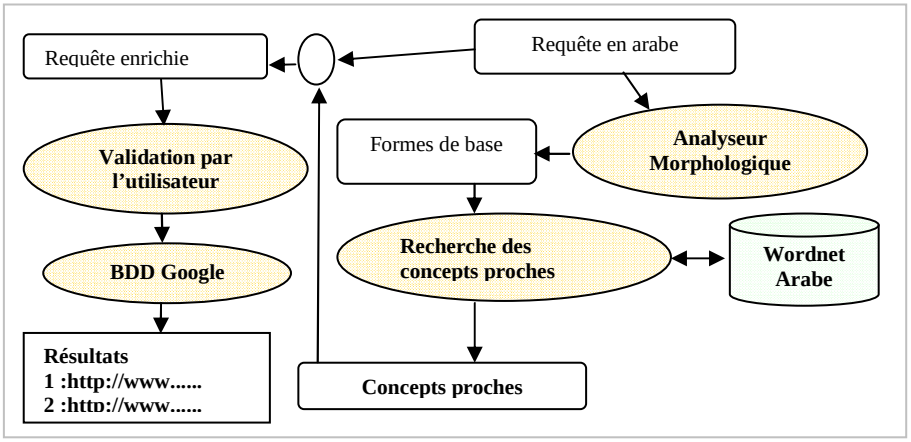


Figure 5 *Architecture de l'interface de recherche pour la reformulation des requêtes (utilisation d'une ressource externe avec le moteur de recherche Google)[7,8,9,10]*

L'évaluation de l'apport réel de l'enrichissement de la requête arabe dans le cas de l'architecture étudiée de la figure 5 est une tâche très délicate et demande beaucoup d'investigations, c'est pour cette raison que nous avons orienté notre évaluation vers l'utilisation d'un corpus fixe avec un moteur de recherche sous la forme d'une API (Lucene).

- **Reformulation de la requête par utilisation d'une ressource externe avec l'API Lucene** (voir figure 6).

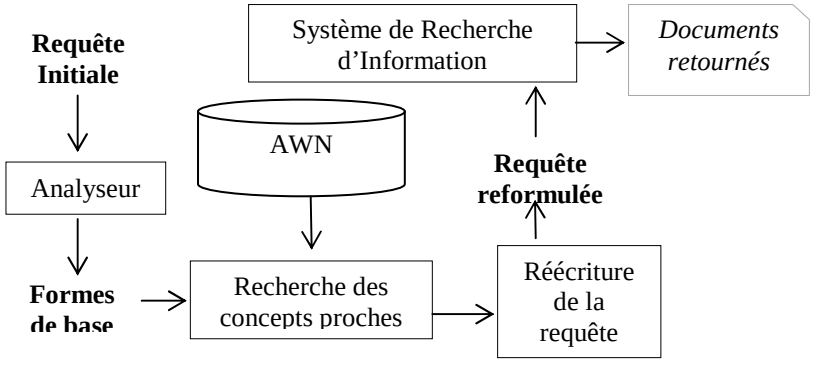


Figure 6 *Architecture de l'interface de recherche pour la reformulation des requêtes (utilisation d'une ressource externe et l'API Lucene) [7,8,9,10,13]*

Les résultats obtenus dans le cadre de cette architecture (figure 6) nous ont permet d'une part de confirmer que cette technique de reformulation améliore considérablement le rappel, d'autre part de mesurer l'apport (6%) d'une telle approche dans l'amélioration des performances globales d'un SRI Arabe.

- **Reformulation de la requête par analyse locale (local clustering)** (voir figure 7). Les résultats obtenus dans le cadre de cette architecture nous ont permet de mesurer l'apport d'environ (9%) d'une telle approche dans l'amélioration des performances globales d'un SRI Arabe.

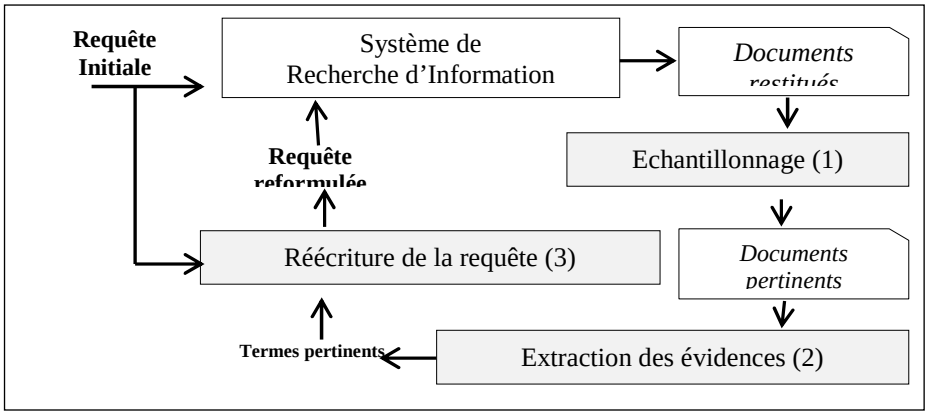


Figure 7 *Architecture de l'interface de recherche pour la reformulation des requêtes (analyse locale) [6]*

Finalement, dans cet axe de recherche, les résultats obtenus de l'expérimentation des différentes approches pour la recherche d'information, nous ont permet d'une part, de confirmer l'hypothèse de départ, à savoir, l'amélioration des performances du SRI Arabe. D'autre part, d'ouvrir la voie pour tester d'autres techniques avec les mêmes données de ces expérimentations pour déterminer la meilleure approche afin de l'intégrer dans un SRI Arabe.

4. Annotation des corpus

Un corpus annoté peut servir aussi bien aux informaticiens (à des fins d'évaluation de parseur, d'entraînement de parseur, d'extraction de lexique ou de grammaire) qu'aux linguistes ou aux psycholinguistiques (pour connaître les distributions fines de certains mots ou de certaines catégories, ou connaître la fréquence relative de certaines constructions etc.).

Dans cette optique, nous avons proposé, dans le cadre d'un projet de master [16], de réaliser une plateforme pour l'annotation manuelle des corpus textuelle pour la langue arabe. Cette plateforme possède des interfaces utilisateurs qui sont très simple à utiliser par les non informaticien. Cependant, des améliorations

et des évolutions sont prévues, qui devraient nécessiter encore plusieurs années de développement dans un premier temps. Les améliorations possibles concernent l'intégration des méthodes d'analyse automatique et de fouille du contenu textuel arabe pour l'extraction d'information afin d'aider l'annotateur.

5. Construction semi-automatique d'ontologies à partir de textes arabes[2]

La construction d'ontologies à partir de textes constitue un sous-domaine à part entière de l'ingénierie des ontologies. Dans le contexte du Web sémantique, ces ontologies servent essentiellement à l'annotation sémantique de ressources et à la structuration de bases de connaissances.

Les travaux dans ce domaine existent déjà pour d'autres langues comme l'anglais ou le français, malheureusement pour l'arabe les choses ne font que commencer.

[2] Projet CNEPRU agrée pour trois ans à partir de l'année 2016

Cette problématique constitue en effet un nouvel enjeu important pour le TALN. Les systèmes automatisés de traitement de l'information fonctionnant dans des domaines de connaissances spécialisés ne peuvent être efficaces que s'ils reposent sur des ressources terminologiques et/ou ontologiques, construites pour le domaine et l'application concernés.

Dans un système de TALN nous avons besoin de deux types de sources de connaissances, l'une organise les connaissances linguistiques concernant la langue et l'autre stocke les connaissances générales du monde (extralinguistiques). L'objectif de cet axe de recherche est de concevoir différentes participations pratiques et théoriques pour la création d'une ontologie, plus précisément une ontologie lexicale dans une perspective d'utilisation en TAL.

La construction manuelle d'une ontologie est une tâche très difficile, complexe et qui nécessite beaucoup de temps et de ressources. Dès lors, le recours à des méthodes automatiques ou semi automatiques est devenu indispensable, toutefois le recours aux experts pour la validation des résultats au cours de ce processus de la création permet d'aboutir à une ontologie plus précise.

Nos recherches dans le cadre de cet axe ne sont encore qu'à leurs débuts. Un premier travail réalisé [12, 15] a consisté à commencer par la collecte et le prétraitement d'un corpus à travers la normalisation, puis la suppression des mots vides et la lemmatisation; Ensuite, pour extraire les termes de l'ontologie, une méthode statistique pour extraire des termes simples et complexes appelée « méthode des segments répétés » est appliquée. Pour sélectionner les segments avec un poids suffisant, nous avons appliqué deux filtres : un filtre de pondération TF-IDF (Term Frequency-Inverse Document Frequency) et un filtre coupant. Pour relier ces termes par des relations

sémantiques, nous avons proposé une méthode d'apprentissage automatique des marqueurs linguistiques à partir du texte. Cette méthode nécessite un ensemble de paires de relations, qui sont extraites à partir de deux ressources externes: un dictionnaire arabe de synonymes et d'antonymes et une base de données lexicale Arabe.

Les résultats obtenus sont encourageants, néanmoins beaucoup d'investigations restent à faire. Par ailleurs le travail réalisé nous a permis de découvrir l'importance d'acquisition (semi) automatique des ressources ontologiques pour la langue arabe.

6. Désambigüisation du sens des textes arabe

Un mot peut avoir plusieurs sens selon son contexte d'utilisation, à cet effet, la désambigüisation devient une tâche importante afin de lever l'ambigüité des mots en question. Dans la littérature, nous trouvons des méthodes qui sont utilisées pour estimer le sens le plus pertinent du mot ambigu. Cette estimation est basée sur le calcul de la proximité (sémantique score de cohérence) entre le contexte actuel (contexte d'apparition du mot ambigu), et les différents contextes d'utilisation de chaque sens du mot.

Nos recherches dans le cadre de cet axe ne sont encore qu'à leurs débuts. Un premier travail réalisé dans [11] était d'utiliser la mesure de densité conceptuelle et WordNet Arabe pour désambiguïser les mots d'une requête pour la recherche d'information. L'évaluation de l'apport réel de cette approche dans l'amélioration de l'accès à l'information dans un système de recherche d'information arabe a fait l'objet d'une communication dans les travaux de [11]. Les résultats obtenus sont encourageants.

7. Segmentation thématique des textes arabe pour la recherche d'information

La segmentation thématique du texte TTS (Topical Text Segmentation) est une issue importante dans les applications de recherche d'information. Elle consiste à diviser les textes en segments, chacun d'eux correspondant à un thème différent. Une application directe de TTS consiste à extraire des segments appropriés pour répondre à une requête, au lieu de fournir des textes complets, dans lesquels l'utilisateur ne trouve pas facilement les quelques phrases satisfaisant son besoin en information. Nous considérerons dans ce travail, qu'un document n'est pas un segment thématique, et que retourner un document non segmenté constituera un échec dans le traitement de la tâche de segmentation thématique.

Dans cet axe de recherche nous proposons d'évaluer une approche automatique de segmentation thématique des textes Arabes. Cette dernière est basée sur une représentation conceptuelle du contenu des documents déduite de la base de données lexicale WordNet Arabe (AWN). L'avantage de la représentation conceptuelle est de réduire les effets synonymiques du vocabulaire.

L'évaluation de l'apport réel de cette approche dans l'amélioration de l'accès à l'information dans un SRI arabe a fait l'objet d'une communication dans les travaux de [14].

8. Construction d'un WordNet Arabe « ARAWORD »[3]

Nous avons une version limitée d'un WordNet Arabe. Ce dernier est une base de données lexicale librement disponible pour l'arabe

[3] Projet CNEPRU agrée pour trois ans à partir de l'année 2013

standard. Cette base de données suit la conception et la méthodologie du Princeton WordNet pour l'anglais (http://WordNet.princeton.edu/) et d'EuroWordNet pour les langues européennes (http://www.illc.uva.nl/EuroWordNet/). Sa structure est celle d'un thésaurus, il est organisé autour de la structure des synsets, c'est-à-dire des ensembles de synonymes et de pointeurs décrivant des relations vers d'autres synsets. Chaque mot peut appartenir à un ou plusieurs synsets, et à une ou plusieurs catégories du discours. Ces catégories sont au nombre de quatre : nom, verbe, adjectif et adverbe. Il faut noter toutefois que WordNet Arabe est une des rares ressources pour la langue générale arabe disponible en ligne. Il compte actuellement 11269 synsets et 23481 mots (http://www.lsi.upc.edu/~mbertran/).

Cette version de WordNet arabe, en plus d'être incomplète, elle présente l'inconvénient d'être seulement une traduction vers l'arabe de celle de Princeton WordNet. Donc, cette version ne prend pas en compte les particularités de la langue Arabe. Alors pourquoi ne pas faire l'effort de fabriquer une vraie base de données lexicale à large couverture de la langue arabe. Nous espérons par ce travail, apporter une contribution significative au projet de création d'une ontologie lexicale pour la langue arabe et qu'il permettra avec les travaux futurs, d'aboutir à la création concrète d'un WordNet arabe.

9. Conclusion

En guise de conclusion, les travaux de recherche que nous sommes entrains d'entreprendre peuvent se résumer dans une question fondamental en TALN qui concerne la représentation du contenu: dans un texte, qu'est-ce qui est porteur de sens ? Autrement dit, il s'agit dans nos travaux d'extraire des connaissances contenues dans un texte et de les modéliser.

Du moment où le sens n'est pas réductible à la somme de constituants d'une phrase, alors l'unité de travail n'est plus uniquement la phrase, mais le texte nécessitant ainsi pour sa compréhension, la prise en compte de l'ensemble des éléments d'une situation d'énonciation. Du point de vue recherche d'information, la question n'est plus comment indexer automatiquement du contenu, mais plutôt comment satisfaire les besoins en information d'un utilisateur en lui offrant une interface d'interrogation en langue naturelle, autrement nous basculons d'un paradigme orienté tâche vers un autre orienté beaucoup plus utilisateur, ce qui a pour conséquence, la prise en compte de nouvelles connaissances (connaissances de l'utilisateur, de ses croyances, de la situation…) dans le cadre TALN en particulier ou du web sémantique en générale.

Les principaux points traités dans notre travail de recherche concernent :

- La modélisation linguistique propre à la langue arabe.
- La recherche d'information de contenus en arabe (recherche sémantique, reformulation de la requête).
- L'annotation des corpus arabe
- La construction semi-automatique d'ontologies à partir de textes arabes.
- La segmentation thématique des textes Arabe pour la recherche d'information.
- La construction de ressources pour le TALN arabe (base de données linguistique, base de données lexicale)

Les résultats obtenus dans le cadre de nos travaux sur le TALN arabe, problématique centrale dans le cadre du Web sémantique, nous ont montré qu'il existe une forte complémentarité entre les outils développés et que de nombreux problèmes restent ouverts. En effet deux projets CNEPRU ont été concrétisés :

- Projet CNEPRU « Construction d'une base de données linguistique pour le traitement automatique de la langue arabe » agrée pour trois ans à partir de l'année 2010.

- Projet CNEPRU « Construction d'un WordNet Arabe « ARAWORD » agrée pour trois ans à partir de l'année 2013.

Les perspectives de nos travaux sont nombreuses, nous citons :

- Intégration des outils développés dans un schéma d'exploitation complet.
- Réutilisation des avancées dans le cadre du web sémantique.
- Exploration d'approches développées pour d'autres langues.
- Réflexions sur la poursuite des travaux sur un domaine restreint.

Nous espérons finalement par le modeste travail réalisé, apporter une contribution significative au projet TALN arabe et qu'il permettra avec les travaux futurs, d'aboutir à la création concrète d'une spécialité Master option TALN arabe dans l'université de Tlemcen.

10. Bibliographie

[1] Mohammed El Amine ABDERRAHIM, F. BREKSI REGUIG; *A morphological analyzer for vocalized or not vocalized Arabic language*; Journal of Applied Sciences 8(6): pp 984-991, 2008; ISSN 1812-5654.

[2] Mohammed El Amine ABDERRAHIM; *Vers la recherche d'information de contenus en arabe fondée sur l'enrichissement des requêtes* ; SIIE'2009 : 2ème Conférence Internationale "Systèmes d'Information et Intelligence Economique" SIIE 2009, Ecole Supérieur de Commerce Electronique, université Nancy ; Hammamet – Tunisie, 12-14 Février 2009, Proceedings IHE éditions, ISBN 9978-9973-868-21-3 ; pp 598-607. http://siie2009.loria.fr/

[3] Mohammed El Amine ABDERRAHIM; *Apport des ontologies dans un système de recherche d'informations arabe ;* 3ème Colloque international en traductologie et TAL, Oran Algérie ; 17-18 janvier 2010.

[4] Mohammed El Amine ABDERRAHIM, ABDERRAHIM Med Alaeddine; *Using Arabic Wordnet for query expansion in information retrieval system* ; IEEE,The Third International Conference on Web and Information Technologies, 16-19 June, 2010, Marrakech – Morocco. ISBN 978-9954-9083-0-3 http://www.ucam.ac.ma/icwit2010/html/index.php.

[5] Mohammed El Amine ABDERRAHIM; *Vers une base de données linguistique pour le traitement automatique de la langue Arabe ;* Colloque international sur le traitement du dictionnaire Arabe et le trésor de la langue Arabe, Université Hasiba Ben Bouali Chelef Algérie, Le 22 et 23 novembre 2011.

[6] Mohammed El Amine ABDERRAHIM, ABDERRAHIM Med Alaeddine; *Réinjection Automatique de la pertinence pour la Recherche d'Informations dans les textes Arabes* ; IEEE, 4th International Conference on Arabic Language Processing (CITALA), May 2–3, 2012, Rabat, Morocco. ISBN 978-9954-9135-0-5; pp 77-81 ; http://www.citala.org

[7] ABDERRAHIM Mohammed. Alaeddine, Mohammed El Amine ABDERRAHIM; *Using Arabic Wordnet for semantic indexation in information retrieval system*; IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 2, January 2013; pp 661-667; ISSN 1694-0784.

[8] Mohammed El Amine ABDERRAHIM; *Utilisation des ressources externes pour la reformulation des requêtes dans un système de recherche d'information* ; The Prague Bulletin of Mathematical Linguistics PBML, Number 99, avril 2013 ; pp 85–97 ; ISSN 0032-6585. (ISSN of the print version is 0032-6585 and of the online version 1804-0462)

[9] Mohammed El Amine ABDERRAHIM; *Query Reformulation Guided by External Resource for Information Retrieval*; International Journal of Information Science and Engineering, Vol. 7, No 4, 2013; pp 894 - 899; ISSN 1307-6892.

[10] Mohammed El Amine ABDERRAHIM; *Concept Based vs. Pseudo Relevance Feedback Performance Evaluation for Information Retrieval System*; International Journal of Computational Linguistics Research, Volume 4, Issue 4, December, 2013, Pages 149-158; ISSN 0976-416X.

[11] ABDERRAHIM Mohammed Alaeddine, ABDERRAHIM Mohammed El Amine et Chikh Mohammed Amine; *Using Arabic Wordnet for the Disambiguation of Users Queries in Information Retrieval System*; International NOOJ 2014 Conference, University of Sassari - Italy, 3-5 June, 2014; http://nooj2014.uniss.it/call.html

[12] Benabdallah Ali, ABDERRAHIM Mohammed El Amine et Chikh Mohammed Amine; *Automatic construction of ontologies from Arabic texts*; International NOOJ 2014 Conference, University of Sassari - Italy, 3-5 June, 2014; http://nooj2014.uniss.it/call.html.

[13] Mohammed Alaeddine ABDERRAHIM,· Med Dib, Med El Amine ABDERRAHIM, Med Amine Chikh ; Semantic indexing of Arabic texts for information retrieval system ; International Journal of Speech Technology ; Special Issue, Springer September 2015 ; pp 1-8 ; DOI 10.1007/s10772-015-9307-3 (http://link.springer.com/article/10.1007/s10772-015-9307-3). ISSN: 1381-2416 (print version) 1572-8110 (electronic version)

[14] Fatima Zahra BERRAKEM, Mohamed Amine ABDERRAHIM, Mohamed Amine CHIKH ; Les apports de la base de données lexicale WordNet Arabe (AWN) pour la segmentation thématique des textes Arabe ; Conférence Internationale sur le Traitement de l'Information Multimédia (CITIM'2015) 12-, Université de Mascara, Algérie, 13 Mai 2015.

[15] Ali Benabdallah, Mohammed Alaeddine ABDERRAHIM, Med El Amine ABDERRAHIM; Extraction of terms and semantic relationships from Arabic texts for automatic construction of an ontology; International Journal of Speech Technology ; Springer mars 2017 ; Volume 20 Number 2 mars 2017 Pages 289-296 ; DOI 10.1007/s10772-017-9405-5; ISSN: 1381-2416 (print version) 1572-8110 (electronic version) (http://link.springer.com/article/10.1007/s10772-017-9405-5).

[16] Bouhassoun Kouider, Lagha Mohamed, Med El Amine ABDERRAHIM ; Réalisation d'un outil pour l'annotation des corpus textuels ; Projet de fin d'étude Master en Informatique; Université Abou Bekr Belkaid Tlemcen, Juin 2010.