# Chapitre 3

# Implémentation

# deAraWord

# Chapitre 3 : Implémentation de AraWord

## 3.1 Introduction

Dans cette partie nous allons décrire le travail réalisé. Il comporte deux parties :

- Les données.
- Les programmes.

## 3.2 Description des données

Nous utilisons dans ce travail une base données*wordnet_ar*qui a le même schéma du WordNet AWN, elle contient les 12 tables suivantes : arabicstarters, author, authorship, config, conversion1, form, has_translation, item, link, mappings, starters, word.

**Description des tables utilisées**

**La table Item** : Cette table contient des informations sur le synset, les colonnes sont définies par :

| Nom de la colonne | Description |
|---|---|
| authorshipid | L'id de l'entrée associée dans la table theauthorship, elle donne des informations sur l'auteur, la date de création du synset, etc. |
| gloss | Une description du synset, parfois avec des exemples d'usage. |
| headword | Si le mot est un adjectif. |
| itemid | Un id unique assigné au synset. |
| name | Tête du synset |
| pos | Partie du Discours (Part of speech) assignée au synset |
| pwind | Le numéro du synset dans le WordNet de Princeton |
| source | La source de cet item |
| type | Soit un terme formel d'ontologie ou un synset du WordNet |

**La table Word**

Cette table contient des informations à propos des mots dans les synsets.

| Nom de colonne | Description |
|---|---|
| authorshipid | L'id de l'entrée associée dans la table theauthorship, elle donne des informations sur l'auteur, la date de création du synset, etc. |
| corpus | Nom du corpus |
| frequency | Fréquence dans le corpus |
| synsetid | L'id de la synset pour laquelle ce mot appartient. |
| value | La forme base du mot. |
| wordid | Un identifiant unique pour ce mot. |
| w_num | Le numéro du mot dans le WordNet de Princeton |

**La table Form**

Cette table contient des informations sur les différentes formes de mots, par exemple, les formes du pluriel, ainsi que les racines des mots.

| Le nom de la colonne | Description |
|---|---|
| authorshipid | L'id de l'entrée associée dans la table theauthorship, elle donne des informations sur l'auteur, la date de création du synset, etc. |
| form_case | Le cas |
| gender | Le genre |
| number | Le nombre |
| person | La personne |
| tense | Le temps |
| type | Le type de la forme du mot, par exemple « racine ». |
| value | La forme actuelle du mot. |
| wordid | L'id du mot (de la table Word) pour laquelle cette forme est attachée. |

**La table Link**

Cette table contient des liens entre les différents synsets ou mots.

| Nom de la colonne | Description |
|---|---|
| authorshipid | L'id de l'entrée associée dans la table theauthorship, elle donne des informations sur l'auteur, la date de création du synset, etc. |
| link1 | Id du premier synset/mot impliqué dans la relation |
| link2 | Id du secondsynset/mot impliqué dans la relation |
| type | La nature de la relation entre les deux items, hyponym, ontonym… |

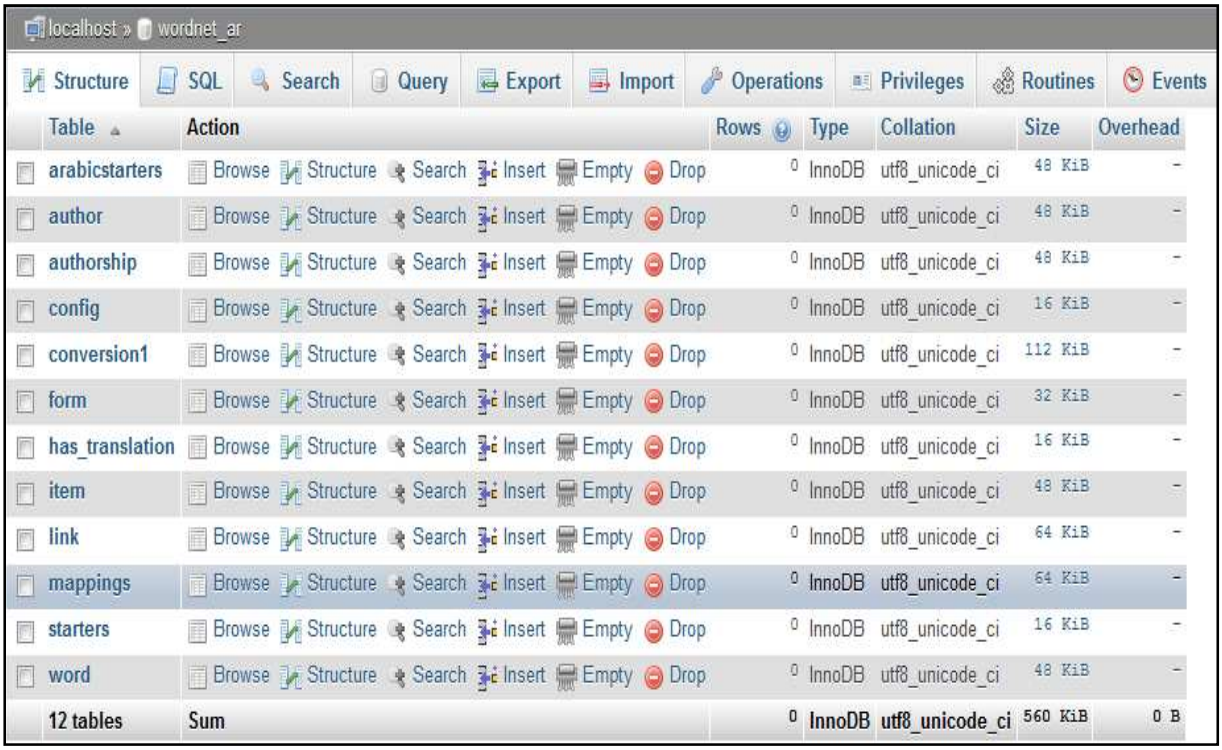
# 3.3 Les programmes

## Choix du Langage

Pour coder notre application nous avons utilisé NetBeans dans sa version 7.2.1.NetBeans est un Environnement de Développement Intégré (EDI) Open Source,qui supporte le développement sous plusieurs langages (Java, C/C++, etc).

## L'application

Les différentes tables sont créées grâce àWampServer version 2.2.WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement les bases de données, la figure suivante (figure 3.1) montre les différentes tables créées à l'aide de l'outil PHPMyAdmin.



localhost » wordnet_ar

Structure | SQL | Search | Query | Export | Import | Operations | Privileges | Routines | Events

| Table | Action | Rows | Type | Collation | Size | Overhead |
|---|---|---|---|---|---|---|
| arabicstarters | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 48 KiB | - |
| author | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 48 KiB | - |
| authorship | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 48 KiB | - |
| config | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 16 KiB | - |
| conversion1 | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 112 KiB | - |
| form | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 32 KiB | - |
| has_translation | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 16 KiB | - |
| item | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 48 KiB | - |
| link | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 64 KiB | - |
| mappings | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 64 KiB | - |
| starters | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 16 KiB | - |
| word | Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8_unicode_ci | 48 KiB | - |
| 12 tables | Sum | 0 | InnoDB | utf8_unicode_ci | 560 KiB | 0 B |

**Figure 3.1 Schéma de la base de données AraWordNet**

## Schéma générale de l'application

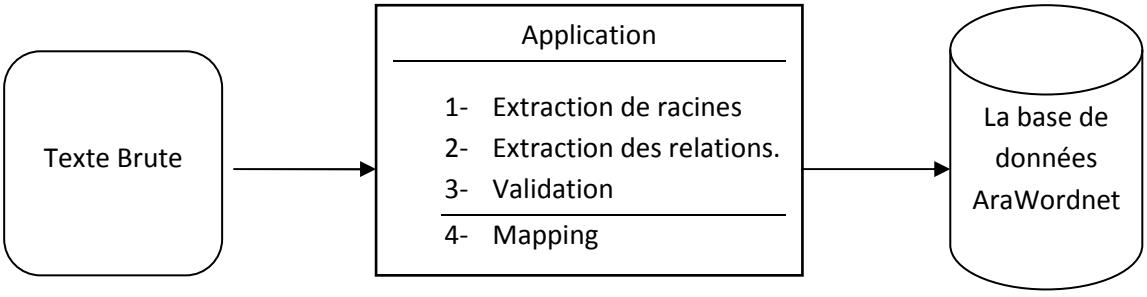



**Figure 3.2 L'architecture générale de notre application.**

L'application reçoit un fichier texte en entrée, elle exécute les opérations nécessaires du traitement et enregistre les résultats dans la base de données.

### a) Extraction des racines

Cette étape consiste à extraire les racines des mots du texte donné en entrée.

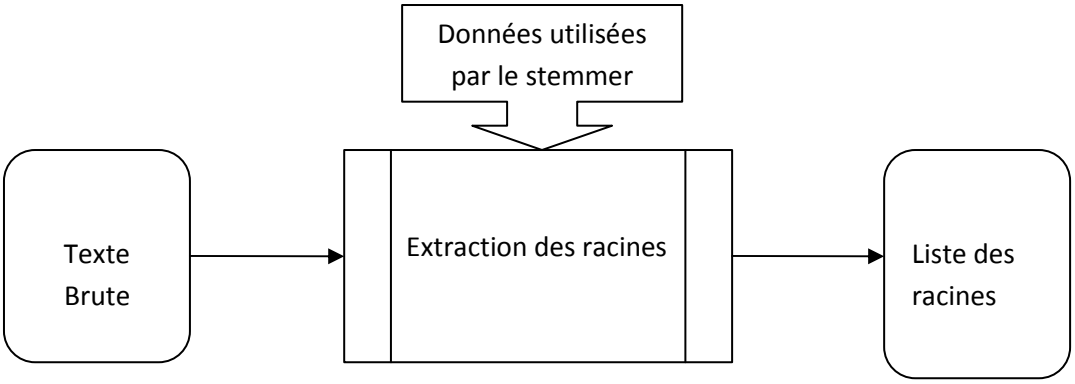



**Figure 3.3 Extraction des racines**

Pour l'extraction des racinesnous avons utilisé le stemmer de Khoujaqui est librement disponible sur le net.Il comprend plusieurs fichiers de données utiles, comme une liste de tous les caractères diacritiques, des caractères de ponctuation, des articles définis, et 168 mots vides, Il supprime d'abord les articles définis, les préfixes et suffixes, puis tente de trouver la racine. Si aucune racine n'est trouvée, alors le mot est laissé intact. Le stemmer supprime également les mots vides.**[Khoja, 1999]**

### b) Extractions des relations sémantiques

Cette étape consiste à extraire les relations entre les mots (racines).

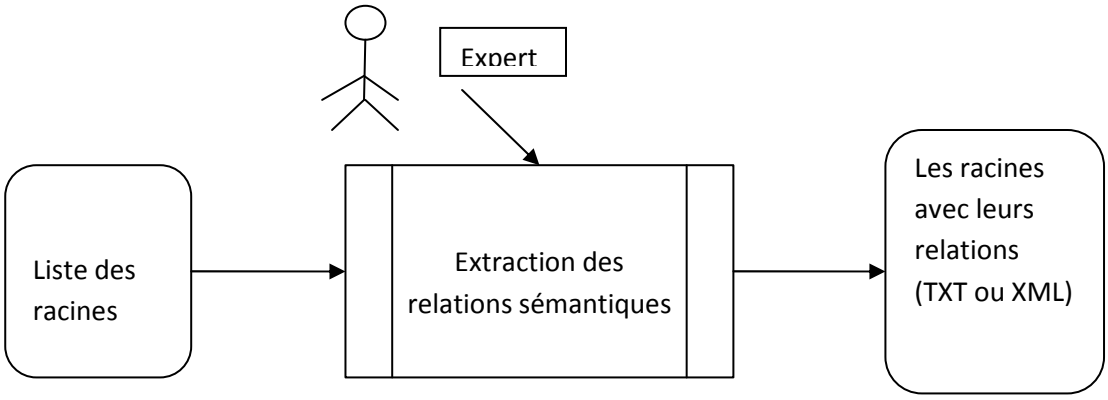



**Figure 3.4 L'extraction des relations sémantiques**

Les relations ainsi trouvées sont sauvegardées dans un fichier texte ou un fichier XML.

### c) La validation des entrées

Cette étape consiste àvérifier la validitédes concepts ainsi que les relations entre eux.

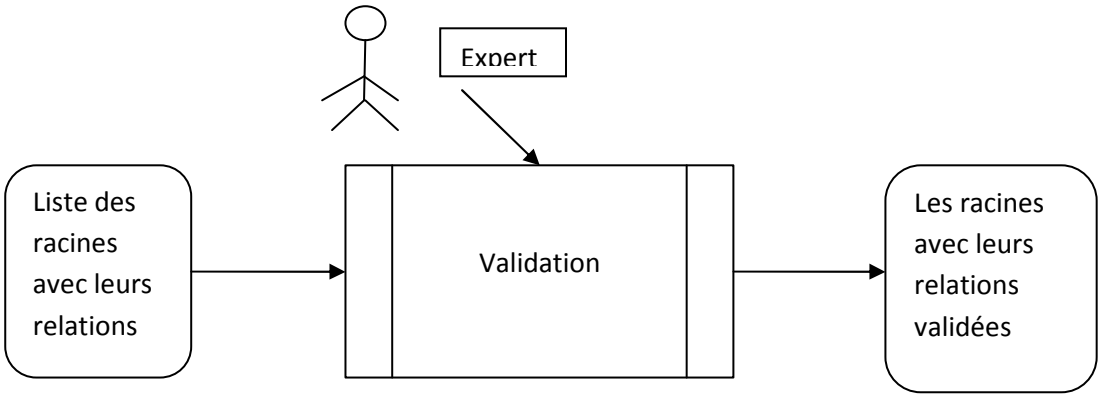



**Figure 3.5 La validation**

### d) Le Mapping

Cette opération consiste à transférer le contenu du fichier des racines avec leurs relations validées (Texte ou XML) vers notre Base de Données AraWordnet.

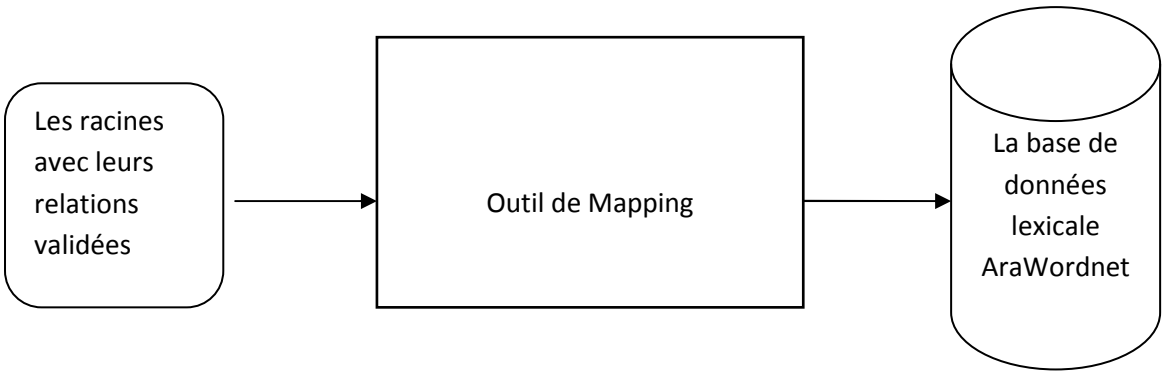



**Figure 3.6 Le Mapping**

L'outil de mapping est un programme qui prend en entrée le fichier généré par notre application (format texte ou xml), et génère la base données lexicale AraWordNet.

# 3.4 Interface de notre Application

Les figures 3.7, 3.8 et 3.9 présentent des imprimes écran de l'application réalisée.

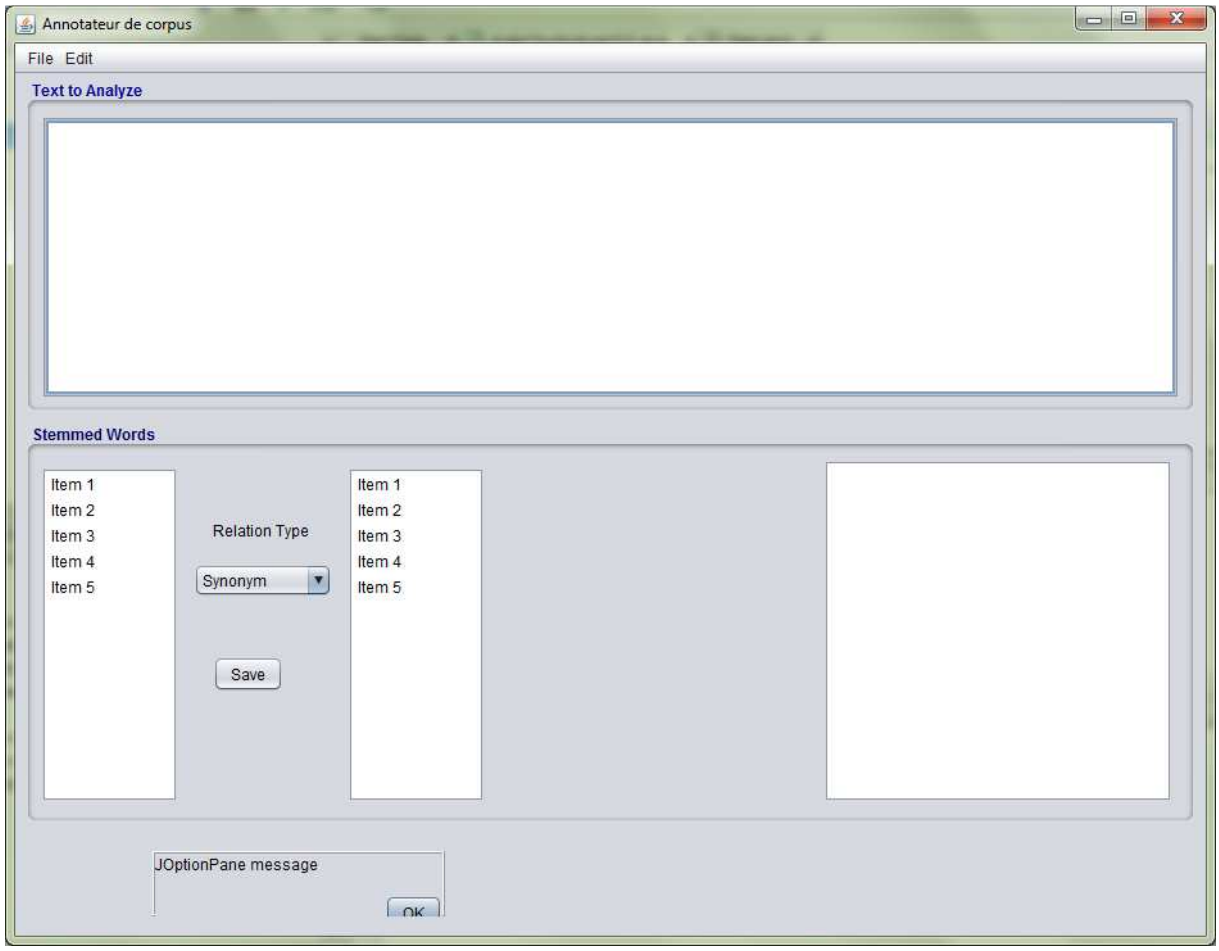



**Figure 3.7Imprime écran interface de notre application**

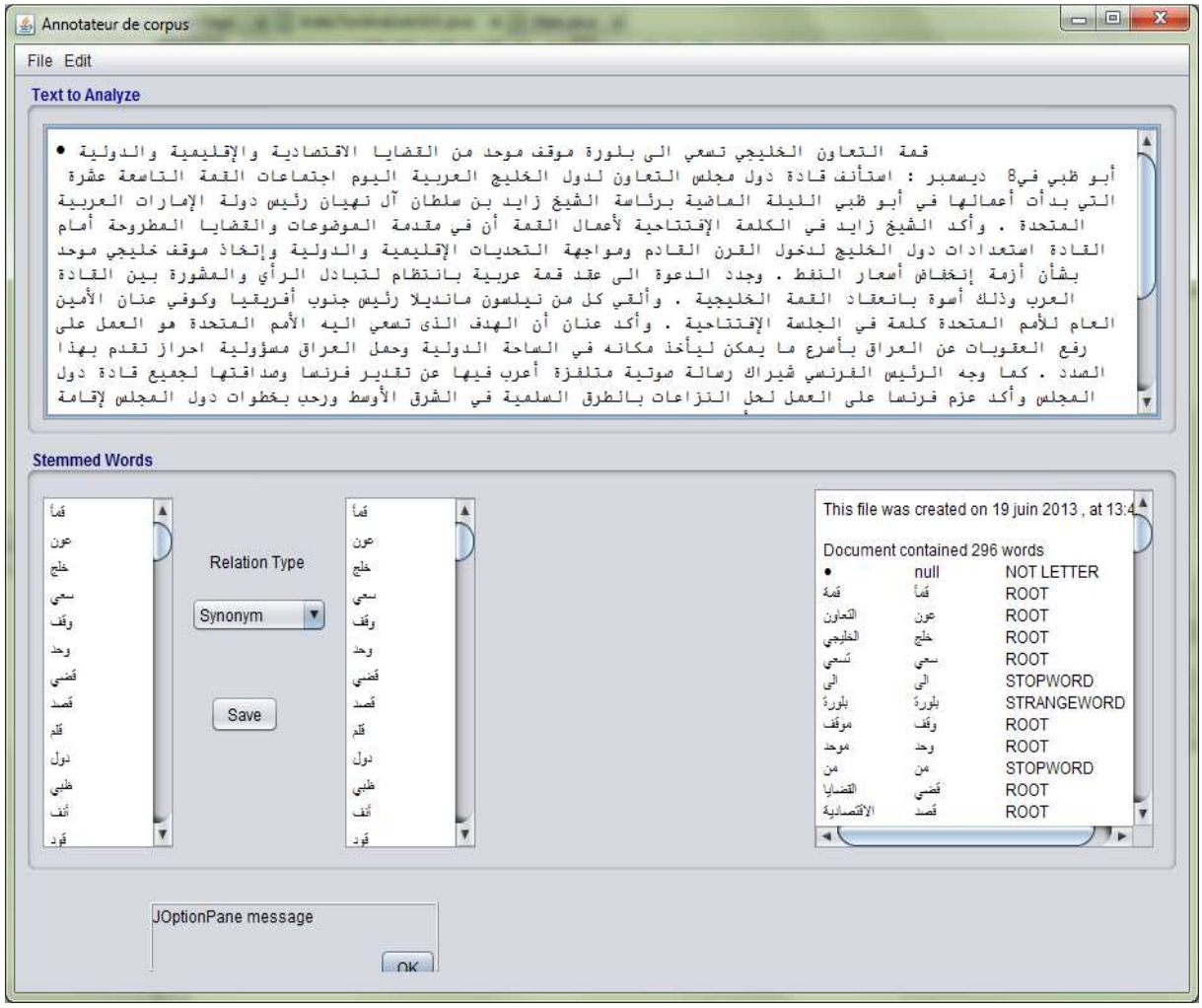



**Figure 3.8Imprime écran interface de notre application**

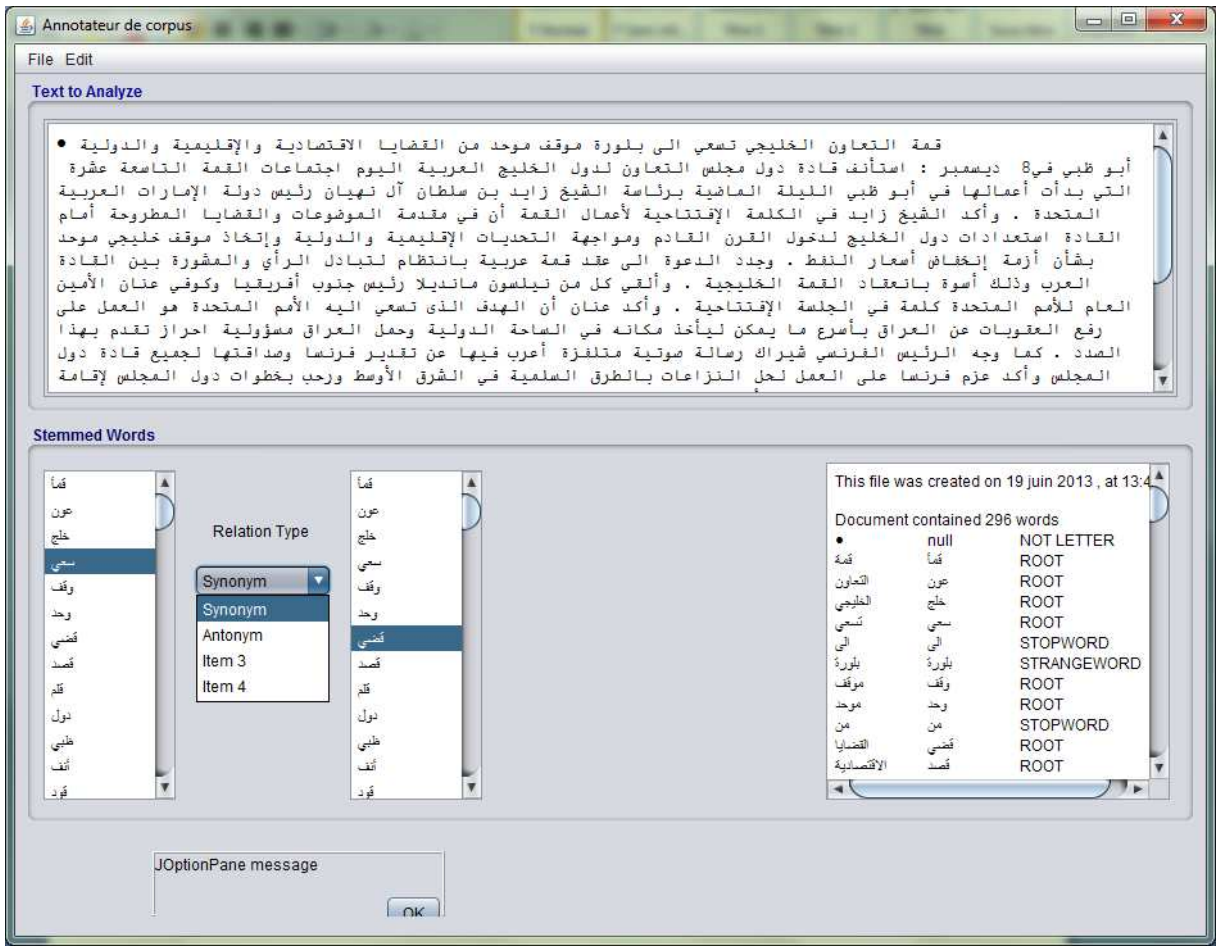



**Figure 3.9Imprime écran interface de notre application**

# 3.5 Conclusion

Cette application doit être déployée dans un environnement réseaux pour permettre la saisie en parallèle de plusieurs entrées de la base de données. Une étape de validation des entrées lexicales par un expert linguiste est nécessaire afin de corriger d'éventuelles erreurs. A la fin de cette phase un outil automatique se charge de faire le mapping des entrées validées obtenues vers la base de données déjà créée.