Nasredine Semmar, Meriama Laib & Faina Ramdani
CEA, LIST, Laboratoire d'Ingénierie de la Connaissance Multimédia Multilingue, France
Christian.fluhr
CEA, LIST, Service Robotique Cognitique et Interaction, France

Evaluation d'un analyseur linguistique pour l'arabe dans un moteur de recherche interlingue

Résumé

La multiplication des documents, notamment sur le Web, dans de nombreuses langues a conduit au développement de moteurs de recherche interlingue. La recherche interlingue consiste à formuler une requête dans une langue et à rechercher des documents pertinents dans d'autres langues. Cet article présente une évaluation de l'impact de l'analyse linguistique sur la recherche d'information en langue arabe. Le moteur de recherche utilisé est basé sur un modèle booléen et utilise une analyse linguistique profonde pour l'indexation et la recherche dans une base de données textuelles composée de termes simples ou composés et d'entités nommées. Les documents retrouvés pertinents sont renvoyés groupés par classes représentant la répartition des mots de la requête dans la base de données textuelles. Chaque classe contient une liste de documents classés par ordre de pertinence.
Mots-clés: Recherche d'information, indexation, analyse linguistique, bases de données textuelles, évaluation

1. Introduction

L'arabe est une langue qui possède un mécanisme de dérivation fondé sur une forme canonique de trois ou quatre lettres à laquelle viennent s'ajouter des proclitiques et des enclitiques. Des voyelles s'y ajoutent également pour former des dérivés et des formes flexionnelles. Ces caractéristiques impliquent un traitement particulier des requêtes et des documents à indexer pour améliorer la précision des moteurs de recherche conçus pour des langues latines (Grefenstette et al., 2005).

Nous présentons dans la section 2 les principaux composants du moteur de recherche interlingue développé au LIC2M, en particulier, nous nous focalisons sur les modules de l'analyse linguistique de l'arabe. Dans la section 3, nous décrivons la méthode utilisée pour l'évaluation de l'impact de l'analyse linguistique sur la recherche en donnant quelques exemples d'interrogation. Nous discutons dans la section 4 les résultats obtenus en interrogeant une base de 50 documents avec des requêtes longues et courtes en langue arabe. La section 5 conclut notre étude et présente nos travaux futurs.

2. Le moteur de recherche interlingue

Le moteur de recherche interlingue développé au LIC2M permet, à partir d'une requête en une seule langue, de fournir des réponses trouvées dans des documents qui sont dans d'autres langues. Ce moteur de recherche traite actuellement six

langues: français, anglais, arabe, chinois, allemand, italien et espagnol, il est composé des modules suivants (Besançon et al 2003) (Semmar et al., 2005):

- Une analyse linguistique pour identifier les termes représentant les concepts, mots simples ou composés ainsi que certains homographes. Elle permet de normaliser les termes (infinitif pour les verbes, singulier pour les noms, masculin singulier pour les adjectifs) et aussi identifier les synonymes.
- Une analyse statistique qui attribue des poids aux mots simples et mots composés sur l'ensemble des documents indexés. Ce poids est lié à l'hétérogénéité de répartition du concept dans la base de documents. Il sera maximum si le concept est complètement discriminant, c'est-à-dire s'il apparaît dans un seul document, et minimum s'il n'est pas discriminant et apparaît dans tous les documents.
- Un reformulateur pour reformuler les requêtes de l'utilisateur pendant la recherche. La reformulation consiste à inférer à partir des mots d'origine de la requête d'autres mots exprimant le même concept. La reformulation peut être monolingue (synonymie, hyponymie, etc.) ou bilingue (traduction mot à mot).
- Un comparateur pour calculer la proximité sémantique entre la requête et les documents indexés à partir des mots communs (mots de l'intersection requête/documents). Ce comparateur consiste, d'une part, à identifier les meilleurs intersections requête/documents, et d'autre part, à regrouper les intersections identiques et leur attribuer un poids. Le résultat est présenté sous forme d'une liste de classes d'intersections dans un ordre croissant de pertinence.

2.1. L'analyse linguistique

L'analyse linguistique est effectuée par un ensemble de modules dont le nombre et la nature varient selon la langue traitée. Certains modules sont génériques et communs à toutes les langues traitées par le système alors que d'autres modules, plus spécifiques, ne sont utilisés que dans des cas précis définis selon la langue concernée (Laïb et al., 2006). L'analyseur linguistique du LIC2M se compose des modules suivants:

1. Le découpeur (tokenizer) découpe les chaînes de caractères du texte en mots, en prenant en compte le contexte ainsi que les règles de découpage.
2. La consultation des dictionnaires des formes simples permet de récupérer des informations linguistiques concernant les mots à reconnaître.
3. La recherche d'alternatives orthographiques permet de récupérer les formes voyellées des mots non voyellés ou semi-voyellés en consultant le dictionnaire des formes simples (Debili, Zouari, 1985)- (Zouari, 1989).
4. Si après la consultation du dictionnaire des formes simples, les mots agglutinés ne sont pas reconnus, c'est le rôle du segmenteur de les décomposer. Lorsque leur forme de surface le permet, ces mots sont donc segmentés en proclitique-radical-enclitique ou en proclitique-radical ou en radical-enclitique ou en proclitique-enclitique (Buckwalter 2002) (Larkey et al., 2002). Le segmenteur permet par exemple de décomposer le mot agglutiné وهواهم en و + هوى + هم en utilisant une règle de réécriture (Darwish, 2002) pour transformer le mot هوا en هوى.

5. La recherche d'expressions idiomatiques permet de repérer les expressions idiomatiques et de les considérer comme des mots simples. Cette reconnaissance se fait à l'aide de règles de déclencheurs, qui sont généralement des lemmes. Cette étape permet par exemple de considérer le mot ذُو القِعْدَة comme un seul mot, tout comme les autres mois du calendrier de l'hégire.
6. Si après ces étapes, un mot reste inconnu, le système lui attribue une catégorie par défaut en s'appuyant sur des informations révélées par sa forme de surface. Par exemple, s'il s'agit d'un mot en caractères latin majuscule comme ONU, il sera étiqueté en nom propre. S'il s'agit d'un mot en caractères arabes, il lui sera attribué plusieurs étiquettes à la fois (nom commun, nom propre, adjectif, verbe, etc.).
7. Après l'analyse morphologique la majorité des mots restent ambigus notamment à cause du nombre élevé des voyellations possibles. Le désambiguïseur morphosyntaxique réduit le nombre des ambiguïtés en utilisant des matrices de désambiguïsation. Ce sont des matrices de bi-grams et tri-grams obtenues à partir d'un corpus étiqueté et désambiguïsé manuellement.
8. Après la désambiguïsation morphosyntaxique vient l'analyse syntaxique qui permet à l'aide de règles écrites à la main de reconnaître les relations de dépendance entre les mots dans un même syntagme. L'analyse syntaxique permet par exemple de considérer le mot إِدَارَة المَشَارِيع الأَوْلَوِيَّة comme un mot composé de trois mots.
9. La reconnaissance des entités nommées (Abuleil, Evans, 2004) permet à l'aide de listes ainsi que des règles de déclencheurs de reconnaître les noms propres spécifiques, comme les noms de personnes, de lieux, d'organisations, de dates, d'événements, etc. Ainsi, un énoncé comme الأَوَّل مِن شَهْرِ مَارس est reconnu comme une date et الشَّرْق الأَوْسَط est reconnu comme un nom de lieu.

L'analyse linguistique sert donc à identifier les termes représentant les concepts, mots simple ou composés ainsi que certains homographes. Elle permet de normaliser les termes et d'identifier les synonymes. Cette analyse utilise un ensemble de ressources linguistiques comme les dictionnaires (formes simples, proclitiques, enclitiques, synonymes) et des règles d'extraction (mots composés, entités nommées, d'expressions idiomatiques).

2.2. Analyse statistique

L'analyse statistique consiste à attribuer un poids aux mots simples et aux mots composés sur l'ensemble des documents indexés, selon le "degré d'information" qu'ils contiennent. Ce poids est lié à l'hétérogénéité de répartition du terme dans la base de documents. Il sera maximum si le terme est complètement discriminant, c'est-à-dire s'il apparaît dans un seul document, et minimum s'il n'est pas discriminant et apparaît dans tous les documents (Andreewsky et al., 1981). (Salton, Mcgill, 1983).

2.3. Reformulation de la requête

Dans certains cas, l'analyse linguistique et l'analyse statistique expliquées ci-dessus ne suffisent pas à établir un lien entre la requête et les documents pertinents. Dans ce cas, il est nécessaire d'ajouter un élément sémantique au processus sur la base de la requête originale afin d'inférer ce que recherche l'utilisateur. Il s'agit donc d'étendre la requête posée en utilisant d'autres formulations de l'idée qui y est exprimée pour que soient retrouvés les documents susceptibles d'être pertinents. (Debili, et al., 1988). Cette reformulation peut aussi bien être dans la même langue (synonymes, hyponymes, etc.) que dans des langues différentes, et pour ce faire, le système du LIC2M utilise des dictionnaires de reformulation monolingue et bilingue.

2.4. Calcul de la proximité sémantique

Le comparateur sert à calculer la proximité sémantique entre la requête et les documents indexés à partir des mots communs (mots de l'intersection requête/documents). Ce comparateur consiste, d'une part, à identifier les meilleures intersections requête/documents, et d'autre part, à regrouper les intersections identiques et leur attribuer un poids. Le résultat est présenté sous forme d'une liste de classes d'intersections dans un ordre croissant de poids. Les documents de la base sont indexés et stockés dans des fichiers inversés. On construit un index pour chacune des langues des documents constituant le corpus et on applique l'analyse linguistique pour les documents à indexer et pour les requêtes.

3. Résultats expérimentaux

Pour évaluer l'interrogation monolingue arabe du moteur de recherche, nous avons constitué cinquante requêtes contenant certains pièges:

- Par exemple, une requête comportant le mot non voyellé arabe مارس devrait nous renseigner sur la qualité des analyses morphosyntaxique et syntaxique effectuées. En effet, le mot مارس peut avoir deux sens correspondant à deux voyellations différentes: celle qui correspond au mois de mars et celle du verbe faire ou pratiquer. Les résultats de la recherche nous renseigneront sur la capacité de l'analyseur linguistique à effectuer une bonne désambiguïsation. Car dans ce cas précis, le contexte peut permettre de différencier entre les deux catégories grammaticales, et donc entre les deux sens du terme. Le mot مارس peut également avoir une variante régionale utilisée essentiellement dans les pays du Moyen-Orient, il s'agit du mot آذار. Il serait intéressant de savoir si les textes contenant cette variante ou les deux formes (آذار/ مارس) sont renvoyés par le moteur de recherche.
- Pour le cas des mots polysémiques, nous avons choisi une requête contenant le mot قطر. Selon la voyellation de ce mot, il peut être un nom قَطَر (le Qatar) ou un verbe قَطَّرَ (distiller).

Pour évaluer la désambiguïsation, nous avons utilisé les deux éléments <title> et <desc> lors de la constitution de la liste des requêtes (Figure 1).

<top>
<num>50</num>
<title>إدارة موارد المياه</title>
<desc> بحث وثائق حول تعزيز قدرات إدارة موارد المياه وإشراك القطاع الخاص في أنشطة إعادة استخدام الماء </desc>
<narr></narr>
</top>

Figure 1: Exemple de requête en arabe pour l'interrogation du moteur de recherche.

Pour évaluer le moteur de recherche, nous avons créé un fichier de jugements humains au format appliqué dans les campagnes TREC et nous avons utilisé l'application *trec_eval* pour analyser les résultats.
Nous avons soumis l'ensemble des requêtes constituées pour comparer le fichier des résultats renvoyés au fichier de jugements humains constitué. Les expérimentations ont été effectuées comme suit:

- Un run avec uniquement le contenu de la balise <title> pour 50 documents renvoyés.
- Un run avec uniquement le contenu de la balise <desc> pour 50 documents renvoyés.

L'application *trec_eval* nous renvoie deux types de fichiers en résultat:

- Un fichier contenant la liste des documents renvoyés par le moteur de recherche. En comparant cette liste au fichier de jugements humains, nous allons pouvoir comprendre quel module du moteur de recherche a mal analysé la requête posée.
- Un fichier contenant un certain nombre de métriques (rappel, précision, etc.).

Les résultats du run concernant l'élément <title> (requête courte) et le run concernant l'élément <desc> (requête longue) sont illustrés par les courbes rappel/précision de la figure 2.

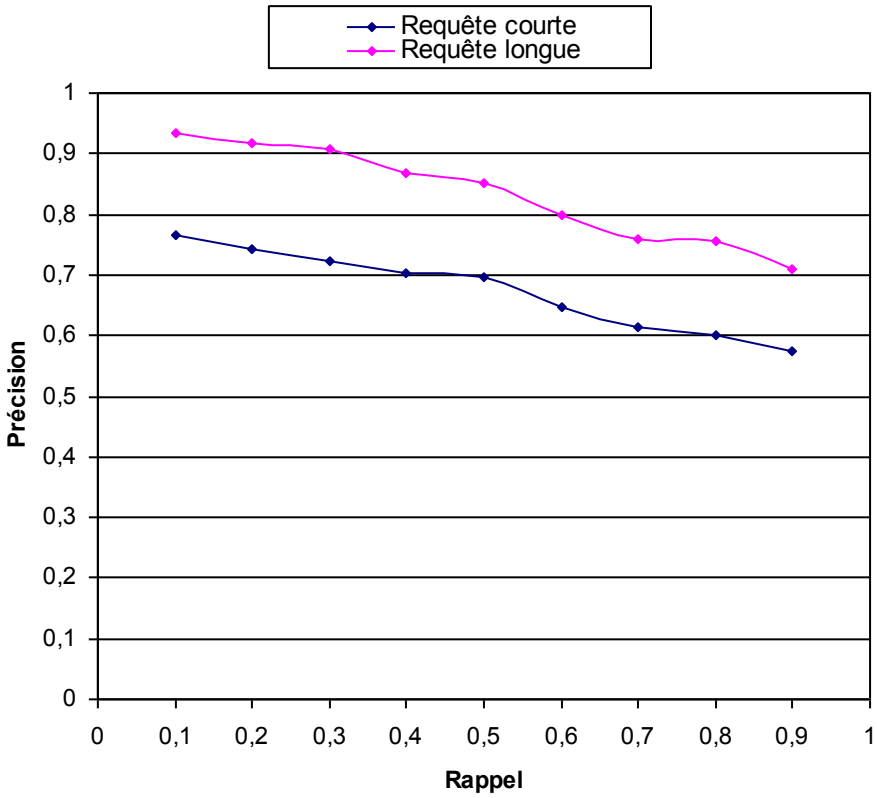

Figure 2: Courbes rappel/précision pour les 2 runs sur les éléments <title>et <desc>.

Nous constatons que la précision moyenne pour les requêtes courtes est de l'ordre de 65.98% et la F-mesure moyenne est de l'ordre de 51.55%. Pour les requêtes longues, la précision est de l'ordre de 83,30% et la F-mesure moyenne est de 56,30%.

4. Discussion

Les courbes rappel/précision des deux runs montrent que le moteur de recherche du LIC2M fournit de bons résultats. A titre indicatif, ces résultats sont nettement meilleurs que ceux des moteurs de recherche ayant participé à la compagne TREC-2002. Lors de cette campagne d'évaluation, la précision maximale obtenue pour la tâche monolingue était d'environ 75% pour les éléments <title> et<desc>. Les scores de F-meure sont relativement élevés et montrent un assez bon rapport harmonique entre précision et rappel.

Sachant que le corpus utilisé (50 documents) est de taille largement inférieure au corpus de TREC-2002 (800 000 documents), il est légitime de s'interroger sur l'impact de la taille du corpus sur les résultats obtenus.

Par ailleurs, nous constatons que pour un même rappel, la précision est meilleure lorsqu'on soumet une requête longue (requête sur l'élément <desc>). Ceci est dû au fait que l'analyse linguistique donne de meilleurs résultats lorsque les mots sont dans des contextes plus ou moins grands. La désambiguïsation morphosyntaxique est alors plus aisée.

Pour évaluer le bruit et le silence, nous avons comparé la liste de ces documents renvoyés par le moteur de recherche au fichier de jugements humains que nous avons constitué en début d'évaluation. Cette comparaison nous a permis d'identifier les raisons des erreurs constatées et ayant engendré soit du bruit soit du silence.

Par exemple, pour la requête courte comportant uniquement le mot قطر, le document pertinent Ara_Id_44 n'a pas été renvoyé car le désambiguïseur morphosyntaxique a choisi la catégorie du verbe à l'accusatif avec deux lemmes et deux sens différents: قَطَرَ (verbe à l'accusatif, goutter) et قطر (verbe à l'accusatif, distiller).

Lors de l'analyse et de l'indexation du document Ara_Id_44 (document censé être retourné par le moteur de recherche), ce même mot قطر est étiqueté avec une autre catégorie mais qui prend le même lemme: قطر (nom propre, Qatar).

Au moment de la comparaison, le système compare uniquement les lemmes retenus pour les mots de la requête et ceux des documents. Dans cet exemple, les lemmes sont différents, par conséquent le système considère que le document Ara_Id_44 n'est pas pertinent.

Par ailleurs, certains documents non pertinents peuvent être renvoyés à cause d'une correspondance entre les lemmes qui n'est pas correcte. Par exemple, le document Ara_Id_3 a été retourné comme pertinent pour la requête composé du mot مارس pourtant il ne l'est pas. Ceci vient du fait que le moteur de recherche ne compare que les lemmes, il ne prend pas en compte les catégories grammaticales.

D'autre part, nous avons constaté que les documents contenant le mot آذار qui est une variante du mot مارس utilisée au Moyen-Orient ne sont pas renvoyés par le système.

5. Conclusion et Perspectives

Malgré la taille du corpus largement en deçà des normes habituelles, cette évaluation nous a révélé certaines failles du moteur de recherche, comme l'absence d'un dictionnaire de reformulation monolingue en langue arabe. Cette absence diminue les chances de retrouver des documents qui seraient pertinents mais qui ne contiendraient pas les mêmes mots que la requête. Nous avons également constaté que le désambiguïseur morphosyntaxique utilisé par défaut dans le moteur de recherche ne renvoie pas certains documents pertinents pour certaines requêtes courtes car il se base sur la comparaison d'une seule catégorie morphosyntaxique trouvée selon un des lemmes retenus lors de l'analyse linguistique. Cependant, lorsqu'on remplace ce désambiguïseur par un autre qui prend en compte les catégories grammaticales des mots analysés et non leurs lemmes, un nombre plus élevé de documents pertinents est renvoyé mais la quantité de documents non pertinents renvoyés augmente également. Par ailleurs, nous avons aussi remarqué que les règles de réécriture ont un impact sur la qualité de la segmentation des mots agglutinés et par conséquent sur la performance du moteur de recherche (Aljlayl, Frieder, 2002) (Abdelali et al., 2004). Pour améliorer la qualité des résultats du moteur de recherche, nos travaux vont maintenant s'étendre, d'une part, à l'intégration d'un désambiguïseur de sens des mots dans l'analyse linguistique, et d'autre part, à la prise en considération des failles révélées dans cette évaluation.

Références

Abdelali, A., Cowie, J.; Soliman, H., S.: 2004, Arabic Information Retrieval Perspectives. Actes du *TALN-2004*.

Abuleil, S., Evens, M.: 2004, Named Entity Recognition and Classification for Text in Arabic. Actes du *IASSE-2004*, 89-94.

Aljlayl, O., Frieder, O.: 2002, On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach. Actes du *ACM Eleventh Conference on Information and Knowledge Management*.

Andreewsky, A., Binquet, J.P., Debili, F., Fluhr, C., Pouderoux, B.: 1981, Le traitement linguistique et statistique des textes et son application dans la documentation juridique. Actes du *Sixième Symposium sur l'Informatique Juridique en Europe*, Thessaloniki, Grèce.

Besançon, R., De Chalendar, G., Ferret, O., Fluhr, C., Mesnard, O., Naets, H.: 2003, The LIC2M's CLEF 2003 system. In Working Notes for the *CLEF 2003 Workshop*.

Buckwalter, T.: 2002, Buckwalter Arabic Morphological Analyzer Version 1.0. *Linguistic Data Consortium*.

Darwish, K.: 2002, Building a Shallow Arabic Morphological Analyzer in One Day. Actes du *ACL-2005*, 47-54.

Debili, F., ZOUARI, L.: 1985, Analyse morphologique de l'arabe écrit voyellé ou non fondée sur la construction automatique d'un dictionnaire arabe. Actes de *Cognitiva-1985*.

Debili, F., Fluhr, C., Radasoa, P.: 1988, About Reformulation In Full Text IRS. Information Processing And Management, England.

Grefenstette, G., Semmar, N., Elkateb-Gara, F.: 2005, Modifying a Natural Language Processing System for European Languages to Treat Arabic in Information Processing and Information Retrieval Applications. Actes de *ACL-2005,* 31-38.

Laib, M., Semmar, N., Fluhr, C.: 2006, Utilisation d'une approche linguistique pour l'indexation et l'interrogation en langage naturel de bases de données textuelles multilingues. Actes du *Premier Colloque International sur le Traitement Automatique de la Langue Arabe*.

Larkey, L.S., Ballesteros, L., Connell, M.E.: 2002, Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis. Actes du *25th annual international ACM SIGIR conference on Research and development in information retrieval,* 275–282.

Salton, G., Mcgill, M.: 1983, Introduction to Modern Information retrieval. *McGraw Hill*, New York.

Semmar, N., Elkateb-Gara, F., Laib, M., Fluhr, C.: 2005, A Cross-language information retrieval system based on linguistic and statistical approaches. Actes du *Deuxième Congrès International sur l'Ingénierie de l'Arabe et l'Ingénierie de la Langue*.