Classification automatique de définitions en sens

Fabien Jalabert (1 & 2), Mathieu Lafourcade (2)
fabien.jalabert@ema.fr , mathieu.lafourcade@lirmm.fr

(1) LGI2P - Ecole des Mînes d'Alès
Parc Scientifique Georges Besse
30 035 - Nïmes Cedex 1
www.lgi2p.ema.fr

(2) LIRMM - Université Montpellier II
34 392 - Montpellier Cedex 5
www.lirmm.fr

Mots-clefs – Keywords

Traitement automatique des langues naturelles, classification automatique, désambiguïsation sémantique lexicale
Natural language processing, unsupervised clustering, word sense disambiguation

Résumé - Abstract

Dans le cadre de la recherche en sémantique lexicale, l'équipe TAL du LIRMM développe actuellement un système d'analyse des aspects thématiques des textes et de désambiguisation lexicale basé sur les vecteurs conceptuels. Pour la construction des vecteurs, les définitions provenant de sources lexicales différentes (dictionnaires à usage humain, listes de synonymes, définitions de thésaurus, ...) sont analysées. Aucun découpage du sens n'est présent dans la représentation : un vecteur conceptuel est associé à chaque définition et un autre pour représenter le sens global du mot. Nous souhaitons effectuer une catégorisation afin que chaque élément ne soit plus une définition mais un sens. Cette amélioration concerne bien sur directement les applications courantes (désambiguïsation, transfert lexical, ...) mais a aussi pour objectif majeur d'améliorer l'apprentissage de la base.

In the framework of research in meaning representation in NLP, we focus our attention on thematic aspects and conceptual vectors. A vectorial base is built upon a morphosyntactic analysis of several lexical resources to reduce isolated problems. A conceptual vector is associated with each definition and another one with the global meaning of a word. There is no effective meaning division and representation the the knowledge base. We study in the article a clustering method that merge definitions into senses. This applies on common problems (word sense disambiguation, word translation, ...) and mainly to improve knowledge base learning.

Introduction

Dans le cadre de la recherche en sémantique lexicale, l'équipe TAL du LIRMM développe actuellement un système d'analyse des aspects thématiques des textes et de désambiguïsation lexicale basé sur les vecteurs conceptuels (Lafourcade et al., 2002). Les vecteurs représentent les idées associées à tout segment textuel (mots, expressions, textes, ...) via l'activation de concepts. Pour la construction des vecteurs, nous avons pris notamment l'hypothèse d'un *apprentissage multi-source* afin de pallier le bruit définitoire (par exemple les problèmes dus au métalangage comme dans la définition d'‹*aboyer*› : *crier* ***en parlant*** *du chien*).

L'utilisation de multiples sources fait disparaître la notion d'atomicité du sens. Un vecteur est associé globalement à un mot et plus finement à chaque définition. Mais aucun découpage des sens n'apparaît réellement à ce stade. Nous proposons dans cet article d'effectuer une classification non-supervisée (catégorisation) afin de regrouper les définitions similaires en sens. Les méthodes de catégorisation sont nombreuses et bénéficient de nombreux travaux mais ne sont pas directement adaptées pour traiter des définitions de dictionnaires.

L'étude suivante décrit la spécificité de ce problème ainsi que les choix proposés en réponse avant de présenter en détail la procédure mise en œuvre.

1 Catégorisation des définitions

Les méthodes de classification automatique sont nombreuses (Alpert, Kahng, 1995), (Berkhin, 2002) mais ne sont cependant pas directement applicables dans le cadre de cette étude, car la catégorisation et la classification dans ces domaines traitent un grand nombre de données à répartir dans un faible nombre de classes en un processus unique. Dans notre cas, la masse de données est importante (actuellement pour 110 000 termes, plus de 430 000 définitions et vecteurs conceptuels), mais la catégorisation porte sur les définitions d'un seul terme (environ 5 définitions en moyennes par source, certains termes fortement polysémiques peuvent en avoir plus de 50).

Cependant, dans notre cas, une catégorisation ne s'applique qu'à quelques dizaines de définitions tout au plus. Les algorithmes du domaine de la fouille de données recherchent une efficacité globale dans un grand ensemble de données. Notre approche du problème se distingue donc par une importance moindre du coût calculatoire et s'appuie sur une profondeur d'analyse supérieure.

1.1 Choix de l'algorithme

Le choix de l'algorithme repose de différents constats :

Le volume de la donnée est faible. Le problème est donc moins restrictif concernant le choix des méthodes d'analyses et de l'algorithme. Cependant, il est impossible d'envisager un entrainement sur lequel repose certains algorithmes dont les *Support Vector Machine (SVM)* par exemple (Vapnik, Chervonenkis, 1964), (Burges, 1998).

Les dictionnaires sont supposés fiables et par conséquent deux définitions d'un même dictionnaire ne peuvent appartenir à une même classe résultat.

L'aspect hiérarchique n'est pas prépondérant dans les sens. Si une relation partielle d'hyperonymie est présente en sémantique, il est fréquemment impossible de généraliser deux sens car leur découpage ne repose pas exclusivement sur une hiérarchie (mais sur une analogie par exemple).

Le nombre de classes est inconnu *a priori*. Cependant, l'hypothèse de fiabilité des dictionnaires décrite ci-dessus implique que le nombre de sens d'un terme donné est supérieur ou égal au

nombre maximum de définitions présentes dans une des sources. Nous avons donc opté pour l'algorithme des *k-moyennes*. Les centroïdes sont initialisés avec les définitions du dictionnaire qui en a le plus grand nombre. Puis, de façon itérative pour chaque dictionnaire, et de façon gloutonne, chaque définition est affectée aux différentes classes et les centroïdes sont recalculés.

Il faut noter que l'hypothèse précédente n'est pas vraie. Certains dictionnaires divisent deux sens là ou certains n'en proposent qu'un plus général. Lorsque cela se produit simultanément pour deux sens différents, deux dictionnaires possédant chaque sens peuvent en réalité signifier 5 sens feuilles et deux sens hyperonymiques. Ce problème s'avère cependant isolé et dans le cas de termes à très forte polysémie, ces termes sont peu nombreux mais en revanchent sont fréquents en usage. Nous recherchons actuellement des solutions face à cette difficulté.

2 Principe général de l'algorithme

2.1 Déroulement général

L'algorithme utilisé suit le principe des *k-moyennes*. À l'initialisation, soit D l'ensemble des sources lexicales d_i et d_{max} le dictionnaire comportant le plus grand nombre n_{max} de définitions. Alors on construit n_{max} classes et à chacune est affectée une définition de d_{max}. Soit C l'ensemble des classes c_i obtenues. L'algorithme a pour objectif de rechercher un partitionnement optimal pour la fonction d'évaluation globale ($Eval$) suivante : $Eval(C) = \frac{D_{Inter}(C)}{\sum_{i=1}^{n} D_{Intra}(c_i)}$

avec D_{Inter} la distance entre les catégories $c_i \in C$:

$$D_{Inter}(C) = \sqrt[p]{\frac{1}{n^2} \sum_{i=1,j=i+1}^{n,n} D_A(c_i, c_j)^p} \qquad \text{avec} \qquad c_i, c_j \in C$$

et avec D_{Intra} la distance interne à une catégorie $c \in C$:

$$D_{Intra}(c) = \sqrt[p]{\frac{1}{n} \sum_{i=1}^{n} D_A(d_i, m)^p} \qquad \text{avec} \quad d_i \in c$$

et avec $m = \frac{\sum_{i=1}^{|C|} c_i}{|C|}$ centroïde de la catégorie

L'algorithme itère pour chaque source et procède à l'affectation de valeur minimum entre les définitions de la source et les différentes classes. Etant donné le faible nombre de sources lexicales, il est nécessaire que toutes les sources soit affectées sur un ensemble de centroïdes déjà amorcé. L'algorithme effectue ainsi au minimum $k = 2 \times |\{sources\}|$ itérations. Quand une source a déjà été affectée, ses éléments sont supprimés des classes avant d'être à nouveau réaffectés. L'algorithme se termine lorsqu'il y a convergence. En pratique, il est rare que la convergence ne soit pas obtenue après k itérations.

2.2 Problème de l'affectation de coût minimal

A chaque étape de l'itération, l'algorithme possède une matrice de distances entre les catégories et les définitions qui doivent être affectées. Il se pose donc le problème de trouver l'affectation de coût minimal : ce problème est équivalent au problème couplage maximum de valeur minimum dans un graphe biparti, ou encore à un problème de combinaison linéaire. La méthode Hongroise (Kuhn, 1955) (algorithme de complexité $O(n^3)$, où n est la cardinalité de la plus grande des deux partitions du graphe) est un cas particulier de (Ford, Fulkerson, 1956) qui considère de problème dans le cas plus général d'un graphe.

3 Profondeur d'analyse et aspect multicritère

Le dernier aspect de cet algorithme et la méthode d'analyse qui permet d'obtenir les distances. Nous l'avons vu, les impératifs de complexité sont moins restrictifs que ceux généralement présent dans la fouille de donnée. Ainsi nous avons fait le choix d'une approche multicritère afin de pallier statistiquement les défauts isolés inhérents à chaque type d'analyse du sens.

3.1 Types de critères

Chaque critère est sollicité à deux étapes différentes de l'algorithme pour fournir une mesure de distance. A chaque itération de l'algorithme, ils doivent d'une part proposer une matrice de distance pour l'affectation des définitions d'une sources dans les différentes catégories, puis d'autre part à la fin de l'itération permettre d'évaluer les distances inter et intra-catégories. Les critères sont pondérés entre eux et suivant le type de source lexicale utilisé. Ces critères s'articulent autour de trois axes principaux :

Le critère de distance angulaire est simplement basé sur l'utilisation de l'angle entre deux vecteurs. Ce critère est particulièrement important dans le cas des ‹*définitions manuelles*›, qui sont très succintes, et qui sont généralement insérées dans la base pour corriger et fixer des sens.

Le critère d'analyse *ad-hoc* extrait le contenu lexical des définitions en recherchant des locutions *ad-hoc*. Ils recherchent par exemple des informations de domaine (‹*mécanique*›, ‹*biologie*› ...), d'étymologie (‹*du latin ... qui signifie ...* ›), d'usage (‹*ancien*›, ‹*argot*›, ...). Il ne s'agit pas de proposer une mesure de distance mais de proposer un bonus ou un malus pondéré en fonction de chaque cas. Ce critère recherche aussi d'autres motifs dans le cas de dictionnaire semi-structurés (à l'aide de XML ou SGML par exemple).

Le critère d'analyse de contenu lexical extrait les similitudes entre les contenus lexicaux des différentes définitions. Il peut être paramétré avec des fonctions de fréquence et de cooccurence de termes qui permettent après un apprentissage en corpus de refléter l'usage des termes. Ce critère est plus amplement détaillé dans le paragraphe suivant.

3.2 Critère d'analyse de contenu lexical

Ce critère compare les définitions par la présence simultanée de termes. La cooccurence obtenue est pondérée par la position et les informations morphosyntaxiques de chaque définition. En effet, la plupart des définitions sont décrites en *genre et différence*, c'est à dire par un hyperonyme suivi de la description souvent ordonnée des caractéristiques propres. Cependant, l'apposition d'un complément (placé en début de définition, avant le verbe et le sujet) est fréquente dans les définitions et constitue souvent une forte participation au sens.

Le déroulement de cette évaluation est le suivant :

– A l'aide d'une analyse morphosyntaxique (Chauché, 1984), nous supprimons les termes appartenant au méta-langage, les déterminants, pronoms, prépositions et plus généralement les termes qui ne participent pas fortement à la thématique de la définition. Puis en fonction de l'arbre morphosyntaxique, nous réordonnons tous les termes restants en traitant par exemple les appositions de compléments, mais aussi en plaçant les gouverneurs avant les adjoints[1]) ...

[1]Par exemple dans ‹*voile à bateau*› on donne plus d'importance à ‹*voile*› qui est le le gouverneur (ou tête) qu'à ‹*bateau*› qui est son adjoint, tandis que dans ‹*bateau à voile*›, ‹*bateau*› est favorisé

Enfin, une fois cette liste de termes dont l'ordre est sensé refléter la participation décroissante à la thématique, nous indiçons[2] tous ces termes.
– Ensuite nous appliquons une fonction décroissante de 1 à 0 en fonction de la valeur de cette indice ind : $f(ind) = k_1 - k_2.log(ind)$ où k_1 et k_2 sont des constantes réelles.
– Pour les termes qui ont plusieurs occurences dans une définition (dans le cas où on compare deux définitions entre elles) ou dans un ensemble de définitions (dans le cas où on compare une définition avec une classe de définitions), on remplace les occurences multiples par une seule dont la valeur est la somme des précédentes. Soient $ind_1..ind_j$ les indices occurences du mots m dans une définition, alors ces occurences sont remplacées par $\sum_{k=1}^{j} f(ind_k)$
– Enfin, on mesure une proximité entre documents sources par une fonction qui effectue la somme des produits des indices pondérés pour chaque terme présent dans les deux définitions ou groupes de définitions comparés : $prox_{d_1,d_2} = \sum_{t\in(d_1\cap d_2)}\Big(f(\ ind_{t\in d_1}\ ).\ f(\ ind_{t\in d_2}\ )\Big)$

Remarques :
– Notons que cette fonction n'est pas une mesure de similarité ni de distance. Elle ne respecte notamment pas la propriété de minimalité : $prox(x, x) \neq 0$
– d_i est une définition et c_j est une catégorie, on les considère tous deux comme des ensembles de termes indicés, les indices commençant à 1 à chaque début de définition.
– On peut proposer plusieurs variantes en fonction des choix suivants :
  – La position d'un terme peut être *brute* c'est à dire directement dans la définition, ou peut être calculée dans l'arbre morphosyntaxique.
  – Il est possible de comptabiliser ou non plusieurs fois un terme qui possède de multiples occurences dans une même définition ou classe de définition.
  – Enfin, pour comptabiliser ces occurence multiples, on peut considérer l'indice minimum ou encore effectuer la somme des valeurs obtenues en appliquant f aux indices.

3.3 Pondération distributionnelle

Enfin, les résultats précédents peuvent être pondérés en ajoutant des critères de fréquence ou de cooccurence dans les corpus :

– Plus un terme est rare dans la totalité des dictionnaires ou du corpus, plus il est discréminant. On peut donc choisir de pondérer la fonction de proximité de la façon suivante : $prox_{freq}(d_1, d_2) = \sum_{t\in(d_1\cap d_2)}\left(\frac{f(\ i_{t\in d_1}\ ).\ f(\ i_{t\in d_2}\ )}{log(freq(t\in d_1)).\ log(freq(t\in d_2))}\right)$
– La cooccurence permet de pondérer le résultat obtenu par les définitions en tenant compte de l'usage de la langue dans les corpus. Plusieurs méthode d'application sont possibles :
  – La première consiste à pondérer l'importance des termes en valorisant ceux présents qui sont corrélés avec le mot que l'on souhaite définir. Ainsi, soit m un terme, d_1 et d_2 deux définitions de ce terme que l'on souhaite comparer et t_k les termes présents dans ces définitions : $prox_{coocc_1}(d_1, d_2) = \sum_{t\in(d_1\cap d_2)}\Big(f(i_{t\in d_1})\ .\ f(i_{t\in d_2})\ .\ coocc(t, m)\Big)$
  – On peut d'autre part étendre la comparaison en comparant tous les termes des définitions et en pondérant par la cooccurence. La similarité ne dépend plus alors de la présence simultanée d'un même terme dans les deux définitions mais de la présence dans les définitions de termes sémantiquement proches :
  $prox_{coocc_2}(d_1, d_2) = \sum_{t_1\in d_1, t_2\in d_2)}\Big(f(i_{t_1})\ .\ f(i_{t_2})\ .\ coocc(t_1, m)\ .\ coocc(t_2, m)\Big)$
  – Enfin il est possible de conjuguer les deux critères par une pondération supplémentaire de la seconde formule avec m, par exemple :
  $prox_{coocc_3}(d_1, d_2) = \sum_{t_1\in d_1, t_2\in d_2)}\Big(f(i_{t_1})\ .\ f(i_{t_2})\ .\ coocc(t_1, m)\ .\ coocc(t_2, m)\ .\ coocc(t_1, t_2)\Big)$

[2]Par ordre croissant en commençant à 1.

Ces critères présentes des résultats encourageants, mais une étude comparative serait nécessaire afin de déterminer les paramètres adéquats et les méthodes les plus performantes ou complémentaires.

Résultats et conclusion

Nous avons présenté dans cet article une nouvelle méthode permettant de catégoriser les définitions provenant de multiples sources lexicales afin d'obtenir des sens. Les résultats obtenus sont assez encourageants, et plus particulièrement, la méthode d'analyse pondérée par la cooccurence de termes. Cependant de nombreux travaux restent encore nécessaires : la classification est efficace pour une bonne proportion du lexique, mais s'avèrent moins pertinente pour des termes possédant un grand nombre de sens, qui sont moins nombreux mais très fréquents (les sens sont trop nombreux et insuffisamment démarqués). L'observation des erreurs commises montre que les interversions relèvent souvent de cliques de sens et notre réponse à ce problème s'oriente donc actuellement à réduire le nombre de sens et fusionner les multiples classes.

Mais ceci pose de nouveaux problèmes : avant tout, réduire le nombre de sens implique que plusieurs définitions d'une même source peuvent être affectées à une même classe, dont les conséquences peuvent être un perte d'efficacité globale. Enfin, les méthodes naïves statistiques que nous avons mises en jeu pour détecter et choisir le nombre de sens n'offrent pas de résultats satisfaisants. Cependant, malgré toutes ces difficultés, l'obtention de sens si imparfaite qu'elle est actuellement s'avère bénéfique pour l'apprentissage de la base vectorielle. L'amélioration de cette dernière laisse prévoir réciproquement un impact positif sur la catégorisation.

Références

C.J. Alpert, A.B. Kahng *Recent Directions in Netlist Partitioning : A Survey* Integration : VLSI J., vol. 19, 1995, 93 pp.

P. Berkhin *Survey of clustering data mining techniques* Accrue Software Research Paper, 56 pp.

C.J.C. Burges *A Tutorial on Support Vector Machines for Pattern Recognition* Journal : Data Mining and Knowledge Discovery, vol. 2, number 2, pp. 121-167, 1998.

J. Chauché *Un outil multidimensionnel de l'analyse du discours* Coling'84, Stanford, July 1984

L.R. Ford, D.R. Fulkerson *Maximal Flow through a Network* Candidan Journal of Mathematics,p. 399, 1956.

H.W. Kuhn *The hungarian method for the assignment problem* Naval Res. Logist. Quart., pages 83–98, 1955.

M. Lafourcade, V. Prince, D. Schwab *Vecteurs conceptuels et structuration émergente de terminologies* Revue TAL Volume 43 - n 1/2002, pages 43 à 72

V. Vapnik, A. Chervonenkis *A note on one class of perceptrons* Journal Automatic and Remote Control, vol. 25, 1964.