Exploration des paramètres discriminants pour les représentations vectorielles de la sémantique des mots

Frank Meyer, Vincent Dubois

France Telecom R&D
Avenue Pierre Marzin
22307 Lannion cédex
franck.meyer@francetelecom.com
vincen.dubois@francetelecom.com

Résumé : Les méthodes de représentation sémantique des mots à partir d'une analyse statistique sont basées sur des comptes de co-occurrences entre mots et unités textuelles. Ces méthodes ont des paramétrages complexes, notamment le type d'unité textuelle utilisée comme contexte. Ces paramètres déterminent fortement la qualité des résultats obtenus. Dans cet article, nous nous intéressons au paramétrage de la technique dite Hyperspace Analogue to Language (HAL). Nous proposons une nouvelle méthode pour en explorer ses paramètres discriminants. Cette méthode est basée sur l'analyse d'un graphe de voisinage d'une liste de mots de référence pré-classés. Nous expérimentons cette méthode et en donnons les premiers résultats qui renforcent et complètent des résultats issus de travaux précédents.

1 Introduction

Le but des méthodes de représentation sémantique des mots basées sur des vecteurs est d'associer à chaque mot d'un corpus un vecteur (en général dans R^N) de telle manière que la distance sémantique entre 2 mots soit reflétée par la distance entre les 2 vecteurs les représentants. On cherche donc à modéliser le sens des mots sous une forme numérique, objective, dans un espace vectoriel. Les méthodes de représentation sémantique par vecteurs les plus connues sont Latent Semantic Analysis (LSA) (Deerwester, Dumais, Furnas, Landauer et Harshman 1990) et Hyperspace Analogue to Language (HAL) (Lund et Burgess, 1996). Nous allons d'abord présenter brièvement le principe de ces deux méthodes et expliquer pourquoi, dans le cas des mots, la méthode de type HAL nous apparaît comme plus appropriée. Après avoir rapidement présenté les principaux travaux dans le domaine du choix des paramètres de HAL, nous exposerons notre méthode. Une expérimentation destinée à illustrer ses principes est ensuite décrite, ainsi que ses principaux résultats.

LSA et HAL sont des méthodes basées sur une analyse statistique d'un corpus de documents textuels. Les documents peuvent être des courts textes, des paragraphes voire des phrases. En sortie d'analyse, LSA et HAL produisent une matrice finale qui représente chaque mot i par son vecteur v_i. Pour mesurer la proximité sémantique de 2 mots, on utilise une fonction de distance entre les 2 vecteurs u et v qui les représentent. La distance utilisée est souvent la distance du cosinus. D'autres distances sont couramment utilisées : city-

block,Kullback-Leibler..., souvent couplées avec des techniques de standardisation comme TF-IDF (Levy *et al.*, 1998).

LSA est avant tout utilisée pour la représentation sémantique des documents, plutôt que des mots, bien que les deux types d'objets soient analysables avec cette méthode. Dans LSA, la représentation sémantique est obtenue par une matrice des comptes mots × documents. Chaque cellule (i, j) de la matrice correspond à la fréquence du mot i dans le document j. On suppose qu'initialement le nombre de documents est suffisant pour obtenir des statistiques fiables : classiquement, LSA utilise un corpus de plusieurs centaines ou plusieurs milliers de documents. Sur cette matrice des comptes de mots on effectue ensuite une Décomposition en Valeurs Singulières (Eckart and Young, 1936 ; Golub and Reinsch, 1970 ; Golub and van Loan, 1983). Cette opération vise d'une part à réduire la dimension de la matrice et d'autre part à effectuer une transformation de l'espace de représentation pour extraire les structures sémantiques sous-jacentes. LSA est une méthode très utilisée et qui a reçu de nombreuses améliorations depuis sa création, comme par exemple Probabilistic LSA (PLSA) (Hofmann, 1999).

Dans HAL, la représentation sémantique vectorielle est obtenue par une matrice de comptes de co-occurrences mots × mots. Dans HAL on utilise une fenêtre contextuelle à gauche et à droite du mot courant analysé lors du traitement du corpus. Chaque cellule (i, j) compte le nombre de fois où le mot j apparaît après le mot i dans la fenêtre contextuelle de S mots. S est un paramètre de la méthode. Dans la version initiale de HAL (Lund et Burgess, 1996), les comptes de co-occurrences sont pondérés par un paramètre décroissant selon le nombre de mots qui séparent 2 mots analysés. Sur la matrice des comptes de co-occurrence, comme pour LSA, une technique de réduction de dimension est utilisée, basée sur la suppression des colonnes de faible variance. En général, la dimension finale de la matrice est du même ordre de grandeur que pour LSA.

Les techniques dérivées de HAL considèrent des événements de type "le mot y est apparu après" et "le mot y est apparu avant " dans une fenêtre contextuelle autour d'un mot x qui est à représenter sous une forme vectorielle.

Ces informations de position (relative ou exacte) des mots ou des séquences par rapport à un mot donné sont plus riches que celles de LSA qui utilise un modèle "sac de mots" où l'ordre des mots n'est pas pris en compte. Pour la représentation vectorielle des mots HAL a un autre avantage sur LSA : la construction de la matrice de représentation sémantique est beaucoup plus rapide : elle s'effectue en temps linéaire alors que LSA est basée sur une décomposition en valeurs singulières, au mieux quadratique. Pour la représentation vectorielle de la sémantique des mots, les techniques de type HAL apparaissent donc comme plus adaptées.

Cet article s'intéresse aux techniques dérivées de HAL, et plus particulièrement à leurs paramètres. En effet, à sa création HAL est une méthode très empirique, le choix des différents paramètres ayant été peu étudié.

Nos principales questions dans cet article sont les suivantes :

- quelles sont les tailles de fenêtre de contexte les plus adaptées ?
- quelle est l'unité textuelle la plus adaptée pour les comptes de co-occurrences avec les mots à modéliser ? La plupart des études sont effectuées avec des co-occurrences mots x mots, des N-grams de mots par exemple sont-ils également envisageables ?
- quelle peut être le degré d'influence de la langue sur ces paramètres ?
- quelle peut être le degré d'influence de la fonction de distance sur les résultats ?

2 Travaux précédents

Church et Hanks (1990) proposèrent une première mesure de la notion de lien sémantique (word association) basée sur la quantité d'information mutuelle. Ils étudièrent les cooccurrences des mots sur des corpus de plusieurs millions de mots à l'intérieur d'une fenêtre contextuelle de 5 mots, en prenant en compte l'ordre des mots. La mesure du lien entre 2 mots ne dépendait que des fréquences de ces mots, de la fréquence de leurs cooccurrences et leurs positions relatives (sans utilisation d'une représentation vectorielle).

Flint et Chater, (1992) furent parmi les premiers à étudier les méthodes de représentations vectorielles à base de co-occurrences de mots, à partir d'un corpus de Usenet. Ils utilisèrent comme unité contextuelle uniquement les 150 mots les plus communs de leur corpus, et une fenêtre contextuelle de 2 mots avant et après le mot courant analysé. La fonction de distance utilisée était le coefficient de corrélation de Spearman. Il n'y eut pas de comparatifs de performances sur cette première étude.

Lund et Burgess (1996) ont développé HAL avec des paramètres initiaux établis de manière empirique (fenêtre contextuelle de taille 10, décroissance linéaire des poids des co-occurrences). Ils notèrent que les résultats de HAL étaient inchangés en utilisant uniquement les 200 mots de contexte les plus variants.

Levy et Bullinaria (1998, 2001) ont établi les seuls tests réellement approfondis dans le domaine des représentations vectorielles de type HAL. Ils ont montré :

- que les fenêtres contextuelles de petite taille étaient préférables,
- que les mots classiquement considérés comme creux et généralement mis dans une liste d'exclusion apportaient en fait beaucoup d'information,
- que la réduction de dimension, si elle pouvait dans certains cas avoir un intérêt en termes de temps de traitement, n'améliorait par contre pas les représentations sémantiques, et que la décomposition en valeur singulière, coûteuse en temps de calcul, n'apportait aucun avantage dans HAL,
- que les fonctions de distances, bien que souvent discutées, ne montraient pas de différences très significatives dans les performances finales.

A notre connaissance, les études des unités textuelles, en dehors des travaux de (Schütze, 1993) et (Sahlgren et Swanberg, 2001) n'ont pas été effectuées sur des méthodes de type HAL. Enfin, aucune étude n'a été faite sur plusieurs langues simultanément.

3 Notre approche

Nous avons voulu construire une méthode de comparaison des paramètres de HAL valable pour plusieurs langues, et facile à implanter. Nous avons défini une méthode de construction d'un critère permettant d'évaluer les performances d'un algorithme. Nous avons ensuite appliqué notre méthode de construction de critère pour un corpus disponibles en 3 langues occidentales : anglais, français, espagnol. Nous avons ensuite sélectionné les paramètres que nous souhaitions étudier. Dans cet article notre but n'est pas de faire des tests exhaustifs mais d'exposer le principe de la méthode.

3.1 Principe de construction d'un critère de qualité sémantique

L'évaluation de la qualité de la représentation sémantique des mots d'un modèle de type HAL est souvent basé sur l'utilisation d'un test de synonymie standardisé comme le TOEFL. Le TOEFL propose des QCM où on doit choisir un synonyme d'un mot donné parmi 4. Le test du TOEFL est par exemple utilisé par (Landauer et Dumais , 1997) et (Levy et Bullinaria, 1998). Le test du TOEFL est limité à la langue anglaise. Nous avons choisi une méthode plus générale et facilement reproductible. Nous partons de la remarque suivante : dans la majorité des cas un mot ne peut avoir un sens proche d'un autre que s'il appartient à la même catégorie grammaticale (partir/aller ; fils/enfant ; Jésus/ Christ...). Cette condition n'est bien sûr pas suffisante, mais nécessaire. Nous pouvons utiliser facilement cette contrainte de classe pour évaluer des représentations vectorielles.

Le critère est alors construit pour un corpus A, pour une instance V_r d'un modèle vectoriel V de paramètres r, et pour une langue L_s parmi un ensemble de langues L. La méthode de construction est la suivante :

1. Choisir q classes grammaticales génériques pour les mots. Les classes doivent être faciles à déterminer dans A quelle que soit L_s dans L. Par exemple on pourra prendre 3 classes de mots : noms communs (substantifs), formes verbales (avec ou sans accord) et noms propres (personnes ou lieux).
2. Trier les mots du corpus A pour la langue L_s par ordre de fréquence décroissante.
3. Sélectionner les n premiers mots les plus fréquents dans chaque classe. Soit M l'ensemble des q.n mots
4. Pour chaque mot i de M :
   Sélectionner les k plus proches voisins de i selon la représentation vectorielle V_r
5. Sur le graphe de voisinage obtenu, calculer le critère h de séparation de classe de V_r, donné par :

$$h = \frac{\sum_{i=1}^{n} \sum_{j=1}^{k} \delta\left(C(i), C\left(NN_j(i)\right)\right)}{n.k} \qquad (1)$$

avec δ : symbole de Kronecker , δ(x,y)=1 si x=y et δ(x,y)=0 si x≠y
avec C(x) : classe du mot x
avec $NN_j(x)$: j ième plus proche voisin du mot x

Le critère h correspond au pourcentage d'arcs du graphe de voisinage qui relient 2 mots de même classe.

Le critère h vaut 1 pour des classes de mots parfaitement séparées : chaque mot a pour plus proches voisins des mots de la même classe. Le critère h vaut 0 pour des classes de mots totalement mélangées : chaque mot a pour plus proches voisins des mots d'une classe différente. Le critère h nous permet de comparer deux instances de modèles vectoriels de V sur le même corpus A et la même langue L_s.

Soit V_P et V_Q deux modèles de représentation vectorielle des mots d'un corpus A dans une langue L_s (c'est à dire que V_P a été généré avec un jeu de paramètres P et que V_Q a été généré avec un autre jeu de paramètre Q).

Soit h_P le critère obtenu par un modèle V_P et soir h_Q le critère obtenu par un modèle V_Q. Nous faisons l'hypothèse que si $h_P < h_Q$ alors le modèle V_P est probablement sémantiquement meilleur que le modèle V_Q.

3.2 Modèle et paramètres explorés

Nous avons utilisé un modèle général de représentation sémantique proche de HAL. Les mots sont simplement des séquences de lettres séparés par des délimiteurs. L'algorithme construit une matrice représentant M mots dans un espace à N dimension. Chaque cellule (i,j) de la matrice correspond au compte des cooccurrences entre un mot i et une unité textuelle j dans une fenêtre contextuelle. Les mots à représenter sont donc les données, et les attributs sont donc les différentes unités textuelles. Une instance de ce modèle correspond à une exécution de cet algorithme avec des paramètres fixés. Les paramètres explorés sont les suivants :

Window Size : Taille de la fenêtre contextuelle utilisée pour les comptes de cooccurrence, en nombre de mots avant et après le mot courant analysé. La taille totale de la fenêtre d'analyse du corpus , incluant le mot courant analysé est donc de WindowSize × 2 + 1. La fenêtre contextuelle parcourt l'ensemble du corpus mot à mot, du premier mot au dernier. Pour une fenêtre d'analyse de taille 2, un corpus trivial de 6 mots "A B C D E F" serait parcouru en 6 étapes : [nul nul A B C] [nul A B C D] [A B C D E] [B C D E F] [C D E F nul] et [D E F nul nul]. Nous avons exploré les valeurs de 1 à 5 et 10 pour l'unité textuelle de type HAL (voir paramètre suivant) et les valeurs 1 à 3 uniquement pour l'unité textuelle de types Exact Position Context, OnlyOne-N-Gram et Multi-N-Gram. Ces types d'unité textuels sont en effet très consommateurs de mémoire et sont donc peu compatibles avec de grandes tailles de fenêtre contextuelle.

Text Unit Feature : Attributs d'unités textuelles utilisés. Ce paramètre a 4 valeurs possibles :

HAL : les unités textuelles sont ceux de la version d'origine de HAL. Chaque mot cible est représenté par ses comptes de cooccurrence (dans la fenêtre contextuelle) avec les autres mots. Les comptes sont pondérés par un paramètre proportionnel à la proximité des mots-attributs avec le mot cible. S'il y a M mots à représenter, et N mots différents dans le corpus, la matrice aura donc M lignes et N colonnes. Voir (Lund et Burgess, 1996) pour les détails.

Exact-Position-Context : Contexte avec Position Exacte : chaque mot cible est représenté par ses comptes de co-occurrence avec les autres mots, en distinguant chaque position possible dans la fenêtre contextuelle. La position relative de chaque mot en cooccurrence est concaténée avec le mot pour former l'attribut d'unité textuelle. Par exemple, pour une fenêtre de taille 2 et le mot cible X, la séquence de mots [A B X C A] produira les co-occurrences suivantes : (X, "A-2"), (X, "B-1"), (X, "C+1"), (X, "A+2"). S'il y a M mots à représenter, et N mots différents dans le corpus, la matrice aura donc M lignes et N × Window Size × 2 colonnes.

OnlyOne-N-Gram : chaque mot cible est représenté par ses comptes de cooccurrences avec le N-Gram de mots avant et après lui. Le N-Gram de mot a la même taille que la fenêtre contextuelle. Par exemple, avec une fenêtre contextuelle de taille 2, la séquence de mots (A B X C A) produira les 2 cooccurrences suivantes : (X, A_B_avant) et (X, C_A_après).

Multi-N-Gram : chaque mot cible est représenté par ses comptes de cooccurrences avec tous les N-Grams de mots possibles, avant et après lui. Les N-Grams de mots ont une taille variant entre 1 et Window Size. Par exemple, avec une fenêtre contextuelle de taille 2, la séquence de mots (A B X C A) produira les cooccurrences suivantes (X, A_avant), (X, B_avant), (X, A_B_avant), (X, C_après), (X, A_après), (X, C_A_après).

Distance type : Type de distance utilisé pour la recherche des plus proches voisins. Nous nous sommes limités, dans notre première expérimentation, aux deux types de distance les plus utilisées. Pour calculer la distance entre 2 mots, on utilise leur représentation vectorielle correspondante. Soit $v[v_1...v_m]$ et $w[w_1...w_m]$ 2 vecteurs issus de 2 lignes de la matrice de comptage, les distances possibles sont :

Cosine (distance du cosinus) : on utilise la distance du cosinus entre les 2 vecteurs v et w des mots que l'on souhaite comparer : $d(v,w)=1-\cos(v,w) = 1 - v.w / ||v||.||w||$.

Euclidian (distance euclidienne sur les vecteurs normalisés). On effectue une normalisation préalable de la matrice en colonne : au lieu de calculer, pour chaque ligne i et colonne j, le nombre de co-occurrence, nous estimons p(mot j/ mot i), avec j : attribut d'unité textuelle, et i : mot cible représenté. Un mot i est alors représenté par un vecteur de probabilité conditionnelle [p(attribut 1 /mot i),..., p(attribut M / mot j)]. La distance entre 2 vecteurs v et w est alors donnée par la distance euclidienne sur ces probabilités conditionnelles.

Language : type de langage utilisé. Dans notre expérimentation, nous nous sommes limités à 3 langues occidentales : anglais, français, espagnol.

3.3 Présentation du corpus choisi

Nous avons choisi La Bible comme corpus. Ce corpus existe avec de bonnes traductions dans de nombreuses langues. Ce corpus est en outre facilement et gratuitement disponible sur Internet, notamment à l'adresse http://bibledatabase.net/. La table 1 montre les caractéristiques générales des 3 traductions de La Bible que nous avons utilisées

	Anglais	Espagnol	Français
Taille (Ko)	4 356	4 031	4 208
Nombre de mots	852 010	795 978	838 294
Nombre de mots différents	13 585	31 531	21 849

TAB. 1 - *Caractéristiques du corpus de La Bible, pour les 3 langues étudiées.*

Noms propres		Noms communs		Formes verbales	
Aaron	Jesus	children	land	are	make
Abraham	Joseph	city	man	came	may
Babylon	Judah	day	men	come	said
Christ	LORD	days	name	did	say
David	Levites	earth	people	do	saying
Egypt	Lord	father	place	go	shalt
God	Moses	hand	son	had	was
Israel	Pharaoh	heart	sons	have	went
Jacob	Philistines	house	things	let	were
Jerusalem	Solomon	king	word	made	will

TAB. 2 - *Liste des 60 mots les plus fréquents de La Bible, selon 3 classes de mots générales.*

Nous avons utilisé 3 catégories grammaticales facilement séparables : noms propres, noms communs, formes verbales. Ces 3 catégories ont été utilisées pour former les listes de références dans chaque langue, de 3x20 mots. La table 2 donne en exemple la liste des 3 x 20 mots de référence utilisés pour la langue anglaise. Ces mots, associés à leur label, servent à calculer le critère de qualité de la représentation, détaillé plus haut. Nous avons choisi de prendre en compte les 2 premiers voisins de chaque mot dans l'espace vectoriel sémantique : le critère est donc calculé à chaque fois sur un graphe de 60 sommets et 120 arcs.

La gestion des ambiguïtés sur 2 classes (ici possible entre forme verbale et noms communs) a été faite d'une manière simple. Pour chaque mot ayant plusieurs significations possibles, nous avons d'abord étudié la possibilité de trouver tel ou tel sens dans le corpus. Par exemple, le mot "land" en anglais n'est pas utilisé comme verbe dans La Bible. Dans le cas où il restait une ambiguïté, nous avons étudié par échantillonnage (de 20 phrases) le sens utilisé le plus fréquent, et utilisé ce sens pour classer le mot.

4 Expérimentation

4.1 Principe de l'expérimentation

Notre expérimentation n'a pas la prétention d'être exhaustive : elle sert ici à illustrer les principes de la méthode générale présentée.

Tous nos tests ont été effectués sans réduction de dimension, sans suppression de mots considérés comme "creux" et sans lemmatisation. Les séparateurs de mots utilités étaient tous les séparateurs usuels (espace et signes de ponctuation). L'expérimentation a été faite pour les paramètres et les plages de valeurs décrits au paragraphe 3.2. Nous avons obtenus un total de 90 instances de modèle, chacun noté avec notre critère d'évaluation. Nous avons utilisé des graphes de voisinage avec k=2 afin d'avoir un voisinage relativement robuste. Nous avons ensuite sélectionné les modèles donnant les meilleures performances. Puis, nous avons comparés les scores obtenus selon le type d'unité textuelle utilisée, et selon le langage utilisé.

Pour les paramètres explorés, les scores vont de 50 (unité textuelle de type onlyOneNgram, fenêtre contextuelle de 3 mots, distance euclidienne, langage français) à 96 (unité textuelle de type MultiNgram, fenêtre contextuelle de 2 ou 3 mots, distance du cosinus, langage anglais).

4.2 Résultats

Les meilleurs performances sont obtenus, dans les différentes langues étudies, avec les paramètres suivants :
- anglais : unité textuelle de type MultiNgram, taille de fenêtre de 2 ou 3 mots, distance du cosinus (score de 96%)
- espagnol : unité textuelle de type ExactPosCtx, taille de fenêtre de 2 ou 3 mots, distance du cosinus (score de 92%)
- français : unité textuelle de type MultiNgram, taille de fenêtre de 3 mots, distance du cosinus (score de 90%)

Si nous utilisons l'unité textuelle de type MultiNgram, une taille de fenêtre de 3 mots, et la distance du cosinus, nous obtenons le meilleur compromis de performance pour les 3 langages : 96% pour l'anglais, 90% pour l'espagnol, 90% pour le français.

La figure 1 montre les performances moyennes obtenues sur les différents langages. La figure 2 montre les performances moyennes obtenues selon le type d'unité textuelle utilisée pour compter les co-occurrences avec les mots à représenter (moyennes calculées sur des plages de paramètres identiques)

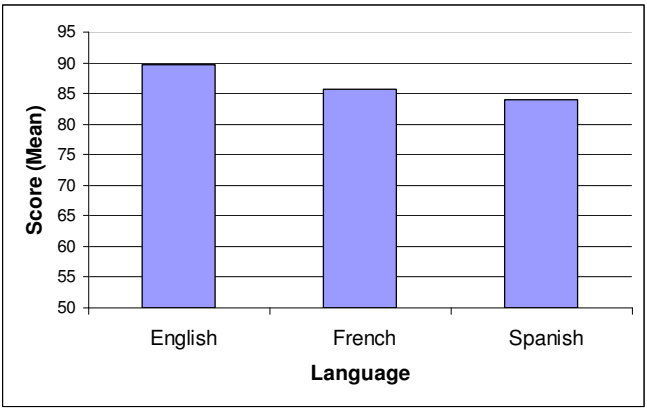

FIG. 1 : *Performances moyennes obtenues sur les différents langages.*

Notons que l'unité textuelle dite OnlyOneNgram correspond à un Ngram "classique" de N mots, N étant la taille de la fenêtre contextuelle. Les classiques Ngrams sont comptés respectivement avant et après le mot cible. Dans notre expérimentation, ces Ngrams classiques donnent des performances moyennes relativement faibles par rapport aux autres unités textuelles.

D'un autre côté, l'approche consistant à compter, dans une fenêtre contextuelle de N mots, les Ngrams de 1 mot, 2 mots... N mots donne de bonnes performances. La meilleure taille de fenêtre contextuelle semble être de 2 ou 3 mots avant et après le mot cible. Au delà d'une taille de 3 mots (avant/après), les performances des représentations décroissent rapidement, quelles que soient les valeurs des autres paramètres.

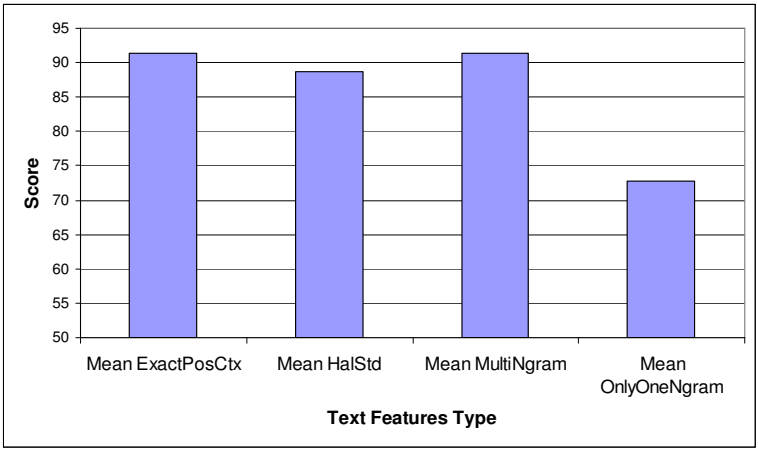

FIG. 2 - *Performances moyennes obtenues selon le type d'unité textuelle utilisée pour compter les co-occurrences avec les mots à représenter.*

Quel que soit le type d'unité textuelle utilisé, les performances les meilleures se situent pour des tailles de fenêtres courtes, avec un optimum à 3 (3 mots avants et 3 mots après le mot courant analysé). Pour l'unité textuelle de HAL, les performances décroissent ensuite régulièrement jusqu'à un score de 78% pour une taille fenêtre de 10.

La distance du cosinus est globalement meilleure que la distance euclidienne sur les fréquences des co-occurrences. Pour des tailles de fenêtres variant de 1 à 3, la distance du cosinus donne en moyenne des scores supérieurs d'environ 3% aux scores obtenus avec la distance euclidienne.

La langue obtenant en moyenne les meilleurs scores est l'anglais, par opposition au français et à l'espagnol. Ceci n'est pas surprenant car les déclinaisons grammaticales du français et de l'espagnol sont beaucoup plus nombreuses et complexes que celles de l'anglais.

4.3 Représentation des graphes de voisinage

Les graphes de voisinages (des mots de références), qui sont utilisés pour calculer notre score de qualité de la représentation, sont une source d'informations intéressantes pour vérifier, visuellement, les modèles sélectionnés par notre critère.

Pour la visualisation des graphes, nous avons utilisé une méthode de dessin de graphes dérivées des Self Organized Map (Kohonen, 1989) (Meyer 1998), implanté dans le package JUNG (JUNG, 2005). La position des sommets était déterminée automatiquement par cette méthode, en fonction des arcs reliant ces sommets. Nos graphes de voisinage comportaient,

pour chacun des 60 sommets, 2 voisins. La position des sommets était déterminée à partir du graphe complet (les arcs de second voisin sont donc pris en compte), mais pour limiter le nombre d'arcs affichés, nous ne dessinions que les arcs de premier voisin.

La figure 3 montre un graphe de voisinage sémantique en anglais (score=96%). Nous voyons qu'à l'intérieur des 3 classes prédéterminées (noms propres, noms communs, formes verbales), se dessinent naturellement des sous classes qui n'étaient pourtant pas explicitées dans le critère d'évaluation, notamment :

- proximité des lieux entre eux à l'intérieur des noms propres
- proximité des principaux personnages de La Bible
- proximité des verbes de mouvement à l'intérieur des formes verbales
- proximité des termes liés aux noms et à la filiation à l'intérieur des noms communs

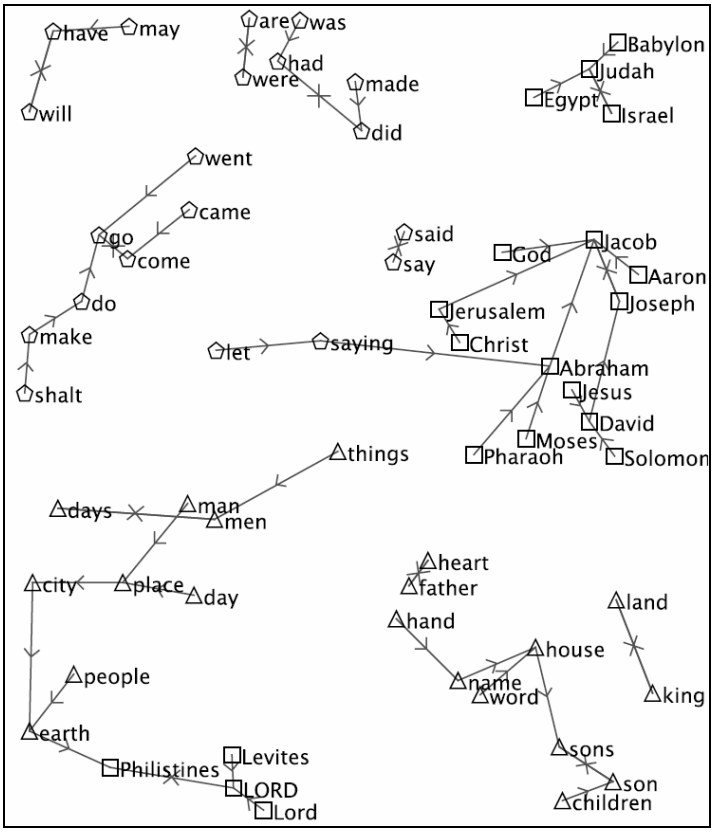

FIG. 3 - *Graphe de voisinage obtenu pour la langue anglaise, en utilisant des multi-3-grams, obtenant un score de 96% (arcs de 1er plus proche voisin représentés ; arcs de 2ème plus proche voisin pris en compte pour le placement).*

Nous avons obtenu des graphes similaires pour les langues française et espagnole, où nous retrouvons des groupements de sous-classes de mots non explicites dans le critère optimisé.

De même, si nous calculons ensuite des graphes de voisinage pour des mots en dehors de la liste de référence, nous retrouvons des regroupements dans des catégories de mots non explicités dans notre critère. Par exemple, sur le modèle réalisé en langue anglaise avec des Muti-3-grams, nous observons les relations de voisinage suivantes : regroupement de métaux (silver, brass, gold), regroupement de directions (north, south, west, east), regroupement de nombreux synonymes (sound, noise), (destroy, slay), (anger, wrath), etc.

5 Conclusion

Nous avons présenté une nouvelle approche pour explorer les paramètres dans les méthodes de représentation vectorielle de la sémantique des mots de type HAL. Nous avons illustré son principe par une expérimentation sur le corpus de La Bible sur différentes langues. Nos résultats vont dans le sens de ceux de Levy et Bullinaria (1998, 2001), en les étendant aux langues française et espagnole :

- les techniques de type HAL fonctionnent mieux avec des fenêtres contextuelles de petite taille (contrairement au paramétrage d'origine de HAL),
- la distance du cosinus semble préférable à la distance euclidienne,
- la réduction de dimension, si elle peut être utile pour des questions de calculs, ne semble pas indispensable pour obtenir de bons résultats,
- de même, lemmatisation et suppression de mots creux ne sont pas utiles (des premiers essais montrent qu'au contraire ces procédés font baisser les performances).

Nous avons montré que les Multi-N-Grams sont des unités contextuelles pertinentes pour des méthodes de type HAL. Nous avons noté de très bonnes performances avec ce type d'unité contextuelle pour des fenêtres courtes et une distance du cosinus, de manière stable sur les trois langues étudiées.

Par la suite, nous souhaitons approfondir notre méthode sur de plus grand corpus, et étudier des unités textuelles mixant des N-grams de lettres et des N-grams de mots.

Réferences

Church, K.W., et Hanks, P. (1989). Word association norms, mutual information and lexicography. *Proceedings of the 27th Annual Conference of the Association of Computational Linguistics*. Association for Computational Linguistics, New Brunswick, NJ, 76-83.

Deerwester, Dumais, Furnas, Landauer et Harshman (1990). Indexing by latent semantic analysis, *Journal of the American Society for Information Science*, 41(6), 391-407.

Eckart C. et Young G. (1936), The approximation of one matrix by another of lower rank, *Psychometrika*, Vol.1, pp 211-218.

Flint, S. et N. Chater, N. (1992). Bootstrapping syntactic categories. In *Proceedings of the 14th Annual Meeting of the Cognitive Science Society*, 820-825.

Golub, G. H., et Loan, C. F. V. (1983). Matrix computations, Johns Hopkins University Press, Baltimore.

Golub, G. H. et Reinsch, C. (1970). Singular ValueDecomposition and the Least Squares Problem. *Numer. Math*. 14, 403-420.

Hofmann, T (1999). Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–99*), pages 289–296, San Francisco, CA, Morgan Kaufmann Publishers.

JUNG, (2005). Framework Development Team JUNG. Jung: Java universal network/graph framework, 2005. http://jung.sourceforge.net/

Kohonen, T (1989). *Self-Organizing Maps*. Springer-Verlag.

Landauer, T et S. Dumais (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 2(104):211–40.

Levy, J.P., J.A. Bullinaria, et M. Patel (1998). Explorations in the derivation of word co-occurrence statistics. *South Pacific Journal of Psychology*, 1(10):99–111.

Levy, J.P. et J.A. Bullinaria (2001). Learning lexical properties from word usage patterns: Which context words should be used ? In Springer, editor, *Connectionist Models of Learning, Development and Evolution: Proceedings of the Sixth Neural Computation and Psychology Workshop*, pages 273–282, London. Springer.

Lund, K. et C. Burgess (1996). Producing high-dimensionnal semantic space from lexical co-occurrence. *Behavior Research Methods, Instruments& computers*, 2(28):203–208.

Meyer, Bernd (1998). Self-organizing graphs – a neural network perspective of graph layout, LNCS Vol 1547, Proceedings of the *6th International Symposium on Graph Drawing*.

Sahlgren, M et D. Swanberg (2001). Using linguistic information to improve the performance of vector-based semantic analysis. In *The 13th Nordic Conference on Computational Linguistics, NoDaLiDa '01*, May 2001.

Schütze, H (1993). Word space. In S. Hanson, J. Cowan, and C. Giles, editeurs, *Advances in Neural Information Processing Systems 5*. Morgan Kaufmann Publishers.

Summary

Statistics-based semantic representations of words are built on text units and words co-occurrences. These methods depend on complex parameters, such as the type of text units used as context. These parameters strongly impact the quality of the results. We focus on the parameters of Hyperspace Analogue to Language (HAL). We propose a new method to explore HAL's discriminant parameters. This method is based on a neighborhood graph analysis of reference pre-classified words. We experiment this method and give results which confirm and extend previous works.