TALN 2011, Montpellier, 27 juin – 1er juillet 2011

Couplage d'espaces sémantiques et de graphes pour le Deft 2011 : une approche automatique non supervisée

Yann Vigile Hoareau Murat Ahat Coralie Petermann Marc Bui
CHArt, 41 rue Gay Lussac, 75005 Paris
hoareau@lutin-userlab.fr, murat.ahat@etu.ephe.sorbonne.fr, coralie.peterman@laisc.net,
marc.bui@ephe.sorbonne.fr

Résumé. Nous décrivons l'approche mise en oeuvre dans le cadre du Défi de Fouille de Texte 2011 pour la piste 2 qui consistait à identifier, pour un article scientifique donné, le résumé qui lui correspond parmi un ensemble de résumés possibles. Cette approche est basée sur le couplage entre les méthodes d'espaces sémantiques pour la représentation des connaissances sémantiques d'une part, et les graphes pour la décision sur l'affectation d'un résumé à un article, d'autre part. La méthode proposée est entièrement automatique, sans phase de paramétrage, non-supervisée et ne nécessite aucune ressource externe.

Abstract. We describe our approach in Deft 2011 for track 2, which is to identify a corresponding summary, for a given scientific paper, from a set of possible abstracts. The approach is based on coupling the methods on the one hand, of semantic space for the representation of semantic knowledge, and, on the other hand, graphs for the decision on the allocation of a resume to a document. The proposed method is fully automatic, without any particular tuning, unsupervised and requires no external resources.

Mots-clés : Espace sémantique, Graphe, Random Indexing.

Keywords: Semantic Space, Graph, Random Indexing.

Introduction

Dans cette édition du Defi de Fouille de Texte 2011, nous avons appliqué une approche originale qui consiste à mixer deux méthodes de représentation des connaissances : les espaces sémantiques qui sont des espaces vectoriels à grandes dimensions et les modèles de graphes. L'intérêt du couplage des deux approches est de bénéficier d'une part des propriétés d'apprentissage non-supervisé ainsi que des propriétés sémantiques latentes associés aux espaces sémantiques et, d'autre part de la sophistication des mathématiques sous-jacentes à la théorie des graphes. Pour ce faire, la première contrainte à respecter est de produire un graphe ayant les mêmes propriétés que l'espace sémantique en ce qui concerne la représentation des relations sémantiques latentes entre les mots ou les documents (**?**). Cette contrainte satisfaite, des applications peuvent alors être réalisées directement à partir du graphe. Un exemple d'application de cette approche mixte est celui de la visualisation des relations sémantiques latentes entre documents au sein de grandes bases de données textuelles (**?**).

Dans la suite de cette article, nous décrivons comment nous avons appliqué cette méthode mixte pour la tâche 2 du Deft 2011. Cette méthode a été instanciée de telle sorte à représenter la relation sémantique entre chaque résumé et chaque article dans un graphe construit à partir d'un espace sémantique, puis à transformer ce graphe complet en un graphe biparti dans lequel chaque article est associé à un et un seul résumé.

L'article est organisé de la façon suivante. Dans la première section nous décrivons les espaces sémantiques et ainsi que l'approche qui consiste à représenter les documents sous la forme d'un graphe ayant les mêmes propriétés que l'espace sémantique. Dans la deuxième section, nous décrivons la chaîne de traitement mise en oeuvre pour implémenter notre méthode. Dans le troisième section, nous présentons très brièvement les résultats de notre approche pour les tâches 1 et 2 de la piste 2. Enfin, nous concluons l'article en présentant les perspectives de recherche qui pourraient prolonger le présent travail.

1 Le couplage espace sémantique et graphe

1.1 Les espaces sémantiques

Les modèles de représentation vectorielle de la sémantique des mots sont une famille de modèles qui représentent la similitude sémantique entre les mots en fonction de l'environnement textuel dans lequel ces mots apparaissent. La distribution de co-occurrence de mots est rassemblée, analysée et transformée en espace sémantique dans lequel les mots ou les concepts sont représentés comme des vecteurs dans un espace vectoriel de grandes dimensions. *Latent Semantic Analysis* (LSA) (**?**), *Hyper-space Analog to Language* (HAL) (**?**) et *Random Indexing* (RI) (**?**) en sont quelques exemples. Ces modèles sont basés sur l'hypothèse distributionnelle de **?** qui affirme que les mots qui apparaissent dans des contextes semblables ont des significations semblables. La définition de l'unité de contexte est un sujet commun à tous ces modèles, même si sa nature dépend du modèle. Par exemple, LSA construit une matrice mot-document dans laquelle chaque cellule a_{ij} contient la fréquence d'un mot donné i dans une unité de contexte j. HAL définit une fenêtre flottante de n mots qui parcourt chaque mot du corpus, puis construit une matrice mot-mot dans laquelle chaque cellule a_{ij} contient la fréquence à laquelle un mot i se retrouve avec un mot j en fonction d'une fenêtre flottante donnée. Différentes méthodes mathématiques et statistiques permettant d'extraire la signification des concepts sont appliquées à la distribution des fréquences stockées dans la matrice mot-document ou mot-mot. Le premier objectif de ces traitements mathématiques est d'extraire la tendance centrale des variations de fréquences et d'éliminer ce qui peut être considéré comme du « bruit » provoqué par la part d'utilisation spécifique de la langue associée à chaque scripteur. LSA emploie une méthode générale de décomposition linéaire d'une matrice en composantes principales indépendantes : la décomposition de valeur singulière (SVD). Dans HAL la dimension de l'espace est réduite en maintenant un nombre restreint de composantes principales de la matrice de co-occurrence. Des représentations vectorielles sont employées pour le stockage et la manipulation de la signification de concepts. À la fin du processus, la similitude entre deux mots peut être calculée selon différentes méthodes. Classiquement, la valeur du cosinus de l'angle entre deux vecteurs correspondant à des mots ou un groupe de mots est calculée afin d'approximer leur proximité sémantique. Une autre méthode équivalente est la distance euclidienne pondérée.

1.2 Random Indexing

Random Indexing (**?**) est un modèle d'espace sémantique qui a les mêmes propriétés que les modèles d'espaces sémantiques précédemment décrits comme LSA ou HAL. La différence avec ces deux modèles est que RI ne s'appuie par sur des méthodes de réduction matricielle mais sur les projections aléatoires. La méthode de construction d'un espace sémantique avec RI est la suivante :

– créer une matrice A ($d \times N$), contenant des *vecteurs-index*, où d est le nombre de documents ou de contextes correspondant au corpus et N, le nombre de dimensions ($N > 1000$) défini par l'expérimentateur. Les vecteurs-index sont creux et aléatoirement générés. Ils consistent en un petit nombre de (+1) et de (-1) et de centaines de 0 ;
– créer une matrice B ($M \times N$) contenant les *vecteurs-termes*, où M est le nombre de termes différents dans le corpus. Pour commencer la compilation de l'espace, les valeurs des cellules doivent être initialisées à 0 ;
– parcourir chaque document du corpus. Chaque fois qu'un terme τ apparaît dans un document d, il faut *accumuler* le vecteur-index correspondant au document d au vecteur-terme correspondant au terme τ.

À la fin du processus, les vecteurs-termes qui sont apparus dans des contextes (ou documents) similaires, auront accumulé des vecteurs-index similaires.

Le modèle a démontré des performances comparables (**?**) et parfois même supérieures (**?**) à celles de LSA pour le test de synonymie du TOEFL (**?**). RI a été aussi appliqué à la catégorisation d'opinion (**?**).

1.3 Le couplage Espace Sémantique–Graphe

Cette section décrit le processus de construction (i) d'un graphe complet représentant les propriétés sémantiques d'un espace sémantique, puis (ii) d'un graphe biparti à partir d'un graphe complet. Le procédé consiste à calculer la distance euclidienne pondérée entre chaque document de l'espace sémantique afin de construite une matrice de connexité. Cette matrice de connexité correspond alors à une représentation de l'espace sémantique sous la forme d'un graphe à N noeuds et N^2 arcs. L'intérêt de cette méthode très simple est de générer automatiquement un graphe qui a les mêmes propriétés que l'espace sémantique et de permettre ainsi d'y appliquer les méthodes issues de la théorie des graphes (**?**).

L'algorithme décrit ci-après a pour objectif de construire un graphe biparti à partir du graphe complet construit à partir d'un espace sémantique. Il prend en entrée un ensemble d'articles ou un ensemble de résumés pour les représenter dans un espace sémantique. Une matrice m "article – résumé" est construite. Cette matrice contient dans chaque cellule $m_{i,j}$, la valeur de la distance euclidienne pondérée entre les vecteurs de l'article i et du résumé j. À partir de cette matrice, un graphe g est produit. Ce graphe g peut être ambigu au sens où un résumé donné peut correspondre à plusieurs articles et *vice versa*. Un processus de désambiguisation est appliqué à ce graphe complet afin de produire un graphe biparti où, à un résumé ne correspond qu'un seul article et *vice versa*. Dans le cas d'une ambiguïté, la distance entre le résumé et l'article est initialisée à 0 et le prochain résumé le plus proche de l'article est recherché. Ce processus est itéré jusqu'à obtenir un graphe biparti. Il est illustré dans la section **??**.

```
Procedure main()
 Var
   A as Article Set;
   R as Resume Set;
   N as number of articles or resumes;
   m as Matrix Article Resume
   g as graph (article --> resume)

 Begin
   spaceSemantic = RandomIndexing(A, R)

   For (i:=1 to N)
     artVector = spaceSemantic(A[i]);
     For (j:=1 to N)
       resVector = spaceSemantic(R[j]);
       m[i,j] = cosine(artVector, resVector);
```

```
      End For; //j
    End For; //i

    g = createGraph(m);
    resolveAmbiguity(g)
 End Procedure //main()



Procedure createGraph(m);
  Var
    m as Matrix Article Resume
    g as graph (article --> resume)
  Begin
    g = emptyGraph();
    For (i:= 1 to N)
      j = Max(m[i,:]
      g.add(i,j);
    End
    Return g;
  End Procedure //createGraph()


Procedure resolveAmbiguity(g)
  While ambiguityExist(g)
    Do
      For (k:=1 to N)
        If (g.degree(resNode[k]) > 1) Then
           ambSet = all articles nodes connected to resNode[k];
           For (l:=1 to size(ambSet))
             If (m[ambSet[l],k] is not maximum) Then
               m[ambSet[l],k] = 0;
               g.delete(ambSet[l],k);
               newJ = Max(m[ambSet[l],:]);
               g.delete(ambSet[l],newJ);
             End
           End For
        End If
      End For
    End While
End Procedure
```

2 La chaine de traitement

2.1 Extraction des documents vers une Base de Données

La première étape a été l'indexation de l'ensemble des documents du corpus dans une base de données. Pour cela, une base de données relationnelle a été construite afin de stocker toutes les informations fournies en gardant leur structure et leurs liens. L'idée était de pouvoir facilement accéder à l'ensemble des données et modifier les unités de contextes utilisées pour l'apprentissage de l'espace sémantique (**??**). Un ensemble de scripts en langage PHP a été développé afin de réaliser ces tâches. Le système de gestion de base de données est MySql pour sa facilité de connexion avec les divers langages utilisés (Php, Java, etc.).

2.2 Indexation de la base de données

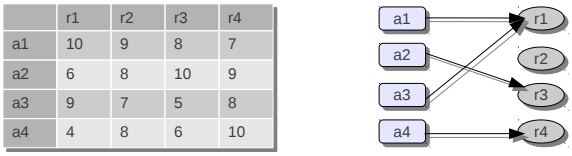
Afin d'automatiser entièrement notre chaîne de traitement, nous avons choisi d'intégrer LuSql [1] pour l'indexation du corpus. LuSql est une application java en ligne de commande permettant de construire des index Lucene à partir de bases de données relationnelles. Il permet de sélectionner précisément les données à indexer en passant en argument de la ligne de commande la requête SQL utilisée. De plus, dans son mode par défaut, il utilise le *multithreading* pour s'exécuter sur plusieurs processeurs et donc optimiser les temps d'indexation. Toujours en vue d'automatiser nos outils, nous avons développé une interface java intégrant LuSql.

2.3 Construction de l'espace sémantique

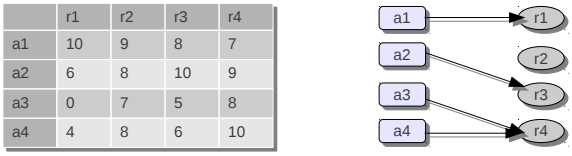
L'espace sémantique a été construit en utilisant la méthode Random Indexing implémenté par la librairie Java SemanticVectors (**?**). Les unités de contexte servant à l'apprentissage sont les documents tels que définis dans le corpus d'apprentissage : l'article et le résumé. Le corpus n'a fait l'objet d'aucun traitement de lemmatisation (**??**). Les mots vides ont été supprimés. L'espace sémantique RI a été construit avec 1000 dimensions. L'ensemble des traitements a été réalisé sur un ordinateur portable à 2 Go de RAM, intel core2 duo cpu à 2.00ghz.

2.4 Construction du graphe biparti

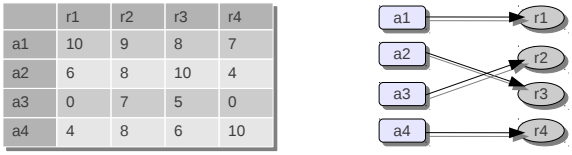
Dans cette étape, nous avons appliqué l'algorithme de la section **??**. Le graphe construit à partir de l'espace sémantique n'avait que peu d'ambiguïtés. Nous attribuons cela d'une part à la qualité des données et d'autre part, à la robustesse de *Random Indexing*. Le processus de résolution des ambiguïtés a permis construire le graphe biparti avec chaque noeud à un degré de 1. Dans la Figure **??**, nous illustrons l'ensemble du processus de la construction du graphe et de résolution des ambiguïtés.



(a) Etape 1 : Graphe avec ambiguïté



(b) Etape 2 : Graphe avec ambiguïté



(c) Etape 3 : Graphe biparti sans ambiguïté

FIGURE 1 – Exemple de construction d'un graphe biparti

1. http ://lab.cisti-icist.nrc-cnrc.gc.ca/cistilabswiki/index.php/LuSql

3 Les résultats

Les performances de la méthode pour les pistes 1 et 2 représentées dans le Tableau **??** sont très satisfaisantes. La méthode réalise un score de 1 pour la tâche 1 et de 0,995 pour la tâche 2.

	Tâche 1 (articles complets)	Tâche 2 (contenu)
Exécution	**1,000**	**0,995**
Moyenne	0,981	0,956
Médiane	0,990	0,959
Ecart-type	0,027	0,042

TABLE 1 – Scores pour les tâches 1 et 2

4 Conclusion

La méthode proposée dans le cadre de notre participation au Deft 2011 repose sur le couplage entre les espaces sémantiques et les graphes. Le faible nombre de documents disponibles pour l'apprentissage constituait une contrainte forte pour notre méthode entièrement basée sur une approche distributionnelle. Les résultats indiquent que la méthode n'a pas souffert de cette contrainte.

Les méthodes d'espaces sémantiques telles que LSA, utilisant des techniques de réductions matricielles nécessitent un nombre important de documents lors de l'apprentissage afin que les processus de réduction puissent s'appliquer de façon efficace. Ainsi, il nous semble fort probable que la robustesse de notre approche serait la conséquence de l'utilisation de *Random Indexing* qui est basé sur les projections aléatoires.

De prochaines expériences seront réaliser afn de tester cette hypothèse car ses implications pourraient s'avérer importantes pour la recherche appliquée : il serait alors possible de tirer profit des propriétés des espaces sémantiques à partir d'un corpus même très limité en nombre.