Proposition VSST 2012 : Une approche prétopologique pour la catégorisation des données micro-blogging

Karim Sayadi, Marc Bui, Vigile Hoareau
Equipe CHArt EA-4004
LaISC EPHE & Univ. Paris 8
41, rue Gay-Lussac, F-75005 Paris

Karim.Sayadi@laisc.net
Marc.Bui@laisc.net
Vigile.Hoareau@laisc.net

Soufian Ben Amor
Laboratoire PRISM – UMR8144,
Univ. de Versailles-St-Quentin-en-Yvelines,
45, avenue des États-Unis, F-78035 Versailles

Soufian.Ben-Amor@prism.uvsq.fr

Résumé

Dans ce papier, nous traitons la question de la détection des thèmes pour de grand corpus de données textuelles non structurées, et plus spécifiquement des données de micro-blogging. Dans ce travail intitulé *Topological Analysis of Dynamic Semantic Space* (TADySS), nous effectuons un couplage entre la construction d'un espace sémantique à partir d'une base de tweets, fournie par le réseau social de micro-blogging *Twitter*, et une approche prétopologique que nous avons développée pour suivre l'évolution des thèmes au cours du temps. Nous partons d'un ensemble d'informations non structurées sous une forme brute, et adoptons un processus cyclique pour catégoriser l'information. L'espace sémantique des tweets est généré à partir de l'indexation de l'ensemble des tweets avec la bibliothèque *Lucene*. Le corpus numérique traité à titre illustratif est d'une taille de 12Go et est constitué de plusieurs fichiers représentant plusieurs millions d'éléments. Ces fichiers issus de la base de données des tweets sont bruités et comportent de nombreuses informations inutiles pour la catégorisation. Nous appliquons donc un *parser* que nous avons développé spéficiquement afin d'extraire les données utiles. Nous procédons ensuite à l'indexation et à la construction de l'espace sémantique à l'aide de l'algorithme *Random Indexing* qui est implémenté à l'aide du *package semantic vectors*).La proximité sémantique entre les tweets repose sur le calcul du cosinus entre les deux vecteurs représentant chaque tweet dans l'espace sémantique, l'appartenance d'un tweet donné à une catégorie étant, quant à elle, déterminée en calculant la similarité entre le tweet considéré et les représentants de chaque catégorie. Le processus est constitué par une boucle de traitements réalisant des allers-retours entre l'index et l'espace sémantique jusqu'à ce que l'ensemble de départ soit divisé en plusieurs ensembles représentant chacun un thème.L'algorithme prétopologique développé, présente un intérêt pour de nombreuses applications. Parmi les plus intéressantes, nous pouvons citer la détection de l'émergence d'événements importants dans l'actualité, ou la détection des communautés virtuelles d'individus associées aux thèmes de discussions.

Abstract

In this paper, we propose a solution to automatically discover the topics from a collection of unstructured text data. In this work entitled Topological Analysis of Semantic Space (TADySS), we perform data analysis by using an association between a semantic space of tweets, provided by the social network Twitter, and a pretopological approach that we have developped to track topics over time. We start from set ofunstructured information and we treat our data through a cyclic process. This cyclic process defines the hidden topic structure of the the explored set of information. The processed digital corpus is about 12 GB of textual files representing several million of elements. These files downloaded from the database of Twitter contains many useless informations. Thus, we have applied a parser that we have developed to extract useful data. The files were indexed using the Apache Lucene package. The algorithm Random Indexing implemented in the Semantic vectors package was then applied to build the semantic space. The semantic proximity between the tweets is based on the calculation of the cosine between two vectors representing each one a tweet in the semantic space. The cyclic process consists of treatments loop performing round trips between the index and the semantic space. Along the number of trips, the starting set is divided into several sets, each one of them represents a topic. The developped pretopological algorithm has many applications. Among the more interesting, we can mention for example the detection of emmergent topic structure, or the detection of virtual communities which discussions are associated with a set of topics.

I. Introduction

Avec l'arrivée du web 2.0 et la numérisation massive des connaissances sous forme de blog, pages web, micro blog (Twitter), articles scientifiques, photos, vidéos, etc., le réseau internet est passé d'un réseau statique à un réseau dynamique avec des données hétérogènes (multimédia, multilingue, distribuées). Nous ne pouvons plus imaginer le parcours entier du web, afin de procéder à la recherche d'informations [1] ou au calcul de sa taille.

Avec l'augmentation de la quantité des données à traiter, il faut aussi changer la manière avec laquelle nous présentons les résultats. L'un des fondateurs du web, Tim Berners-Lee, a proposé en 2001 de faire évoluer le web vers un web sémantique [2], c'est à dire un web qui permet aux ordinateurs d'analyser le sens des informations qu'il contient. Le *web sémantique* est fondé sur RDF(Resource Description Framework) et XML mais aussi sur des techniques et des algorithmes capables de gérer de grandes quantités de textes à valeur sémantique. Ces techniques ont été proposées dans le cadre de la fouille de texte, de l'intelligence artificielle, du traitement du langage naturel et de la recherche d'informations.

Le projet TADySS (*Topological Analysis of Dynamic Semantic Space*) vise au développement des applications de détection d'événements dans des communautés virtuelles ou de détection d'évolution de structure de communautés virtuelles. En partant du principe de la compréhension de texte, un modèle des sciences cognitives [3], nous construisons l'hypothèse suivante : le processus cyclique de compréhension permet de détecter les thèmes de discussion dans les communautés virtuelles et de les suivre dans le temps.

Dans ce travail, nous reconsidérons la fouille de texte en adoptant une nouvelle vision de la recherche d'informations, basée sur l'exploitation des thèmes présents dans les corpus de texte et non plus sur les mots-clés. Les résultats proposés à la fin de chaque fouille de texte présenteront un suivi des thèmes[4] et une prédiction sous forme de détection des thèmes émergents. Pour cela, nous proposons une approche originale. L'algorithme que nous avons développé dans le cadre du projet TADySS est une association entre la prétopologie et la recherche sémantique. Pour l'évaluation de notre travail nous avons utilisé des données de microblogging fournies par le réseau social Twitter. D'autres chercheurs les ont également utilisées [5] pour des fins tout aussi prédictives [6] [7]. L'article est organisé de la façon suivante. Dans la première section nous décrivons les espaces sémantiques, ainsi que l'approche entreprise pour faire le couplage avec la prétopologie, nous présenterons dans ce qui suit notre algorithme qui est né de cette association. Dans la deuxième section, nous expliquons les démarches effectuées pour l'évaluation de l'algorithme avec les données de microblogging. Ces données ont nécessité un traitement important à l'aide d'outils spécifiques que nous décrivons dans cette même section. Dans la troisième section, nous présentons les résultats obtenus et enfin, nous concluons en présentant les prolongements envisagés de ce travail.

II. Le couplage espace sémantique et prétopologie

Nous avons besoin de nouveaux outils pour la fouille, l'organisation et l'annotation des archives électroniques qui s'élargissent à fur et à mesure du développement du web surtout avec la présence grandissante sur la toile des réseaux sociaux à l'image de *Facebook* et *Twitter*. L'annotation par mot clé de ces corpus numériques s'avèrent de moins en moins efficace. Une annotation par thème permet une exploration plus efficace. Nous pourrons naviguer dans le temps pour voir quels thèmes ont été traités ou qui sont actuellement traités, voir quel thème nous intéresse et par la suite consulter les documents qui y sont reliés. C'est à cette fin, que des chercheurs dans le domaine de l'apprentissage non supervisé ont développé des modèles probabilistes [4] pour l'annotation des documents suivant les thèmes. Par définition un thème est un ensemble de mot suivant un vocabulaire fixé.

Le but de la modélisation de la structure thématique des documents est de découvrir automatiquement les thèmes dans une collection de documents. LDA (*Latent Dirichelet Allocation*) comme d'autres modèles fait partie de ce domaine de modélisation probabiliste. L'idée principale de LDA est qu'un document est un ensemble de thèmes. La caractéristique particulière de LDA c'est que tout les documents traités partagent le même ensemble prédéfini de thèmes. La sortie de l'algorithme est la différence de proportion de chaque thème dans les différents documents.

La recherche thématique comme on le constate représente un enjeu de premier plan. Dans ce travail nous allons une voie particulière celle du couplage espace sémantique et prétopologie qui est différente des modèles existant qui sont des modèles probabilistes.

A. Les espaces sémantiques

Les espaces sémantiques constituent une famille de méthodes mathématiques qui permettent de représenter la similarité sémantique de mots, de documents ou de concepts à partir de l'analyse de co-occurrences de termes établis à partir de grands corpus (plusieurs dizaines de milliers de documents). La plus importante propriété des espaces sémantiques est la métaphore topologique : les mots proches dans l'espace sémantique représentent des mots proches du point de vue du sens. Les différentes méthodes de construction d'espaces sémantiques utilisent pour certaines des méthodes issues de l'algèbre linéaire, des probabilités ou des projections aléatoires [8]. Les espaces sémantiques

représentent les mots ou les documents sous la forme de vecteurs dans un espace vectoriel de grande dimension : les mots ayant un sens similaire sont représentés avec des vecteurs ayant des directions proches. Ces méthodes ont montré leur efficacité dans le cas de la catégorisation thématique de textes. Depuis, ces modèles ont également été appliqués efficacement à la catégorisation d'opinions.

B. L'analyse prétopologique sémantique (PSA)

L'idée du projet TaDYSS est que le suivi des thèmes présents dans de grands corpus de données, et le traitement prétopologique de ces derniers permettent la détection de thèmes émergents. Dans le cadre de ce projet, nous proposons un algorithme prétopologique nommé PSA (*Pretopological Semantic Analysis*), algorithme qui réalise le couplage entre la sémantique et les principes de la prétopologie [9] [10]. L'approche prétopologique se réalise par l'intermédiaire de la définition d'une fonction d'adhérence qui permet de structurer le corpus en thèmes à partir de d'items du corpus identifiés et représentant chaque thème. Cette structuration du corpus en thèmes se déroule pas à pas au fur et à mesure de l'examen de celui-ci.

Nous présentons par la suite l'anatomie algorithmique et mathématique de cette proposition.

C. Modélisation

L'objectif global de l'algorithme est de découvrir automatiquement les thèmes pertinents et d'identifier les thèmes émergents à partir d'un corpus de texte. Le thème est défini comme étant une distribution sur un vocabulaire (ensemble fini de mots) présent dans le corpus.

Nous définissons par E l'espace sémantique construit avec l'algorithme *Random Indexing* [11] à partir de l'ensemble des données textuelles où chaque entité de type document (phrase, paragraphe, article), est représentée par un vecteur t. La proximité sémantique entre deux vecteurs t_i et t_j est définie à l'aide de la distance sémantique d_s introduite par l'espace sémantique E. Elle représente la valeur du *cosinus* entre deux vecteurs choisis, par exemple t2 et t3 dans la figure 4. L'espace sémantique E nous permet de définir par un apprentissage non supervisé la proximité entre mots et documents.

L'introduction de la prétopologie par l'intermédiaire de la fonction d'adhérence définit pas à pas la mise à jour de chaque thème. La fonction d'adhérence $a(.)$ collecte les entités textuelles en utilisant la distance sémantique d_s entre le vecteur t_i et le vecteur de l'individu de référence présentant un thème. Un sous-ensemble e des unités de textes collectées par la fonction $a(.)$ est créé.

Pour la détection des thèmes majeurs nous définissons un seuil δ_t. La comparaison entre le cardinal de e, le sous-ensemble généré, et le seuil δ_t détermine si le thème est pertinent ou non. L'ensemble E est alimenté au fur et à mesure avec de nouveaux vecteurs. En raison des capacités limitées de stockage et de traitement, une fenêtre temporelle glissante est construite. Cette fenêtre élimine les thèmes détectés au bout d'un certain moment au profit d'autres thèmes plus d'actualité.

Pour la recherche du thème émergent, nous effectuons deux recherches sémantiques avec à chaque fois comme entrée de la recherche, trois sous-ensembles correspondant aux trois thèmes collectés. Nous comparons les distances sémantiques d_s des résultats obtenus. Le résultat qui a la même d_s dans les résultats des deux recherches est considéré comme un thème émergent et on applique sur ce dernier la fonction d'adhérence $a(.)$ pour la construction du sous ensemble b.

D. Principe de l'algorithme

Notre algorithme implémente une fonction prétopologique construite en nous inspirant du processus de compréhension de texte. Il est consiste en la définition d'un processus cyclique où la représentation globale du texte est mise à jour au fur et à mesure de la progression de la lecture. Le traitement effectué par le processus utilise en entrée un réseau d'unités syntaxiques simples et en relation.

Nous partons d'un grand ensemble de données textuelles. Nous définissons formellement par la suite un thème à partir de la distribution des mots présents dans le corpus. Ce sujet sera le déclencheur du processus cyclique qui fera apparaître les thèmes les plus présents dans le corpus et détectera les sujets émergents, qui prendront par la suite une place plus importante dans l'ensemble de données. Le processus prend en entrée un corpus de texte, et calcul en sortie les thèmes les plus importants et les thèmes émergents, qui seront les plus traités dans le futur.

L'algorithme procède en deux phases :

- Recherche des thèmes les plus traités dans le corpus de texte analysé
  a. Injection d'un thème initial choisi aléatoirement.
  b. Récupération des thèmes avec la proximité sémantique la plus forte.
  c. Retour à l'index de départ pour la collecte des entités de textes correspondantes aux thèmes apparus avec une proximité sémantique forte.
  d. Construction des sous-ensembles sémantiques à partir des entités de textes collectés.

- Recherche des thèmes émergents
  a. Association de plusieurs sous-ensembles sémantiques représentant les différents thèmes détectés dans la première phase
  b. Si un thème apparaît à la même position par rapport à l'ensemble des thèmes choisis dans le nombre de recherches effectuées
     - Retour à l'index de départ pour la collecte des entités de textes correspondantes ;
     - Construction d'un ensemble émergent.
  c. Arrêt du processus, si tous les thèmes détectés ont été traités ;
  d. Si le corpus évolue, le thème détecté sera le déclencheur du nouveau processus.

Dans la suite, nous présentons un pseudo code du principe de l'algorithme que nous venons de décrire.

À chaque itération du processus cyclique de l'algorithme il prend comme entrée le corpus de texte E et en sortie il nous donne e thèmes majeurs et un (ou des) thème(s) émergent(s) s'il y en a.

Algorithm 1 Algorithme PSA

Require: corpus de documents E // Entrée
Ensure: b ensemble des thèmes émergents, e ensemble des thèmes pertinents // Sortie
// K ensemble des tweets collectés, seuil_tweet est la variable seuil à partir de laquelle le thème est considérer comme pertinent, d_s est la distance sémantique entre deux tweets.
Variable : K = ∅, seuil_tweet ← x, d_s ← 0
// x est le nombre de tweet qu'on choisit en fonction du nombre de tweets que nous avons dans le corpus.

```
//Recherche des thèmes les plus traités
theme_temp ← theme_choisi_au_hasard ;
while (parcours de E n'est pas terminé) do
    list_theme [] ← chercher_theme (theme_temp) ;
end while
for i ← 0 jusqu'a taille(list_theme) do
    K ← collecte_tweet (list_theme[i]) ;
end for
if (card(K) > = seuilTweet) then e ← K ;
end if
//Recherche des thèmes émergents
trois_theme1 ← Association (e1, e2,e3) ;
trois_theme2 ← Association (e4, e5,e6) ;
list_theme1 [] ← chercher_theme (trois_theme1) ;
list_theme2 [] ← chercher_theme (trois_theme2) ;
for j ← 0 jusqu'a taille(list_theme1) do
    if la ds list_theme1[j] est egale à la ds list_theme2[j] then b ← collect_tweet(list_theme[j])
    end if
end for
```

Nous passons maintenant à la description du travail fait sur les données de *micro-blogging* pour l'évaluation de notre algorithme et la discussion de ses résultats.

III. Application sur des données de micro-blogging

Quatre milliards de réponses par mois, à une seule question posée au dessus d'un champ de texte vide sur le site de *micro-blogging* Twitter "Qu'est ce qui se passe ?". Cette question anodine, n'intéresse pas seulement les utilisateurs de cette plateforme web, mais aussi les analystes économiques [7], les politiciens [6], les journalistes, et notamment toute l'industrie financière. En ce moment même, il y a des millions de voix qui s'élèvent pour nous dire et nous informer de ce qui est entrain de se passer, mais pour les entendre il faut savoir traiter les données qui s'offrent à nous, relever l'importance de certaines informations et la non-pertinence d'autres.

L'entité de texte offerte – le tweet – fourni par le réseau social de *micro-blogging* Twitter, est un matériau abondant, riche en informations par sa structure particulière. Pour l'évaluation de l'algorithme PSA, notre choix s'est porté sur un corpus de texte constitué de ces entités particulières. Pour le traitement, l'extraction et le filtrage des données, nous avons construit notre propre *parser*. D'autres outils comme *Lucene* ou encore *semantic vectors* ont été également utilisés pour la construction de l'index et de l'espace sémantique, ils seront présentés un peu plus loin dans l'article.

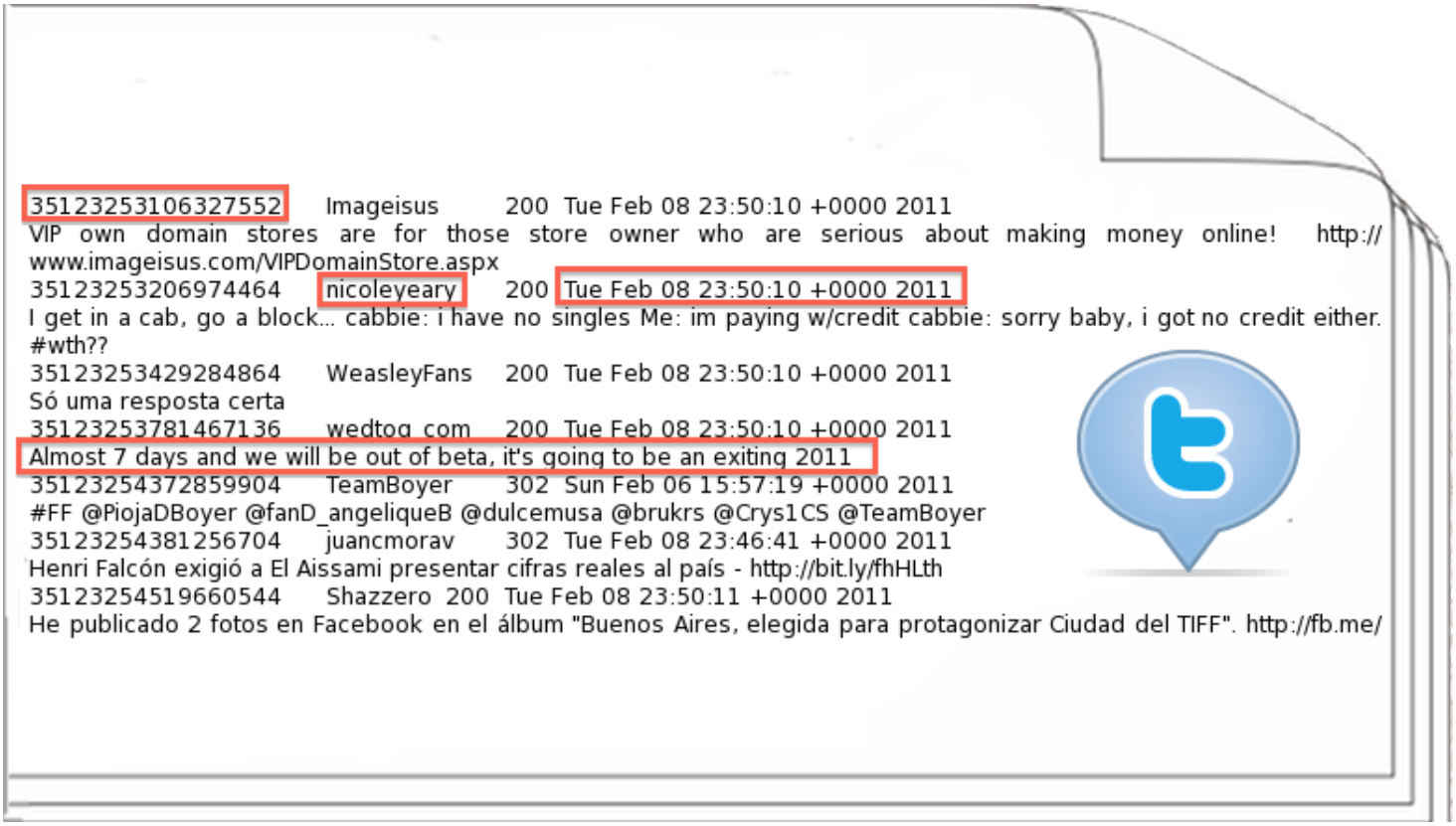

FIGURE 1: Extrait d'un des fichiers texte fournit par le site de micro-blogging Twitter. Ce fichier est téléchargé après l'utilisation d'un *crawler* intégré dans le *parser*. Ce dernier va extraire les informations encadrées en rouge, ici dans l'image, pour la construction de l'index *Lucene*

A. Présentation de Twitter

Twitter est un réseau social de *micro-blogging*, qui permet d'envoyer des messages courts de 149 caractères maximum. Il est souvent décrit comme un réseau social d'élite, où il y a une forte concentration de journalistes, d'artistes et d'ingénieurs. Twitter a acquis grâce à ces communautés co-existantes une certaine identité qui le différencie des autres réseaux sociaux, comme Facebook. Considéré le plus souvent comme un réseau informatif, Twitter permet de relater l'information à une vitesse supérieure à celle de la presse écrite ou audiovisuelle, on parle même de l'émergence d'un nouveau concept informatif. Cette propagation particulière de l'information sur Twitter est due à la structure même de son réseau. Dans un premier temps, un *tweet* (un message écrit sur Twitter) est généralement diffusé en public sur internet, ainsi, que dans l'accueil des *followers*[1] de l'auteur du message. Tout le monde peut donc ReTweeter (relater) les messages écrits par une personne en précédent le *tweet* par un RT, ou commenter le tweet en le mettant en citation. Ils peuvent aussi répondre à l'auteur du *tweet* en faisant précéder le pseudo de ce dernier par un . Dans un second temps, un *hashtag* est le plus souvent créé, lors d'un événement ou d'un fait divers suivi par une des communautés. Ce *hashtag*, précédé par un #, informe sur le sujet du *tweet* qui le précède. Le concept de hashtags participe à la propagation de l'information. Les *hashtags* permettent d'étiqueter les *tweets*, et ainsi les regrouper par sujet.

B. La chaîne de traitement des Tweets

Notre travail, repose sur l'algorithme *Random Indexing* analogue à l'algorithme *LSA* (*Latent Semantic Analysis*). Contrairement à *LSA*, le *Random Indexing* n'utilise pas un algorithme de réduction des dimensions (*Singular Value Decomposition* SVD) qui consomme beaucoup de ressources et est moins performant face à un grand volume de données. RI permet d'éviter de générer la matrice termes-documents en générant directement des vecteurs de taille réduite. La figure 3 décrit la phase de transformation des Tweets se trouvant dans les documents vers les espaces sémantiques contenant des vecteurs termes et des vecteurs documents. RI procède en deux étapes :

1. Les utilisateurs de Twitter qui se sont inscrits aux mise à jour d'un autre utilisateur

FIGURE 2: Execution de la première phase de l'algorithme PSA, détection des thèmes reliés aux deux sujets ; "islamist" et "crisis"

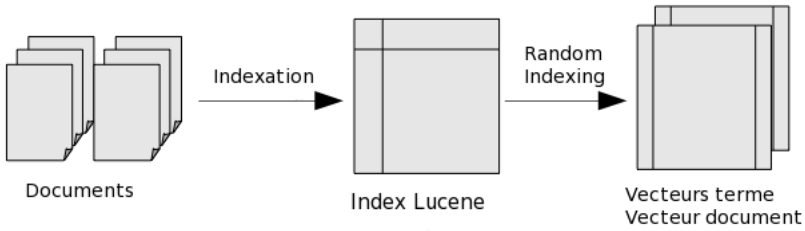

FIGURE 3: Chaine de traitement des tweets

- Premièrement, chaque contexte (c.à.d. chaque *tweet* ou chaque mot dans le *tweet*) dans les données est assigné à un unique vecteur index généré aléatoirement.
- Deuxièmement, les vecteurs sont produits en effectuant un balayage sur le texte, et à chaque fois qu'un mot se répète dans un contexte (par exemple : dans un document, ou dans une fenêtre contextuelle glissante), un vecteur index avec une dimension est ajouté au vecteur contexte pour le mot considéré. Les mots qui sont représentés par un vecteur d'index contextuel avec une dimension d sont la somme effective des contextes de mots.

Une fois l'espace sémantique construit avec à l'intérieur les vecteurs termes et contexte, nous pouvons procéder au calcul de la proximité sémantique entre chaque vecteur. Pour cela, nous calculons le *cosinus* entre les deux vecteurs. Dans la figure 4, le cosinus entre les *tweets* t2 et t3 est inférieur aux autres cosinus. Les deux *tweets* t2 et t3 sont considérés comme proches sémantiquement.

C. Outils et préparation des données

1) Le parser : Lors de la construction de l'index *Lucene*, nous avons été confrontés à un problème majeur dû à la structure même de l'index. En effet, pour construire un index *Lucene* comme il est schématisé dans la figure 5, il faut

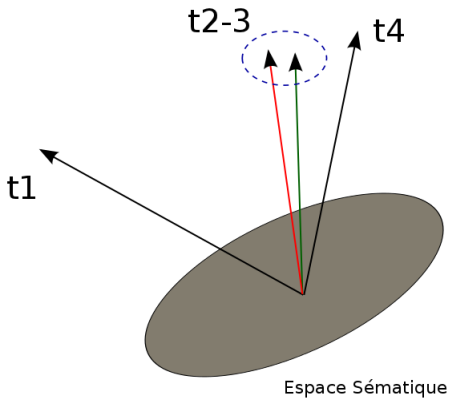

FIGURE 4: le calcul de la proximité des *tweets*

tout d'abord initialiser un ensemble de variables appelées *fields* qui vont constituer les *Documents*[2]. Nous avons, de ce fait, et pour chaque tweet, construit un document avec plusieurs *fields*, l'un pour le tweet, l'autre pour le pseudo, un pour la date de la publication du tweets, etc. La base de données fournie par le site de *micro-blogging* Twitter, n'est pas structurée, très bruitée et ne permet pas d'importer les informations qu'elle contient directement dans les méthodes que nous avons développées pour l'indexation. Le besoin de filtrer cet ensemble d'informations sans consommer beaucoup de ressources est vital, car nous traitons un large ensemble de données. De ce fait, nous avons intégré dans la ligne du projet le développement d'un *parser*, qui va extraire les données, et effectuer un filtre pour éliminer tous les bruits, inutiles à l'indexation.

2) Apache Lucene : Pour la construction des index, nous avons utilisé la librairie Java *Lucene*. *Lucene* est une librairie qui offre des paquets permettant l'ajout des fonctions d'indexation et de recherche d'information dans le programme développé. La librairie peut être vue comme un noyau adapté pour une application de recherche, où l'extraction de l'information et l'interface graphique viennent se greffer autour. *Lucene* est notamment utilisé dans plusieurs plateformes : NetFlix, Digg, MySpace, Fedex, ...

3) Semantic Vectors : Semantic Vectors est une librairie qui offre des algorithmes pour la construction d'espaces sémantiques. La construction des espaces prend comme entrée l'index construit avec *Lucene*. *Semantic Vectors* implémente l'algorithme *Random Indexing* [11] qui produit un résultat identique que le *LSA* [12]. Cependant, il est plus performant, plus rapide, et demande beaucoup moins de ressources mémoire lors du traitement de grandes quantités d'informations issues comme dans notre cas avec Twitter.

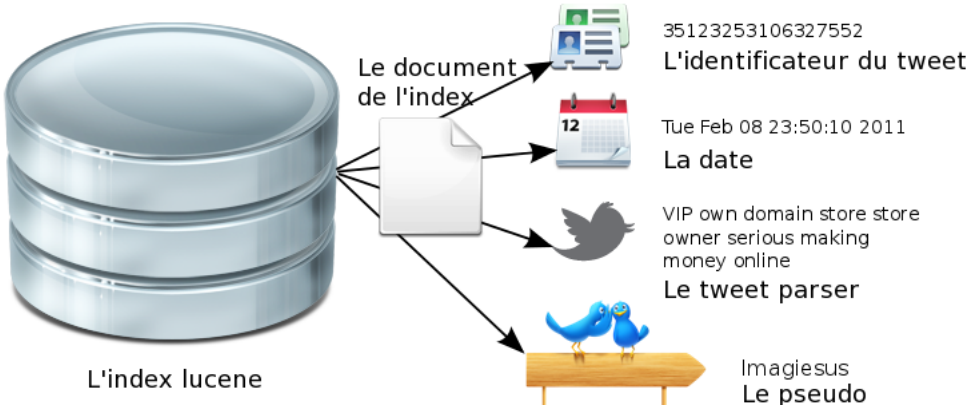

FIGURE 5: Schéma du contenu de l'index *lucene* construit avec le *parser*

4) Préparation des données : Twitter fournit un grand ensemble de données non structurées. Les fichiers fournis par la plate-forme sont en format binaire. Pour les extraire dans un format texte, il faut exécuter un *crawler*[3], ce dernier est

2. Documents est l'entité des index Lucene
3. Un agent informatique qui parcoure l'adresse d'un site internet pour extraire des données

intégré dans le *parser*. Le *crawler* exécute une requête qui en réponse va télécharger les fichiers `.txt` correspondants. Comme le montre la Figure 1, une fois téléchargés ces fichiers contiennent les informations relatives à chaque *tweet* à savoir l'identificateur du *tweet*, le pseudo de l'utilisateur, la date et enfin le *tweet* même. Ces informations sont ensuite extraites par le *parser* et utilisées pour la structuration de l'index. Ensuite, le *parser* les utilise pour la construction de l'entité de l'index, le *Document*. L'identificateur du *tweet* sera utilisé comme identificateur du *Document*, le pseudo et la date comme des informations complémentaires. Des informations qu'on pourra introduire comme options pour filtrer notre recherche sur l'ensemble de l'index. Et enfin le *tweet* qui est l'entité textuelle sur laquelle sera effectuée la recherche. Avant d'être ajouté dans le *Document* le *tweet* passe par un filtre dans le *parser* qui éliminera les *stop words* comme {la, les, à, aux, ...} et les différents bruits comme { §, @, %, etc ..}.

Au final 10 millions de *tweets* correspondant à 10 jours entre le mois de janvier et le mois de février 2011 sont indexés. L'index de 4Go est ensuite utilisé pour la construction de l'espace sémantique en utilisant le *Package Java Semantic vectors*. L'espace sémantique construit contient 14 754 400 termes et a une taille de 14Go. Une fois l'index et l'espace sémantique de départ construit nous lançons l'algorithme PSA avec notre fonction prétopologique implémentée. Les résultats de l'exécution des deux phases de l'algorithme sont discutés dans la section qui suit.

IV. Résultats

Table I: Les thèmes associés aux sous ensembles sémantiques relatifs aux six sujets suivants : *mubarak, egypt, revolution* et *revolution, islamist, radical*

0.9012648083322518	egypt
0.8916136576453128	mubarak
0.8916136576453128	hosni
0.8916136576453128	revolution
0.8916136576453128	seeking
0.7543718195375027	freedom
0.6363650229440665	free
0.6026549380611794	violent
0.5921409380611794	islamist
0.5916136765453128	jfk
0.5921409380611794	free
0.5367401449989081	people
0.36257685486326147	peaceful
0.34115381376593884	jan

0.9012648083322518	islamist
0.8916136576453128	fundamental
0.8916136576453128	monster
0.8916136576453128	radical
0.8916136576453128	seeking
0.7543718195375027	freedom
0.6363650229440665	free
0.5921409380611794	violent
0.5921409380611794	jfk
0.5921409380611794	impossible
0.5921409380611794	inevitable
0.5367401449989081	people
0.5296269416745712	make
0.36257685486326147	peaceful
0.34115381376593884	jan
0.3290919906317447	revolution
0.285907913563485	egypt

A. Discussion

Comme nous l'avons expliqué précédemment l'algorithme est divisé en deux phases, le résultat de la première phase, à savoir la recherche des thèmes les plus récurrents dans le corpus, est présenté dans la figure 2. Nous lançons le processus

en incluant dans la boucle de traitement un nom d'événement. Dans notre exemple, le thème de départ est *"islamist"*, le processus prétopologique de l'algorithme PSA attache à ce mot les *tweets* qui sont sémantiquement proches de ce sujet et affiche par la suite, par ordre décroissant, les thèmes collectés autour de ce sujet. Comme le modèle cognitif de compréhension des textes, l'algorithme PSA construit un réseau de thèmes interconnectés, ce qui permet à la fonction prétopologique implémentée dans ce dernier de rattacher des *tweets* qui n'ont pas une relation significative à première vue. Le thème de départ *"islamist"* a collecté des *tweets* qui appartiennent à la révolution égyptienne et qui contiennent le mot *"crisis"* comme il a été plusieurs fois utilisé pour relier l'information sur la révolution égyptienne, il se retrouve comme un thème à part entière qui, entre à son tour dans le processus cyclique et rattache d'autres thèmes au réseau complexe construit, comme *job, family, judge*.

Les résultats de la deuxième phase de l'algorithme PSA sont représentés dans la Table 1. Le premier tableau admet les thèmes associés aux trois sous-ensembles sémantiques relatifs aux trois sujets suivants : *mubarak, egypt, revolution*. Le deuxième tableau représente quant à lui les thèmes associés, dans un ordre décroissant, aux trois sous-ensembles sémantiques de trois autres sous-ensembles relatifs aux sujets suivants : *revolution islamist radical*.

Dans la Table 1, notre thème candidat est *jfk*, mis en bleu dans les deux tableaux. Dans la recherche sémantique associant les six thèmes, le mot *jfk* apparaît avec la même proximité sémantique, 0.59 dans les 2 tableaux. Nous créeons par la suite un nouveau sous-ensemble sémantique où nous rassemblerons tous les *tweets* correspondant à ce thème, comme par exemple ce *tweet* : "Those who make peaceful revolution impossible, make violent revolution inevitable" - JFK #Egypt".

Les *tweets* collectés correspondant aux thèmes *jfk* ont été écrits entre le 1^{er} février et le 10 février. Nous avons fait la constatation qu'à la date du 22 mars 2011, donc bien après la collecte des *tweets* que nous avons dans notre corpus, que beaucoup d'articles dans la presse, partagés par la suite sur Twitter parlent de Wael Ghonim reçevant le prix annuel John F Kennedy pour son travail et son militantisme dans la transition démocratique de son pays. C'est à partir de cette constatation que nous concluons que le thème *jfk* est bel est bien un thème émergent.

B. Environnement d'exécution

L'exécution du *Parser*, de l'algorithme PSA ainsi que la construction de l'espace sémantique et de l'index ont été faites sur un Dell PowerEdge T710 fourni par le laboratoire LaISC, avec un processeur Intel Xeon X5672, 3.20GHz, 12M de mémoire cache, 40 Go de mémoire vive DD3 avec une fréquence de 1333MHz et ubuntu server 12.04 comme système d'exploitation.

V. Conclusion

Plus de 383 millions d'utilisateurs dans le monde se connectent à l'outil de *microblogging*, Twitter. De nombreux chercheurs se sont penchés sur la valeur prédictive des entités de texte diffusées par ce réseau social. Plusieurs articles ont ainsi affirmé qu'en analysant un grand nombre de *tweets*, il est possible de prévoir des faits ou des résultats de sondages dans le temps. Notre contribution se résume dans la proposition d'une idée novatrice qui se différencie des aux autres propositions qui sont pour la plupart des algorithmes d'analyse statistique[13]. Notre algorithme d'analyse prétopologique sémantique fait le couplage entre les fonctions prétopologiques et la construction des espaces sémantiques qui représentent pour nous une base pour l'apprentissage artificiel. Notre algorithme PSA, ayant prouvé son efficacité dans la détection des thèmes et des sujets émergents, nous projetons dans nos travaux futur de faire tourner le processus cyclique sur un plus large corpus de données dynamiques, qui seront constamment mises à jour. Le travail de l'algorithme PSA sera distribué avec l'utilisation de l'outil *Hadoop*. Nous projetons également de construire une interface graphique qui facilitera l'exploitation de l'algorithme et la visualisation des résultats.

Références

[1] C. D. Manning, P. Raghavan, and H. Schütze, "An introduction to information retrieval," *Online edition Cambridge*, 2009.

[2] T. Berners-Lee, "Weaving a semantic web," *Carvin Conference*, 2004.

[3] G. Denhière, B. Lemaire, C. Bellissens, and S. Jhean-Larose, "Psychologie cognitive et compréhension de texte : Une démarche théorique et expérimentale," 2004.

[4] D. M. Blei, "Probabilistic topic models," *communication of ACM, Vol.55(4)*, 2012.

[5] A. Choudhary, W. Hendrix, K. Lee, D. Palsetia, and W.-K. Liao, "Social media evolution of the egyptian revolution," *in CommunicationS of ACM, Vol.55(2)*, 2012.

[6] Tumasjan, A. Sprenger, and Sander, "Predicting elections with twitter : What 140 characters reveal about political sentiment," 2012.

[7] J. Bollen, H. Map, and X.-J. Zeng, "Twitter mood predicts the stock market," 2010.

[8] E. Bingham and H. Mannila, "Random projection in dimensionality reduction : Applications to image and text data," *Intl Conference KDD'01 San Francisco USA ACM*, 2001.

[9] Z. Belmandt, "Basics of pretopology," *Hermann, ISBN : 978 27056 8077, Editeurs : Marc Bui, Michel Lamure*, 2011.

[10] "Prétopologie et applications." *Numéro spécial de la revue Studia Informatica Universalis, Volume 7, Numéro 1*, pp. Marc Bui, Ivan Lavallée., 2009, 222 pages, Hermann, ISBN 978-2-7056-6894-5.

[11] M. Sahlgren, "An introduction to random indexing," *SICS, Swedish Institute of Computer Science, www.sics.se*.

[12] T. K. Landauer and S. T. Dumais, "A solution to plato's problem : The latent semantic analysis theory of acquisition, induction, and representation of knowledge," *Psychologica Review, Vol. 104(2).*, pp. 211–240, 1997.

[13] N. Evangelopoulos and L. Visinescu, "Statistical techniques help public leaders turn text in unstructured citizen feedback into responsive e-democracy." *Communications of the ACM, Vol.55(2)*, 2012.