An Efficient Semantic Graph-Based Approach for Text Representation

Ahmed Mabrouk[1] Rim Hantach[1] Philippe Calvez[1]
(1) ENGIE LAB CRIGEN , 361 Avenue du Président Wilson, 93210 Saint-Denis, France
ahmed.mabrouk@engie.com, rim.hantach@external.engie.com,
philippe.calvez1@engie.com

RÉSUMÉ

La représentation des documents est l'une des principaux problèmes dans le domaine de l'analyse des textes tels que l'extraction des thèmes et la similarité entre les textes. L'approche standard comme la représentation par sac de mots ne permet pas de représenter les liens sémantiques entre les termes. Afin de surmonter cette limitation, nous introduisons une nouvelle approche basée sur l'utilisation conjointe du graphe de co-occurrence et d'un réseau sémantique de la langue anglaise appelé *Wordnet*. Pour ce faire, un algorithme de désambiguïsation du sens des mots a été utilisé dans le but d'établir les liens sémantiques entre les termes étant donné le contexte sous-jacent. Les expérimentations réalisées sur des bases de données standards prouvent une bonne performance de l'approche proposée.

ABSTRACT

Text document representation is one of the main issue in the text analysis areas such as topic extraction and text similarities. Standard Bag-of-Word representation does not deal with relationships between words. In order to overcome this limitation, we introduce a new approach based on the joint use of co-occurrence graph and semantic network of English language called *Wordnet*. To do this, a word sense disambiguation algorithm has been used in order to establish semantic links between terms given the surrounding context. Experimentations on standard datasets show good performances of the proposed approach.

MOTS-CLÉS : Représentation des textes, *WordNet*, graphe, désambiguïsation des mots, sémantique.

KEYWORDS: Text representations, *WordNet*, graph, words sense disambiguation, semantic.

1 Introduction

Nowadays, textual information has seen a considerable progression. More and more textual data increase continuously over time. It has been emerged in many real-world application fields such as medicine (Hughes *et al.*, 2017), security (Suh-Lee *et al.*, 2016), meteorology (Ramos-Soto *et al.*, 2015), etc. Big amount of text creates a great need for efficient tools dedicated to manage and process these data. In this context, automated text analysis and natural language processing seem well-suited for analyzing textual data and identifying interesting information in a wide variety of applications. Numerous efforts have been devoted to propose tailored text analysis algorithms, e.g., topics extraction, text analysis, etc. It should be emphasized that successful text analysis is highly dependent on the way by which text corpora is represented. Bag-of-words (**BOW**) representation (Salton *et al.*, 1975) is a traditional formalism for representing textual information that describes the occurrence of terms within a document. Such representation ivolves two things : a vocabulary of known words (generally the most important terms) and a measure of the presence these latter. This approach, as shown in

many works, is doomed to be ineffective and has revealed a number of unexpected problems and weaknesses related to the absence of relationships between considered words. This problem thus induces significant issues, both from semantic representation and text analysis points of view. Note that relationships between words, as proved in (Hirst, 1988), is of great explanatory importance, since it allows to reveal the right meaning of each word in the text, facilitating therefore the task of texts analysis. To cope with this problem, a graph-based text representation (Wang *et al.*, 2011; Jin & Srihari, 2007; Zhou *et al.*, 2010; Rousseau & Vazirgiannis, 2013) has been proposed as an alternative to overcome the limitations of **BOW** method. These latter have been mainly investigated as a way of taking into consideration term dependencies and term orders. Co-occurrence network, has been emerged as one of the most successful formalism for text representation and has been applied in many real-world applications. Unlike **BOW** model, such model provides a prominent framework that enables to represent relations between words. Basically, a document is represented as a graph where vertices represent terms and edges represent co-occurrence relationships.

Many variants of the basic co-occurrence representation have been proposed in the literature. For example, in (Sihag & Kumar, 2013), a co-occurrence network has been used in order to estimate initial centroids parameters for the *K-means* algorithm. In this work, communities in the graph are detected throughout a set of edge deletion operations. Given the resulting groups of vertices, a centrality value is computed for all nodes within each community in order to select the optimal values and that, next, use them as initial centroids parameters for the *K-means* algorithm. Unfortunately, applying co-occurrence network is a very unsatisfactory solution to compute similarities between documents, since we do not take into account semantic relationships between words. In (Hossain & Angryk, 2007), authors propose to use a lexical base knowledge called *WordNet* (Miller, 1995) to build, at a first time, a document-graphs and then use it to perform clustering and text analysis operations. In this context, *WordNet* is used to provide relationships *hypernym-hyponym* between concepts, hence allowing to construct a concept tree graph with up to 18 levels for each term. While this method is quite efficient, it does not work in many cases because this tends to increase so much the size of the graph and leads to the problem of word sense ambiguity. In (Zhang *et al.*, 2011) authors propose a graph-based text similarity measurement. In this approach, semantic relations between words and documents have been established by using *Wikipedia Background Knowledge*. Co-occurrence network is substituted by document bipartie graph where nodes represent documents and their concepts. For a given document, the list of keywords are extracted and then mapped into the background knowledge in order to extract their respective concepts. In this approach, an edge connecting a document node with a concept one *iff* at least one word being part of this latter appears in the specific document. Similarity between two documents increases as much as their corresponding concepts are similar. However, such an approach is doomed to be ineffective because the information such as categories and semantic relationships between concepts bring valuable information that is not exploited during the graph-construction phase, thereby severely limiting its effectiveness. Agirre et al (Agirre *et al.*, 2009), propose a more sophisticated graph-based method to perform word sense disambiguation (**WSD**). At a first step, a graph representing all possible senses of each polysemous word in the text is built using *WordNet*. Once the graph is built, it can be used as a powerful tool to compute the importance of each interpretation in the graph. Here, the importance is computed throughout several sets of graph metrics, among them we can mention : In-degree Centrality, Eigenvector Centrality, etc. These centrality indicators, as it is well-known, are measures that tell us how influencing a node is within a graph. However, as stated in (Srivastava *et al.*, 2014), the reliability of centrality measures is very sensitive on the graph topology. As such, only a subset of centrality measures are well-suited for such and such network topology. Therefore, a more generic indicator is needed in this case. Authors in (Goikoetxea *et al.*, 2016) propose a new approach that learns a word representation from text

and *WordNet* separately, and then merge the obtained models using a set of several combination methods such as simple vector combination, correlation analysis, etc. Agirre et al (Agirre & Soroa, 2009) addressed also the **WSD** problem using a graph-based approach by applying PageRank and Personalized PageRank algorithms on the semantic graph. PageRank algorithm has also been used in (Šajgalík *et al.*, 2013) in order to extract the relevance of latent concepts hidden in text by observing words. In this work, several factors like collocations of words, *hypernym* and *holonym* relations and information content of concepts has been taken into account. These approaches while quite efficient, they are not always tractable in practice, especially on large network because it is very time consuming and cannot be performed on real time. In (Khazaal & Kamaruddin, 2015), authors proposed a graph-based approach to perform documents clustering. After a pre-processing phase, sentence structures in the corpus are identified, and then represented by dependency graphs. After that, the produced graphs are used to perform cluster analysis.

In this paper, we focus our discussion on semantic graph construction. Our approach first constructs the co-occurrence network of the textual documents, and, then uses a lexical knowledge base (**LKB**) to refine the overall relationships encoded in the network. The originality of our algorithm lies in the rules applied in the second step of our approach, that exploit both **LKB** and co-occurrence graph to efficiently infer the right semantic of each word. The rest of the paper is organized as follows. Section 2 recalls the definitions and notations needed to explain our approach. Then, in Section 3, we describe our approach and justify its correctness. Its effectiveness is highlighted through experiments in Section 4. Finally, we terminate by a conclusion.

2 Basics on Graph-based Documents Representation

In this paper, a capital letter V represents a single node in $\mathcal{G}$, a boldface capital letter $\mathbf{Z}-$a set of nodes. The numbers n will always be used to denote the numbers of terms, i.e., $\mathcal{X} = \{V_1, V_2, ..., V_n\}$. Adjacent nodes to V_i in $\mathcal{G}$ are denoted by $\mathcal{N}_{\mathcal{G}}(V_i)$. Our goal is to construct a semantic graph representing textual data from a given textual corpus $\mathcal{T}$.

Definition 1 (Graph) *A graph $\mathcal{G}$ is a pair of $(\mathcal{X}, \mathcal{E})$ where $\mathcal{X}$ represents a set of nodes, and $\mathcal{E} \subseteq \{\mathcal{X} \times \mathcal{X}\}$ corresponds to a set of edges representing relationships between nodes.*

It should be reminded that in the graph-based formalism, relationships encoded in $\mathcal{G}$ can have several semantics, among them we can mention : (i) co-occurrence of terms in the text or a part of this latter (ii) semantic relation : *synonyms*, *antonyms*, *hypernyms*, etc. (iii) terms that often share common words in a slided windows. In the basic version (co-occurrence network), we connect two terms V_i and V_j in the network $\mathcal{G}$ *iff* they appear together at least one time in the same window of a predetermined fixed size. Applying such technique may arise the problem of adding ineffective edges and nodes in the network, leading consequently to the construction of large-scale netowrks. To alleviate this issue, an approach called *K-core* decomposition has been proposed (Sariyuce *et al.*, 2016). This method is particularly attractive due to the simplicity of its definition and the fact that it can be computed in linear time, even for dense graphs. Proposed by (Sariyuce *et al.*, 2016), this method aims to generate from the initial graph $\mathcal{G}$ the most cohesive sub-graph $\mathcal{G}_k$, where its vertices are intuitively suitable candidates for representing the whole co-occurrence graph with a minimum loss of precision. The *K-core* decomposition algorithm consists in identifying particular subsets of the graph (called *K-cores*) by recursively removing all nodes of degree smaller than K, until the degree of all remaining nodes is larger or equal than to K. For more details, readers should refere to (Sariyuce *et al.*, 2016).

Graph-based approaches are getting increasing attention from the **LKB** community. This can be explained by the fact that classical co-occurrence approach is statistical, as nodes are linked based on

local context of co-occurrence only, regardless of any semantic information. These new generation of graphs use a well-known graph-based technique to find and exploit structural properties of the graph underlying a particular **LKB**. Basically, the **LKB** can be defined as a body of background knowledge, mostly of a semantic nature, hierarchically structured and electronically available. *WordNet* is one of the most popular **LKB** that are particularly suitable for NLP and text analysis applications. It is represented in the form of a network containing English words which are grouped into 117.000 synonyms, connected between them by means of semantic and lexical relations (e.g., *hypernym*, *hyponym*, etc.). As they intend to provide particular semantic relations between terms, they can definitely benefit the development of NLP systems and, in particular, when used jointly with co-occurrence graph. However, it may be pointed out that the exclusive use of *WordNet* inventory to add semantic relations between nodes in $\mathcal{G}$, as we shall see hereafter, is not sufficient because it may lead to another kind of problem called *Word Sense Ambiguity*.
To avoid ambiguities between synsets and terms, in the rest of this paper, letters, when superscripted by a hat, e.g., $\hat{\mathbf{V}} = \bigcup_{i=1}^{h} \hat{V}_i$ (note that h value changes from one term to another), represent the synsets associated to the term V in the *WordNet* inventory. Merging the co-occurrence and *WordNet* graphs will result in a semantic graph composed by a mixture of word and synset nodes.

Definition 2 (Semantic graph) *A semantic graph $\mathcal{S}$ is a pair of ($\mathcal{X}^*$,$\mathcal{E}^*$) where $\mathcal{X}^* = \{\mathcal{X} \cup \hat{\mathcal{X}}\}$ and $\mathcal{E}^* \subseteq \{\{\mathcal{X} \times \mathcal{X}\} \cup \{\hat{\mathcal{X}} \times \mathcal{X}\} \cup \{\hat{\mathcal{X}} \times \hat{\mathcal{X}}\}\}$ represents relationships between (i) words (ii) words and synsets(iii) synsets.*

One of the main problems encountered when constructing a semantic graph is the sense ambiguity for several words, i.e., words often have more than one meaning, sometimes fairly similar and sometimes completely different. Such problem is addressed throughout the use of **WSD** algorithms. The right sense of a word in a particular usage can only be determined by examining its context, which can vary from a few set of words to the entire text. In the context of semantic graph $\mathcal{S}$, ambiguous words lead to the notion of ambiguous nodes, which are formally defined as follows :

Definition 3 (Ambiguous node) *Let $\mathcal{X}$ be a set of terms nodes and let $\hat{\mathcal{X}}$ be the set of synsets. A node $V_i \in \mathcal{X}$ is said to be ambiguous in $\mathcal{S}$ iff : $|N_{\mathcal{S}}(V_i) \cap \hat{\mathcal{X}}| > 1$*

In such a case, a WSD process is needed in order to identify the most "relevant" node among the set of adjacent synsets. As a concrete example, the following two sentences depict an example where the sense of the word "bank" is not the same : *"An erosion phenomena appears on the river bank"*, *"How do I withdraw money from my bank account if I don't have a MasterCard ?"* At a first sight, we recognize from the context of the first sentence that the word 'bank' means the edge of a river, whereas in the second one it corresponds to the financial institute. By giving the previous example, several interesting questions emerge. The most common questions are related to the way by which the textual corpus has to be efficiently structured into a graphical formalism in a way that enables to infer efficiently the right meaning of each word. As it turns out, the answer is generally not clear. In the next section, we present details of our approach for constructing a semantic graph from a textual corpus.

3 A New Semantic Graph-Based Construction

Like several other algorithms, ours is an hybrid that combine co-occurrence graph with an external lexical knowledge base (*WordNet*) to infer, at a first time, additional semantic relationships and nodes and that, next, refine the whole graph by assigning to each node its right sense.

3.1 First Phase : Building the Co-occurrence Graph

As said previously, word co-occurrence network is a widely used formalism in many domains. Basically, it aims to link terms occurring together in the same window. In the sequel, we will briefly introduce the process of co-occurrence network construction from collections of separate documents $\mathcal{D}$.

Before we transform the corpus into a graph, a set of preliminary clean-up and pre-processing operations must be done. Text obtained from web or any other source of information usually contains punctuations and special characters which get embedded in the original data. It is thus necessary to get rid of these entities before performing text network analysis. To do this, we used a set of specific regular expressions (to delete emails, urls, date, etc.) and dedicated packages especially NLTK (Loper & Bird, 2002) and NetworkX (Hagberg *et al.*, 2008). Next, a terms matrix (TM) is obtained by measuring the weight of each word throughout the use of *TF-IDF* metric. Note that the resulted TM is an $n \times m$ (where m is the number of sliced windows) asymmetric matrix. Given the TM matrix $\mathbf{X}$, a term correlations matrix (TCM) has been inferred throughout the use of *findAssocs* function which is implemented in the TM package (Feinerer & Hornik, 2017). Note that if two terms V_i and V_j appear together more frequently in the corpus, the strength (or correlation) between them tends to increase. We connected two words V_i and V_j in the graph *iff* the correlation between them is great or equal to a chosen threshold α. In our case α's value has been fixed to 0.6. Here, the parameter α is mainly used to disfavor low-degree correlations by simply setting a threshold on the correlation between words in $\mathcal{G}$, and then removing all those who are below than it. However, despite the deletion of weak dependencies, it stills often inessential words that have not been deleted during the previous phase. To tackle this issue, several approaches can be used. In our case, we advocate to use a variant of the *K-core* decomposition (Sariyuce *et al.*, 2016) in order to select the most relevant nodes based on their centralities in $\mathcal{G}$. Thus, vertices composing the most cohesive connected subgraph are intuitively the most salient keywords for representing the entire graph. After performing *K-core* decomposition on $\mathcal{G}$, we obtain a graph that capture the main structural information in the text. Until now, the used method is statistical, as terms are connected based only on local context of co-occurrence, regardless of any semantic relationships that may occur between them. In the second section, we will illustrate technical details about semantic graph construction based on *WordNet* and **WSD** algorithm.

3.2 Second Phase : Adding Semantic Relationships in $\mathcal{S}$

The next step of our graph-building approach consists of adding semantic relationships between nodes in order to get an initial semantic graph that will subsequently be refined. The second phase of our approach utilizes the build graph and *WordNet* 3.0 taxonomy to convert the simplified co-occurrence to its corresponding semantic graph. For each term V, the list of synsets (denoted by $\hat{\mathbf{V}}$) and their relationships are added to the graph by performing a graph traversal up to the topmost level of *WordNet*. By doing so, we obtain a graph composed with a subgraph of the entire *WordNet* lexicon and the *K-core* network (resulted from the co-occurrence network). After adding the semantic part, a disambiguation step that allows to determine for each polysemous node in $\mathcal{S}$ the right sens is needed. This procedure raises the following question : how can we guarantee the correctness of the word disambiguation process by exploiting valuable information coming respectively from *WordNet* and *K-core* networks ? The problem in such case is that for each word V_i in $\mathcal{S}$, the surrounding context words that serve to detect the right interpretation are often polysemous, which presents an important problem in our case. It should bo noted that the phase of **WSD** is interesting not only for determining he right synset for each node, but also for establishing semantic relationships between disconnected words in the initial *K-core* network, e.g., despite the fact that "bank" and "stock exchange" words are disconnected in *K-core* network, they are (directly or indirectly) semantically connected by referring

to *WordNet* and **WSD** algorithm. This is the reason why this task must be addressed with a very high accuracy, since a poor assessment of the sense can lead to a drastic change in the final graph.
In our case, rather than using the whole graph structure to perform **WSD**, we advocate to exploit a smallest set of nodes (hereafter denoted by Θ_i) to efficiently determines the sense of each polysemous word V_i in $\mathcal{S}$:

$$\Theta_i = \left\{ \bigcup_{V_j \in \mathcal{N}_{\mathcal{S}}(V_i)} \hat{\mathbf{V}}_j \setminus \{\mathcal{X}\} \right\} \tag{1}$$

The set of nodes used to make **WSD** of a given node $V_i \in \mathcal{X}$ can be easily identified from $\mathcal{S}$: it consists of the union of synsets of all and only adjacent co-occurrence words (which form the surrounding context of V_i). Consider the example of semantic graph given in Figure 1 where synsets and words are represented respectively with square and circle nodes .

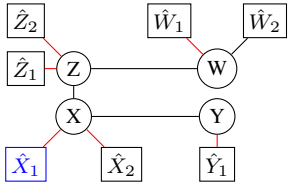

FIGURE 1 – A simple example of semantic graph $\mathcal{S}$

By referring to equation 1, it is easy to see that the set of nodes used to find the right definition of the word X are composed by $(\hat{Z}_1, \hat{Z}_2, \hat{Y}_1)$.
Local measure between each interpretation (or meaning) of $\hat{V}_{i_k}$ and Θ_i is used to determine the relevance of each one. Formally, we define a local measure $\mathcal{M}$ as : $\mathcal{M}$: $(\hat{V}_{i_k}, \Theta_i) \rightarrow [0, 1]$. Values close to 0 indicate that the sense is not important w.r.t V_i, whereas a value close to 1 suggests that the synset is very important. Similarity between a word-synset $\hat{V}_{i_k}$ and context Θ_i is calculated as follows :

$$\mathcal{M}(\hat{V}_{i_k}, \Theta_i) = \frac{1}{|\Theta_i|} \sum_{\hat{Z} \in \Theta_i} \frac{1}{\left[s_{PATH}(\hat{V}_i, \hat{Z}) + 1 \right]} \tag{2}$$

where $s_{PATH}(\hat{V}_i, \hat{Z})$ represents the path $\hat{V}_i - ... - \hat{Z}$ which is extracted from *WordNet*. The more similar the synsets, the shorter the path separating them. The intuition behind $\mathcal{M}$ is that a node synset is important for a given node V_i if it is semantically similar to the synsets surrounding the context words. Then, the best synset value w.r.t to V_i is obtained as follows :

$$\hat{V}_i^* = \underset{\hat{V}_{i_k} \in \hat{\mathbf{V}}_i}{\operatorname{argmax}} \mathcal{M}(\hat{V}_{i_k}, \Theta_i) \tag{3}$$

By using the local approach to handle polysemous nodes in $\mathcal{S}$, we significantly reduce the computational overhead incurred by global **WSD** algorithm (PageRank, Maximum Flow, etc.), which are applied on the whole graph. However, this method, like classical approaches, does not overcome the issues raised by the fact that all synsets [1] of every considered co-occurrence word are used in the **WSD** procedure. Another problem concerning the proposed is related to the order in which polysemous word are disambiguated. This order, as we shall see, can lead to a wide range of different results, especially in high dimensional spaces (graph containing many nodes). In the following, we will consider an extract of a semantic graph example to show the impact of word disambiguation order on the final result.

1. For example, if the polysemous word "operation" co-occurs together with the word "bank", then synsets of this latter such as "'sloping land", "bank building", "sloping land" are used to make the disambiguation process

Exemple 1 *Suppose we have a semantic graph $\mathcal{S}$ representing a simple textual corpus on a set of variables $\mathcal{X} = \{X, Y, Z, W\}$. In this example, the set of co-occurrence (blue edges) encoded by the graph are : $Z - X$, $Z - W$, and $X - Y$. Semantic relations (red edges) are the following : $Z - \hat{Z}_1$, $Z - \hat{Z}_2$, $X - \hat{X}_1$, $X - \hat{X}_2$, $W - \hat{W}_1$, $W - \hat{W}_2$, and $Y - \hat{Y}_1$.We assume that distance between synset (as given in WordNet) and nodes are as follows :*

$$\begin{array}{llll}
\hat{X}_1 \longrightarrow ... \longrightarrow \hat{Z}_1 & : s_{PATH}(\hat{X}_1, \hat{Z}_1) = 10 & \hat{X}_1 \longrightarrow ... \longrightarrow \hat{Z}_2 & : s_{PATH}(\hat{X}_1, \hat{Z}_2) = 20 \\
\hat{X}_2 \longrightarrow ... \longrightarrow \hat{Z}_1 & : s_{PATH}(\hat{X}_2, \hat{Z}_1) = 20 & \hat{X}_2 \longrightarrow ... \longrightarrow \hat{Z}_2 & : s_{PATH}(\hat{X}_2, \hat{Z}_2) = 2 \\
\hat{X}_1 \longrightarrow ... \longrightarrow \hat{Y}_1 & : s_{PATH}(\hat{X}_1, \hat{Y}_1) = 2 & \hat{X}_2 \longrightarrow ... \longrightarrow \hat{Y}_1 & : s_{PATH}(\hat{X}_2, \hat{Y}_1) = 60 \\
\hat{Z}_1 \longrightarrow ... \longrightarrow \hat{W}_1 & : s_{PATH}(\hat{Z}_1, \hat{W}_1) = 2 & \hat{Z}_1 \longrightarrow ... \longrightarrow \hat{W}_2 & : s_{PATH}(\hat{Z}_1, \hat{W}_2) = 40 \\
\hat{Z}_2 \longrightarrow ... \longrightarrow \hat{W}_1 & : s_{PATH}(\hat{Z}_2, \hat{W}_1) = 6 & \hat{Z}_2 \longrightarrow ... \longrightarrow \hat{W}_2 & : s_{PATH}(\hat{Z}_2, \hat{W}_2) = 10
\end{array}$$

Let us now explain how the disambiguation procedure is performed given two different orders : $\prec_1 = (X \prec Z \prec W)$ and $\prec_2 = (Z \prec X \prec W)$. Let us start with $\prec_1$. When we start by disambiguating X, the similarity equations lead to the following results : $\mathcal{M}(X, \hat{X}_1) = 0.157$ and $\mathcal{M}(X, \hat{X}_2) = 0.132$. From these results, we can deduce that $\hat{X}_2$ is more similar to X than $\hat{X}_1$, hence the edge $X - \hat{X}_1$ is deleted from $\mathcal{S}$. At a second stage, the following results $\mathcal{M}(Z, \hat{Z}_1) = 0.224$ and $\mathcal{M}(Z, \hat{Z}_2) = 0.016$ lead to the deletion of the relationship $Z - \hat{Z}_2$. At the end, the disambiguation of node W leads to the deletion of edge $W - \hat{W}_2$. For $\prec_2$, the disambiguation process is performed as follows : when we consider the node Z, the similarities measure, unlike $\prec_1$, leads to the deletion of edge $Z - Z_1$ since $\mathcal{M}(Z, \hat{Z}_1)$ and $\mathcal{M}(Z, \hat{Z}_2)$ are respectively equal to 0.124 and 0.153.

As can be seen in Example 1, at each search step, the prior $\prec_i$ determines the order in which disambiguation nodes are considered when computing the similarities between each sysnet (of the polysemous word) and the surrounding context. For each edge removal, the semantic graph $\mathcal{S}$ is updated, hence, the adjacency synsets of the concerned nodes typically changes at each iteration. Given the iteration $t = 0$, when the adjacency synsets of some nodes is updated, the other semantic similarities that have to be checked at iteration $t = t + h$ are also affected (which is the case for node Z in example 1). Since the optimal order is not necessarily known a priori, term order dependence becomes very problematic in our case, especially for high dimensional semantic graph since they can lead to highly unstable results, i.e., they can produce graphs varying drastically from one order to the other. To alleviate this problem, the following equation has been proposed in order to be used at the beginning in order to estimate an efficient ranking over words before performing **WSD** procedure. For a given variable V_i, the ranking score of this latter is computed as follows :

$$\prec_{V_i} = \alpha \left[\frac{1}{|\hat{\mathbf{V}}_i|} \times \frac{1}{|\Theta_i|} \sum_{V_j \in \mathcal{N}_{\mathcal{S}}(V_i)} |\hat{\mathbf{V}}_j| \right] + \beta \left[|\mathcal{N}_{\mathcal{S}}(V_i) \setminus \{\hat{\mathcal{X}}\}| \right] \tag{4}$$

This formula has been inspired from the *Occam's Razor* principle. As can be seen, this function is divided into two parts : the first term, which represents the complexity of sense identification, leads to the penalization of complex polysemous words, i.e., the case where there is many number of synsets for both word and its surrounding context. The second term tends to promote word that has a very well expressed context (with many adjacent co-occurrence words for a given V_i). α and β are two parameters whose values are between 1 and 0 and $\alpha + \beta = 1$. Parameter values, which are fixed by the user, allow to assign an importance to each part of equation 4. In our case a value of 0.5 has been assigned to both parameters.

4 Experimentations

In this section, we highlight the effectiveness of our method by comparing it with term-concept method (ajgalík *et al.*, 2013) and *TF-IDF* method. For this purpose, we used the 20 newsgroups dataset (each containing 1000 documents) (Lang, 2007) which is a popular dataset of experiments in text applications of machine learning algorithms. From this database, a graph has been constructed from each document by converting its contained text into a semantic graph (as mentioned in section 3) and text matrix representations using the state-of-the-art approaches. To perform the comparison between the considered approaches, the efficiency of the resulted semantic graphs have been evaluated by considering the task of documents clustering procedure [2]. A distance matrix between the set of generated graphs (representing documents) have been computed with *Hamming Distance*, and then used as an input for the *k-means* clustering algorithm. The resulted clusters were compared against those of the true groups using four metrics : Precision, Recall, F-score and Accuracy per resulted group. Table 1 displays the averages of the mentioned metrics for each group (or cluster). As can be seen, except in some cases (clusters 1, 4, 6, 12 and 17), especially when the cluster are very correlated to each other's, our algorithm obtain high classification results (accuracy between 0.75 and 1). The performance of our algorithm follows mainly from its rules dedicated to build the overall graph and especially those dedicated to detect the right interpretation of each polysemous word. This means that despite the fact that two similar documents use different words, our semantic graph allows to establish the similarity between them throughout the detection of common subgraphs (using *WordNet* and the proposed **WSD** algorithm [3]). A very poor result has been obtained for documents related to "religion.misc" topic. This misclassified result can be explained by the fact that "religion.misc" cluster is very similar to other topics such as "religion.christian" and "atheism", hence the *K-means* algorithm does not succeed to efficiently separate them. In table 2, we compare the performance of our approach

Cluster	precision	recall	f-score	accuracy
hardware	0.50	0.38	0.44	0.75
med	0.33	0.08	0.16	0.25
atheism	1.00	1.00	1.00	1.00
politics.mideast	0.50	0.38	0.44	0.75
graphics	0.33	0.16	0.25	0.50
baseball	1.00	1.00	1.00	1.00
religion.misc	0.00	0.00	0.00	0.00
politics.guns	1.00	1.00	1.00	1.00
ibm.pc.hardware	1.00	1.00	1.00	1.00
os.ms.windows	0.50	0.38	0.45	0.75
forsale	0.50	0.38	0.44	0.75
crypt	1.00	1.00	1.00	1.00
politic.misc	0.50	0.12	0.24	0.25
motorcycles	1.00	1.00	1.00	1.00
space	1.00	1.00	1.00	1.00
religion.christian	1.00	1.00	1.00	1.00
electronics	1.00	1.00	1.00	1.00
windows.x	0.33	0.083	0.16	0.25
autos	1.00	1.00	1.00	1.00
hokey	1.00	1.00	1.00	1.00

TABLE 1 – Clustering results using 20 newsgroups

(on the overall dataset) with Key-concept (ajgalík *et al.*, 2013) (standard methods namely Top 10 key-concepts with Naive Bayes, TOP 20 weighted key-concepts with KNN) and *TF-IDF* based methods (Weighted *TF-IDF* vector with KNN and *TF-IDF* vector with Naive Bayes). As can be observed, our proposed method significantly outperforms the others state-of-the-art algorithms. The accuracy of our algorithm is actually about 34.7%, 37.51% 39.3% and 48.7% respectively higher

2. Note that the main purpose of our proposed algorithm consists in giving a correct semantic graph representation that may be applied within several tasks such as documents clustering, topics extraction, etc.

3. It is a part of the semantic graph building

Methods	Accuracy of classification
Top 10 key-concepts with Naive Bayes	41.48
TOP 20 weighted key-concepts with KNN	38.74
Weighted *TF-IDF* vector with KNN	36.95
TF-IDF vector with Naive Bayes	27.55
Our approach	**76.25**

TABLE 2 – Comparison of classification accuracy methods

than the mentioned methods. These results highlight again the effectiveness of our proposed method, notably those resulting from the pre-computed terms order [4].

5 Conclusion

In this paper, we have proposed a new approach for document representation which is based on the combination of co-occurrence and *WordNet* graphs. The problem of **WSD** has been addressed in the construction phase of the semantic graph by considering a local set of synsets and a pre-computed order scores assigned to each node to disambiguate. As shown in the experimentations, our algorithm outperforms significantly state-of-the-art algorithms. For future work, we plan to extend our algorithm for *n-grams* graph based text representation. In addition, more experimentations concerning the pre-computed order impacts will be studied.

Références

AGIRRE E., LACALLE O. L. D. & SOROA A. (2009). Knowledge-based wsd on specific domains : Performing better than generic supervised wsd. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, p. 1501–1506, San Francisco, CA, USA : Morgan Kaufmann Publishers Inc.

AGIRRE E. & SOROA A. (2009). Personalizing pagerank for word sense disambiguation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, p. 33–41, Stroudsburg, PA, USA : Association for Computational Linguistics.

ŠAJGALÍK M., BARLA M. & BIELIKOVÁ M. (2013). From ambiguous words to key-concept extraction. In *2013 24th International Workshop on Database and Expert Systems Applications*, p. 63–67.

AJGALÍK M., BARLA M. & BIELIKOVÁ M. (2013). From ambiguous words to key-concept extraction. In *2013 24th International Workshop on Database and Expert Systems Applications*, p. 63–67.

FEINERER I. & HORNIK K. (2017). *tm : Text Mining Package*. R package version 0.7-3.

GOIKOETXEA J., AGIRRE E. & SOROA A. (2016). Single or multiple ? combining word representations independently learned from text and wordnet. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, p. 2608–2614 : AAAI Press.

HAGBERG A. A., SCHULT D. A. & SWART P. J. (2008). Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference*, p. 11 – 15, Pasadena, CA USA.

HIRST G. (1988). Semantic interpretation and ambiguity. *Artificial Intelligence*, **34**(2), 131 – 177.

HOSSAIN M. S. & ANGRYK R. A. (2007). Gdclust : A graph-based document clustering technique. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, p. 417–422.

4. Terms who are concerned by the **WSD** procedure

HUGHES M., LI I., KOTOULAS S. & SUZUMURA T. (2017). Medical text classification using convolutional neural networks. *Studies in health technology and informatics*, **235**, 246–250.

JIN W. & SRIHARI R. K. (2007). Graph-based text representation and knowledge discovery. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, SAC '07, p. 807–811, New York, NY, USA : ACM.

KHAZAAL A. & KAMARUDDIN S. (2015). Graph based text representation for document clustering. **76**.

LANG K. (2007). 20 newsgroups data set.

LOPER E. & BIRD S. (2002). Nltk : The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, p. 63–70, Stroudsburg, PA, USA : Association for Computational Linguistics.

MILLER G. A. (1995). Wordnet : A lexical database for english. *COMMUNICATIONS OF THE ACM*, **38**, 39–41.

RAMOS-SOTO A., DIZ A. J. B., BARRO S. & TABOADA J. (2015). Linguistic descriptions for automatic generation of textual short-term weather forecasts on real prediction data. *IEEE Trans. Fuzzy Systems*, **23**(1), 44–57.

ROUSSEAU F. & VAZIRGIANNIS M. (2013). Graph-of-word and tw-idf : new approach to ad hoc ir. In *CIKM*.

SALTON G. M., WONG A. & YANG C. (1975). A vector space model for automatic indexing. *Commun. ACM*, **18**(11), 613–620.

SARIYUCE A. E., GEDIK B., JACQUES-SILVA G., WU K.-L. & ÇATALYUREK U. V. (2016). Incremental k-core decomposition : Algorithms and evaluation. *The VLDB Journal*, **25**(3), 425–447.

SIHAG V. K. & KUMAR S. (2013). Graph based text document clustering by detecting initial centroids for k-means. *International Journal of Computer Applications*, **62**(19), 1–4.

SRIVASTAVA A., SRIVASTAVA M., GARG R. & MISHRA P. (2014). Comparative study of web page ranking algorithms. **1**, 26–32.

SUH-LEE C., JO J.-Y. & KIM Y. (2016). Text mining for security threat detection discovering hidden information in unstructured log messages. In *2016 IEEE Conference on Communications and Network Security, CNS 2016, Philadelphia, PA, USA, October 17-19, 2016*, p. 252–260, Philadelphia, PA, USA : IEEE.

WANG Y., NI X., SUN J.-T., TONG Y. & CHEN Z. (2011). Representing document as dependency graph for document clustering. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, p. 2177–2180, New York, NY, USA : ACM.

ZHANG L., LI C. & LIU J. (2011). Graph-based text similarity measurement by exploiting wikipedia as background knowledge. volume 5.

ZHOU F., ZHANG F. & YANG B. (2010). Graph-based text representation model and its realization. In *Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering(NLPKE-2010)*, p. 1–8.