IADIS International Journal on Computer Science and Information Systems
Vol. 13, No. 2, pp. 76-92
ISSN: 1646-3692

AUTOMATIC GENERATION OF ONTOLOGIES: A HIERARCHICAL WORD CLUSTERING APPROACH

Smail Sellah[1,2] and Vincent Hilaire[1]
[1]*LE2I FRE 2005, Univ. Bourgogne Franche-Comté, UTBM, F-90010 Belfort, France*
[2]*HTI automobile groupe APSIDE 10, avenue Léon Blum, 25200 Montbéliard, France*

ABSTRACT

In the context of globalization, companies need to capitalize on their knowledge. The knowledge of a company is present in two forms tacit and explicit. Explicit knowledge represents all formalized information i.e all documents (pdf, words ...). Tacit knowledge is present in documents and mind of employees, this kind of knowledge is not formalized, it needs a reasoning process to discover it. The approach proposed focus on extracting tacit knowledge from textual documents. In this paper, we propose hierarchical word clustering as an improvement of word clusters generated in previous work, we also proposed an approach to extract relevant bigrams and trigrams. We use Reuters-21578 corpus to validate our approach. Our global work aims to ease the automatic building of ontologies.

KEYWORDS

Knowledge Management, Ontologies, Word Clustering, Experience Feedback

1. INTRODUCTION

Knowledge management is an important issue in companies and especially nowadays with the context of big data. Indeed, companies are experiencing a sharp increase of data size. IDC forecasts, in their study "data age 2025", that Datasphere size will reach in 2025 163 zetabites and 60% of this data will be generated by companies (Reinse et al. 2017). This amount of data constitutes a knowledge, which companies need to capitalize on it to innovate. Leveraging this data is a challenging task.

Traditional methods of information retrieval do not take semantics into account, they only return documents containing a given set of keywords. The produced results contain documents that are frequently not relevant and miss some important informations. Our framework focus on how to cluster documents by leveraging word semantics deduced from the documents content

in order to ease researches. Our work consists in two steps. The first step is to identify a structure defining semantic relationships for the whole set of documents. The second step is to build a document representation based on the structure resulting from the first step.

Many structures model semantic relationship between words. WordNet is a well-known lexical database, it describes concepts and relation among them. Thesaurus are another kind of representation, words are linked based on predefined relationships. Ontology is a set of concepts, these concepts are organized in form of network, where concepts are linked if there is semantic relationship between them and these links are labeled with type of semantics. Folksonomies are collaborative structures, they are built through collaborative document annotation processes, this kind of structure is typically used to organize and search documents.

In document clustering, documents are generally represented by a set of weighted words. The weight of a word represents its importance with respect to a given document. Words can be weighted using TF-IDF (term frequency-inverse document frequency) measure. However, this approach misses some important information such as word similarity. Indeed, two documents describing the same idea, but using synonyms, may not be considered as similar. In order to overcome this kind of drawback, we propose the use of an ontology because concepts may reference a whole set of words through their meanings. Thus, even if synonyms are used, the similarity between two documents can be recognized, (Punitha et al. 2012, Hotho et al. 2003) showed that using an ontology improves document clustering.

Ontologies are mostly handcrafted, and usually do not represent the needs and knowledge of companies. Their engineering is thus costly (resources and time). Our goal consists in defining a structure close to an ontology. The general idea is to detect the semantic distance between words, then cluster them into clusters of semantically close words. We believe a cluster, composed with semantically close words, will capture a generic meaning which can be considered as a concept. Using those word clusters, we cluster documents.

In this paper, we work on the first step, we propose a hierarchical word clusters as an alternative to flat word clusters, we believe this hierarchical representation of word clusters will help us to detect more clusters and semantic relationships between clusters. In our previous work (Sellah and Hilaire), we did not considered bigrams (New York, Hong Kong ...), in this paper, we propose an approach to extract relevant bigrams, these bigrams may capture some semantic relationships.

The reminder of this paper is organized as follow. In section 2, we will present some background. In section 3, we introduce an overview of our framework. In section 4 we discuss our experimental results. The section 5 is dedicated to related works presentation and in section 6, we conclude.

2. BACKGROUND

In this section, we will present some background used in this work, we will introduce some semantic distances, clustering algorithms and clustering quality measures.

2.1 Word Similarity

Documents are composed of words organized under the form of sentences grouped in paragraphs. Words are the atomic level, which we can exploit to achieve document clustering.

The objective of word similarity is to define a semantic distance between two words, it can be web-based or corpus-based. Web-based word similarity uses the number pages return by a search engine, for each word separately and together, to define a semantic distance. Corpus-based word similarity uses the occurrence of words and their co-occurrence from a corpus to define a semantic distance. In the following, we introduce some popular measures. Where D(w) is document frequency of word w, D(w1,w2) is document frequency where words w1 and w2 co-occurred.

2.1.1 Corpus-based

Dice and Jaccard use co-occurrence and occurrence of each word to define similarity distance (Spanakis et al. 2009, Iosif and Potamianos 2010,Bollegala et al. 2011, Mei et al. 2015).

$$\text{Jaccard}(\omega_1, \omega_2) = \frac{D(\omega_1, \omega_2)}{D(\omega_1) + D(\omega_2) - D(\omega_1, \omega_2)} \quad (1)$$

$$\text{Dice}(\omega_1, \omega_2) = \frac{2D(\omega_1, \omega_2)}{D(\omega_1) + D(\omega_2)} \quad (2)$$

PMI (Pointwise Mutual Information) uses probabilistic occurrence of words and their probabilistic co-occurrence (Terra and Clarke 2003, Spanakis et al. 2009,). N is the number of documents in corpus.

$$\text{PMI}(\omega_1, \omega_2) = \log_2\left(\frac{\frac{D(\omega_1, \omega_2)}{N}}{\frac{D(\omega_1)}{N}\frac{D(\omega_2)}{N}}\right) \quad (3)$$

2.1.2 Web-based

NGD (Normalized Google Distance) (Cilibrasi and Vitáni 2007) uses Google search engine to define a semantic distance between words. N is the number web pages indexed by the Google search engine which is about 10^10. Where f(x,y) is the number of pages returned by Google engine for both words, f(x) is the number of pages returned of the word x.

$$\begin{aligned} \text{NGD}(x, y) &= \frac{G(x, y) - \min(G(x), G(y))}{\max(G(x), G(y))} \\ &= \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log N - \min\{\log f(x), \log f(y)\}}, \end{aligned} \quad (4)$$

Knowledge based semantic distances are an alternative approach to the statistical ones. Many measures based on WordNet were defined, Meng et al, (2013) propose a review. Also Thesaurus based semantics distances exist, Jarmasz and Szpakowicz (2003) define a semantic similarity measure based on Roget's Thesaurus.

2.2 Clustering

Clustering algorithms consist in organizing a set of objects into clusters, where objects belonging to a same cluster have a high level of similarity. Conversely, objects belonging to different clusters have a low level of similarity. We use the clustering approach to group words and document, we can organize clustering algorithms into graph-based algorithms and vector-based algorithms.

2.2.1 Graph-based clustering

Within the graphs based approach, clusters, also known as communities, have the following property: objects (nodes) of a cluster are highly connected and have few connexions to nodes from other clusters. Many approaches exist in community detection in a graph, here, we introduce some popular approaches.Markov Clustering algorithm (MCL) is an unsupervised graph clustering based on stochastic simulation (Dongen 2000). MCL is characterised by its simplicity and scalability. It uses two operators called expansion and inflation. Where expansion is the normal matrix product and inflation is the Hadamard power of a matrix followed by a normalization step. By varying the Hadamard power R, it alters the number of clusters. MCL combines these two operators repeatedly until there is no change in the resulting matrix. The MCL algorithm is as follows:

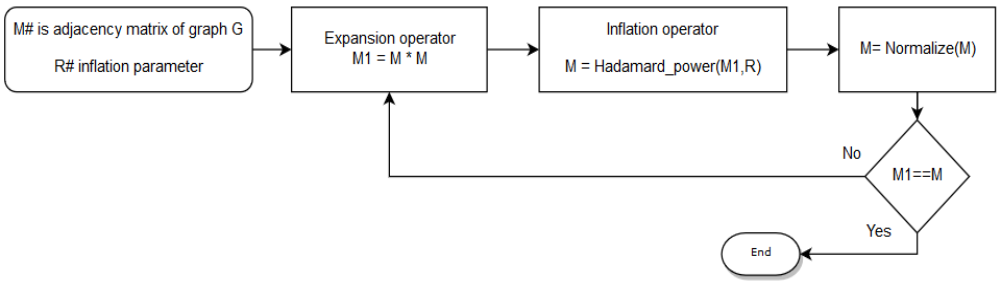


Figure 1. Markov Clustering algorithm

Girvan and Newman (2004) algorithm uses edge betweenness to divide a graph into communities. Edge betweenness of an edge is the number of shortest paths passing through it, the intuition behind this is that edges between two communities tend to have a higher edge betweenness than the edges in a community. Thus, removing edges with the highest edge betweenness divides a graph into communities. The algorithm works as follow:

1. compute edge betweenness of each edge
2. remove the edge with the highest edge betweenness
3. repeat 1 and 2 until there is no more edge to remove

The authors define a measure called modularity to measure the quality of each division, a division is a cluster created by removing an edge with highest betweenness, the division with highest modularity indicates a good division. The idea behind the modularity is that a good clustering produces highly connected nodes in the same cluster, while nodes from different clusters that have fewer connections. Equation (5) (Girvan and Newman 2004) shows how this modularity is computed, where eij is the fraction of all edges in the network that connect cluster i to cluster j and ai = Σj eij.

$$Q = \sum_{i} (e_{ii} - a_i^2) \tag{5}$$

Newman (2003) proposed an alternative approach to Girvan and Newman algorithm. The algorithm starts by setting each node of the graph as a community, then it repeatedly joins communities in pairs, the communities pairs which produces the highest increase of modularity are merged, the process stops when there is no more communities to merge.

Maximizing the modularity is another approach to extract communities from graph, this involves finding a partition that optimizes the modularity. Newman (2006) propose a reformulation of modularity to a so-called modularity matrix B EQ(6), where Aij is the number edges between vertex i and j, ki the degree of vertex i and m = (½)Σi ki. From this modularity matrix, the author tries to find the best partition of two communities which gives the highest modularity. The modularity of a partition s, where si=+1 if vertex i is in one group and -1 otherwise, is defined by EQ(7). The author writes s as combination of normalized eigenvector ui of B, where s= Σi ai ui and ai= uTi.s. From Eq(8), the author chooses s proportional to the eigenvector with most positive eigenvalue, where betai is eigenvalue corresponding to eigenvector ui. Since elements of s have value -+ 1, s is constructed as follow: si =+1 if the corresponding element of ui is positive, -1 otherwise.

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m} \quad (6)$$

$$Q = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s} \quad (7)$$

$$Q = \frac{1}{4m}\sum_i a_i \mathbf{u}_i^T \mathbf{B} \sum_j a_j \mathbf{u}_j = \frac{1}{4m}\sum_{i=1}^{n} (\mathbf{u}_i^T \cdot \mathbf{s})^2 \beta_i \quad (8)$$

After the first partition, each community is partitioned in two communities and so on. The process of splitting stop when a division of a community gives no increase of the modularity.

Louvain algorithm (Blondel et al. 2008) detects communities in large networks with a high modularity. As a first step it starts with N communities, where N is the number nodes of the network, then for each node is moved to a community of its neighbours and it evaluates the modularity, if there is a positive increase of modularity then the node is moved to community with high increase, else the node is left in its community. The process is applied until there is no improvement of modularity. As a second step, a new network is constructed where nodes are communities obtained from previous step then the first step is reapplied. The combination of these two steps is applied until there no improvement in the modularity.

2.2.2 Vector-based clustering

In vector-based clustering, Objects are characterized by a set of features and formalised under the form of vectors. Objects are clustered based on their distance, objects within a given cluster are close in terms of distance and far from objects of other clusters.

Hierarchical clustering starts to consider all vectors (objects) as clusters then it merges the two most close clusters, this process is repeated until all vectors are merged in one cluster. The result is a tree which represents the hierarchy of the vectors. In the process of merging two

clusters, different method can be applied like single, complete, ward, etc. Each method defines its own distance between clusters and uses one of several existing metric distance like Euclidean, cosine, Manhattan to define a distance between two elements.

K-means selects K vectors as centroid, for each k clusters Ci, a centroid is considered as the center of cluster Ci, then all vectors are affected to the cluster with the closest centroid. The centroid of each cluster is recalculated as means of all vectors present in the cluster. This operation is repeated until there is no change in the clusters composition. In the next section, we explain how we build a vectorial representation of a document.

3. OVERVIEW

Document clustering approaches using ontologies generally produce better results (Punitha and Punithavalli 2012, Hotho et al. 2003). Based on these results, we aim to build a structure close to an ontology in the meanwhile minimizing the intervention of humans. Figure 2 summarizes our approach, in the following, we will describe it more in details.

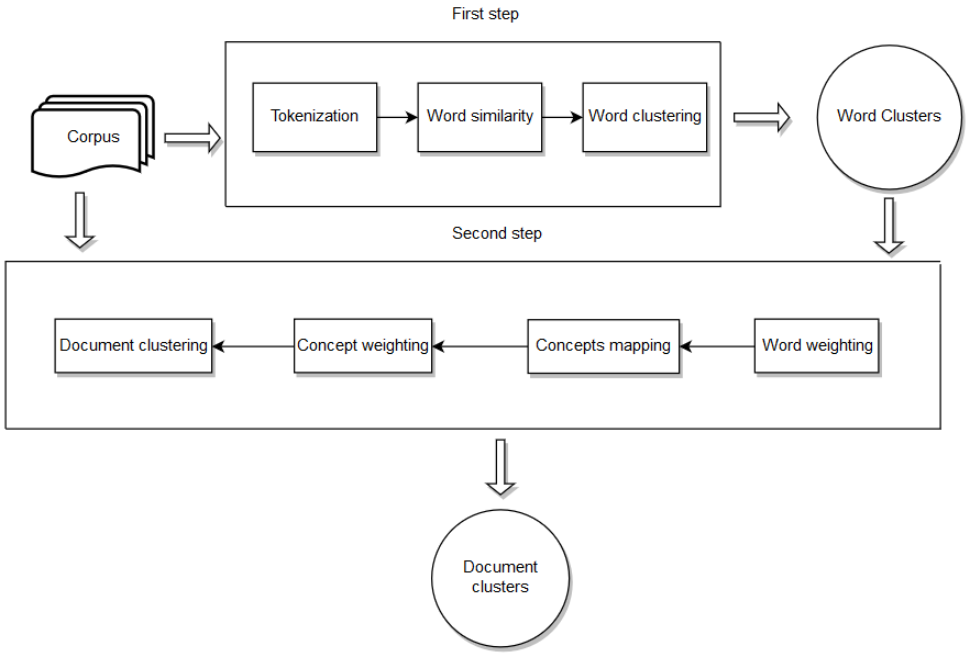


Figure 2. Our approach of document clustering

3.1 Word Clustering

Words are the lowest level at which we can start manipulating documents. One assumption on word similarity is that word co-occurring tend to have the same semantics, as stated by Firth (Firth 1957):" You shall know a word by its company it keeps.". Another assumption is that the topics have different distributions of words. Based on those assumptions, we believe clustering

words into clusters of semantically close words will help us to detect concepts. Indeed, we expect to capture in a cluster all semantic relationships between words related to a concept, and thus, we capture all ways to express the concept.

In this first part, we follow three steps: tokenization, word similarity and words clustering (Sellah and Hilaire 2018(a)). We start by transforming all documents into a set of words, this step is known as tokenization step. This step is an important one, it extracts words that are considered meaningful to the corpus. The choice of the relevant words depends on the corpus, in our case, we focus only on alpha tokens. Stop words and infrequent words are filtered. In the second step, word similarity is computed between all pairs of words. To cluster words, we build a graph, where nodes represent the words and the edges represent PMI distance between the two words. We use Louvain algorithm to detect clusters (communities) in the graph.

We improved this step, by changing the way we compute the PMI distance. In (Sellah and Hilaire 2018(a)), we compute the co-occurrence of two words at document level, in this paper, we consider the statement level, if two words co-occur in the same statement, their co-occurrence is incremented by one. And N represents the size of statements in the corpus.

Another improvement is to consider the obtained clusters, and apply recursively Louvain algorithm for each cluster. This process is stopped when the subclusters modularity is less than 0.3. The modularity ranges between -1 and 1 and clusters with a modularity over 0.3 are considered as good clusters (Newman and Girvan 2004). The result of this process is a set of trees, it can be viewed as a hierarchical representation of clusters.

The root of a tree represents a cluster obtained from the first graph clustering. The internal nodes of a tree represent sub-clustering of their parent node (cluster). The words present in the parent node (cluster) are not automatically present in the internal child nodes (sub clusters), this results from avoiding sub-clusters composed with less than three words. And leafs represent words.

Our aim, after we have recursively divided a cluster into sub clusters, is to extract representative words for roots and their internal nodes. We compute for each word the average distance to other words in a cluster, then, we sort them in decreasing order and select N top words, where N depends on the cluster size, we set N as 10 percent of the size of the cluster.

Until now, we have focused only on unigrams, but in the generated clusters, we have found that clusters composed with less than four words are good candidates to extract bigrams or trigrams. We start from these clusters composed with less than four, then, we generate for each cluster all combinations of words, because words in a cluster are not ordered, then we compute the frequency of bigrams and trigrams. The wrong combinations of bigrams/trigrams will have a low frequency, we expect this frequency to be zero, and the good bigram and trigram will have a high frequency.

3.2 Document Clustering

In this section, we aim to organise all documents in groups of related documents (Sellah and Hilaire 2018(b)). The first step consists of determining which words are important for each document by using TF-IDF. The second step maps word clusters to a document. In this step, a document is represented by a vector V of size N, where N is the number of word clusters (concepts), each element Vi is the weight of the ith cluster (concept).

A cluster (concept) is weighted by a TF-IDF measure, where the term-frequency of a concept is the sum of all words occurring in the document and belonging to the cluster. A word can belong to several clusters, in this case, there is no disambiguation, all occurrences of clusters, where the word occurs, are incremented by its occurrence. The inverse document frequency of a cluster (concept) is the number of document where at least a word in the cluster occurs. Thus,

we determine which clusters (concepts) are important for the document. The final step is the document clustering, we use two approaches to cluster documents: graph-based and vector-based.

In vector-based approach, a document is described by a vector of concepts, where the size of the vector is the number of word clusters and each element of the vector Vi is the weight of the ith concept. Based on the concept-vectors, we apply hierarchical clustering by using the Euclidean distance as metric. Then we use the silhouette method to determine the number of clusters.

In graph-based approach, we build a graph where nodes are documents and edges are the Euclidean distance between two documents. We first compute the Euclidean distance for each couple of documents, then for each document Di, we compute the average distance Mi of its distances to other documents. Next, we create an edge between the document Di and all other documents where the distance between Di and Dj is less than Mi/2. In the case there is no edge created, we create an edge between the document Di and its top five similar documents. After the building of the graph, we apply Louvain algorithm to cluster the documents. We use the modularity to measure the quality of the produced clusters.

4. RESULTS AND DISCUSSION

We use data from Reuters-21578, Distribution 1.0 (Lewis, 2004). Reuters-21578 contains a collection of articles appeared in Reuters newswire in 1987. An advantage of using this collection is that the articles have been associated categories. Thus, we can compare clusters generated by our framework and the article categories. In the clustering step, we consider only articles with at least one topic. Reuters-21578 corpus contains 11367 articles with at least one topic assigned.

The first phase takes the Reuters-21578 corpus. In this paper, we focus only on the title and the body of an article to calculate the semantic distance between words. The framework starts to extract words from the articles, computes PMI measure between each couple of words and then constructs a graph. We use Louvain algorithm to extract clusters from the graph and we measure their quality with the modularity.

In figure 2, we have compared the document approach and statement approach by using the modularity and different threshold values. The figure shows that statement approach produces the best clusters for almost all threshold values. The node size is another parameter to consider, because threshold value has an impact on it. Figure 4 shows the evolution of node size for different threshold values, we can see, in general, that statement approach keeps more words in the clusters than the document approach, which means with statement approach, we detect more semantic relationships between words. Figure 5 shows the evolution of the cluster size for different threshold value, as we can see, statement approach produces more clusters than document size and with a high modularity, which means clusters are good ones. Based on the modularity, word size and cluster size, statement approach overcomes document approach.

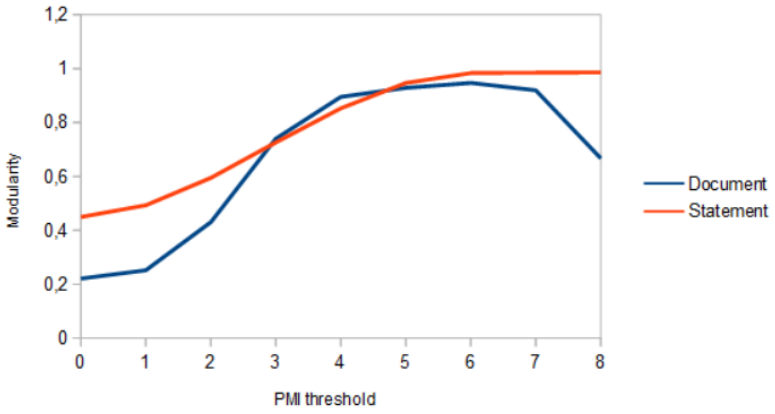


Figure 3. Evolution of the modularity

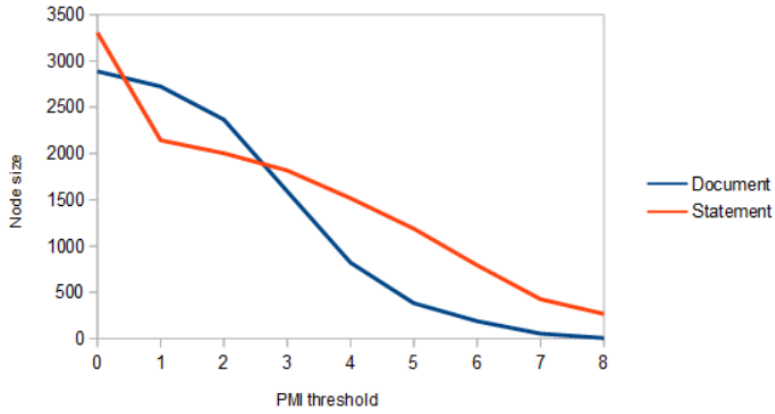


Figure 4. Evolution of node size

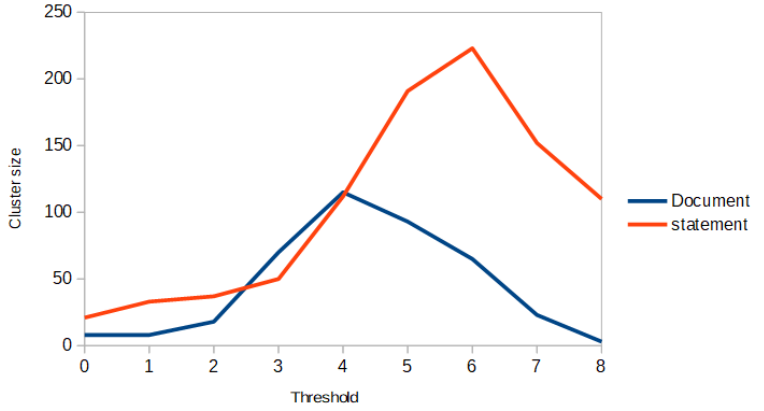


Figure 5. Evolution of cluster size

Figure 6 shows number of clusters by size generated for different threshold values. Bigrams refer to clusters composed only with two words and Trigrams are clusters composed with three words, these clusters are candidates to be significant bigrams and trigrams. Over Trigrams are all other generated clusters with word size is over three. We have extracted 400 unique bigram candidates and 133 unique trigram candidates.

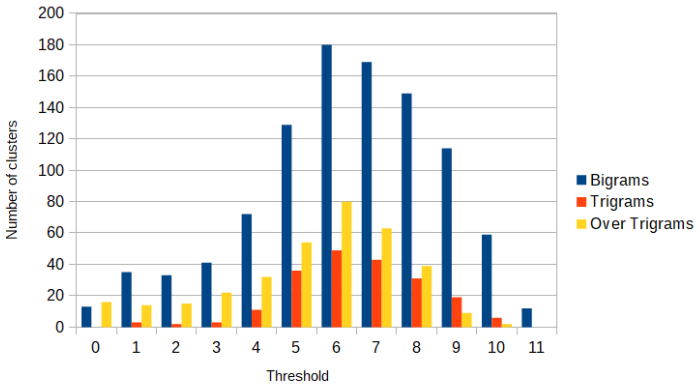


Figure 6. Evolution of number of clusters by size for different threshold value

Table 1. Top 32 bigrams

Bigram	**PMI**	**Bigram**	**PMI**	**Bigram**	**PMI**	**Bigram**	**PMI**
poison pill	11.676	costa rica	10.961	helmut kohl	10.405	goldman sachs	10.285
puerto rico	11.359	jardine matheson	10.82	caspar weinberger	10.397	theodore cross	10.222
margaret thatcher	11.235	hernandez grisanti	10.82	nova scotia	10.368	carter hawley	10.122
eastman kodak	11.121	kidder peabody	10.728	crazy eddie	10.339	jacques delors	10.122
brace jovanovich	11.068	hoechst celanese	10.661	societe generale	10.335	panama canal	10.097
karl otto	11.062	marlin fitzwater	10.556	prudential bache	10.325	lloyd bentsen	10.068
phelps dodge	11.04	protein meals	10.483	honeywell bull	10.304	depletion allowance	10.066
dean witter	10.981	jorio dauster	10.424	asher edelman	10.287	jose sarney	10.004

Bigrams and trigrams candidates are not ordered, we generate all possible combinations for each cluster, for each combination, we compute how many times it occurs without separation between its words. Thus, insignificant bigrams/trigrams and the wrong combinations are filtered. We use PMI to weight bigrams and we keep bigrams with PMI value over 3, table 1 shows top 32 bigrams, from 400 bigram candidates only 201 appear at least in one sentence. Table 2 shows frequency all trigrams which appear at least in one sentence, only 21 trigrams are considered as good from 133 candidates.

Table 3 shows the distribution for unigrams, bigrams and trigrams, we can see unigram size is far higher than others, words composing bigrams and trigrams are also counted in unigrams. As we can see, unigrams represent 94 percent of the corpus content.

Table 2. Frequency of All trigrams extracted from the corpus

Trigram	**Frequency**	**Trigram**	**Frequency**
representative clayton yeutter	219	undersecretary daniel amstutz	16
governor satoshi sumita	99	frozen orange juice	12
drexel burnham lambert	93	swiss sight deposits	6
chief executive officer	93	barney harris upham	5
dean witter reynolds	64	extraordinary items debit	5
karl otto poehl	56	debit extraordinary items	4
harcourt brace jovanovich	46	june venice summit	3
liberal democratic party	20	leading indicators index	2
enhancement program initiative	19	offshore drilling rigs	2
discount window borrowings	18	lift sanctions imposed	2
lloyds shipping intelligence	17	-	-

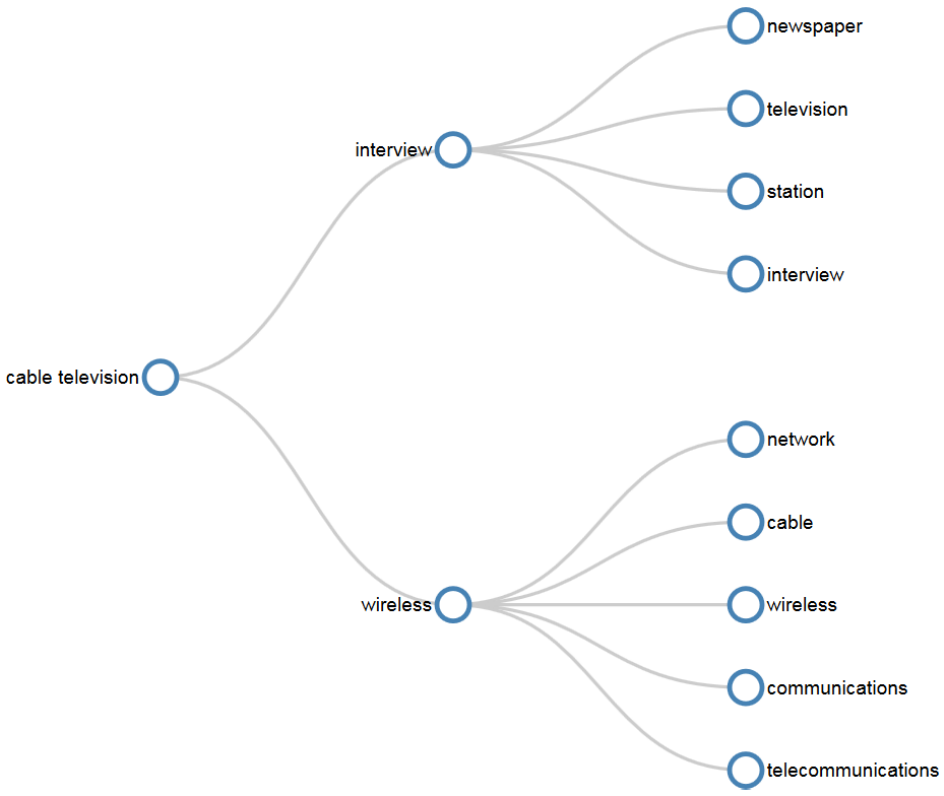
Table 3. Unigram, bigram and trigram distribution

	Unigram	**Bigram**	**Trigram**	**Total**
Size	3487	201	21	3709
Percentage	94%	5.4%	0.6%	100%

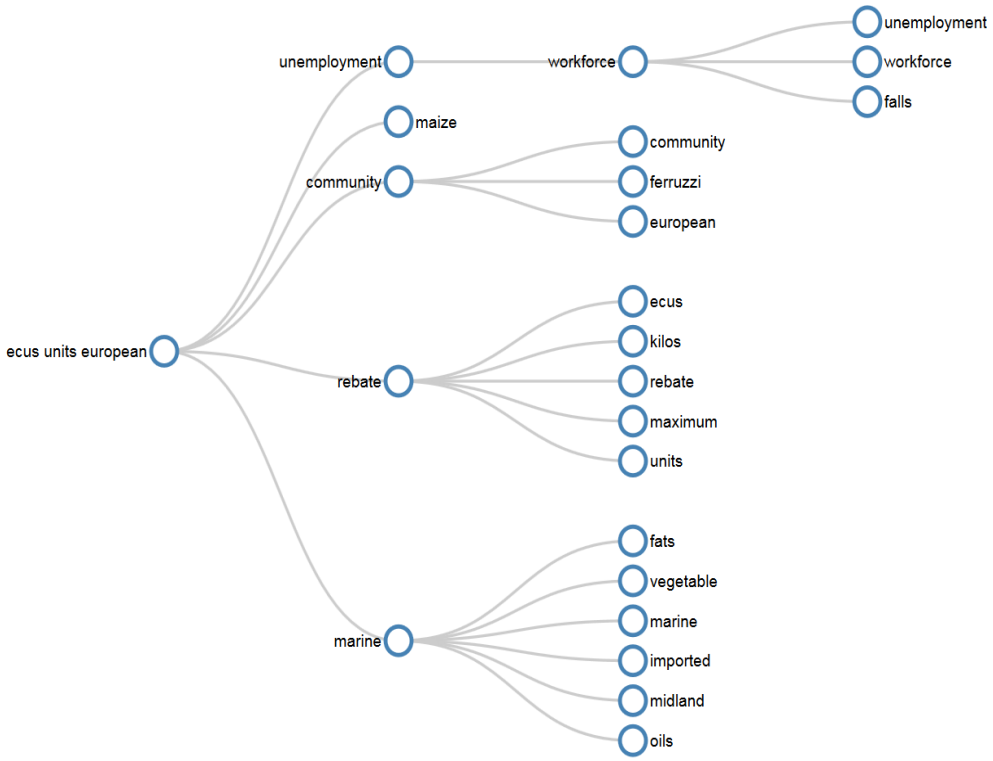
In this paper, we investigated generating hierarchical word clusters, in (Sellah and Hilaire 2018(a)), we have used graph-based clustering using Louvain algorithm to generate a flat word clusters, in this work, we apply Louvain algorithm recursively to word clusters obtained in (Sellah and Hilaire 2018(a)) until the modularity of generated clusters is under 0.3.

The first step is to generate a flat word clusters, for this, we generate the graph based on three parameters: node size, modularity of generated clusters and the size of generated clusters. We choose to set threshold to 5 because it produces the best configuration, it keeps high number of words in the clustering, it produces large number of clusters with a modularity close to 1, which considered as a good value. While threshold with value 6,7 and 8 produce clusters with a high modularity, the node size is much less than threshold value 5 and the clusters are composed only with few words, which is not help to detected concepts. Church and Hanks (1991) observed that paire words with a PMI value over 3 tend to be interesting, based on this observation, we do not consider these word pairs.

Figures 7 shows some hierarchical representation of some flat word clusters. Figures 7 (A) shows the hierarchical decomposition for the initial flat cluster composed of words: cable, telegraph, interview, network, station, telecommunications, communications, wireless, television, newspaper, telephone. This cluster is splitted into two sub clusters and each sub cluster has a representative words, the number of representative depends on the size of words present in the cluster.



(A)



(B)

Figure 7. Hierarchical word clusters

Figure 8 shows the distribution of depth of generated trees, we focus on trees with depth over 2, tree with depth equal to 1 are composed with the root, which represents the initial flat clusters. Therefore, there is no hierarchical representation for these trees with depth equal to 1. Figure 9 shows the evolution of cluster size after applying the recursive clustering for all initial clusters, when threshold ranges from 0 to 2, it produces many clusters, but there are many week semantic relationships which impact the clustering quality, when the threshold is over 6, it produces less clusters with almost no hierarchical clusters. When the threshold is set to 3,4 or 5, it produces almost the same number of clusters, but at 5, it produces more hierarchical clusters with strong semantic relationships. This confirms our intuition on choice of a threshold value, where the modularity, word size and cluster size are good parameters to set it.

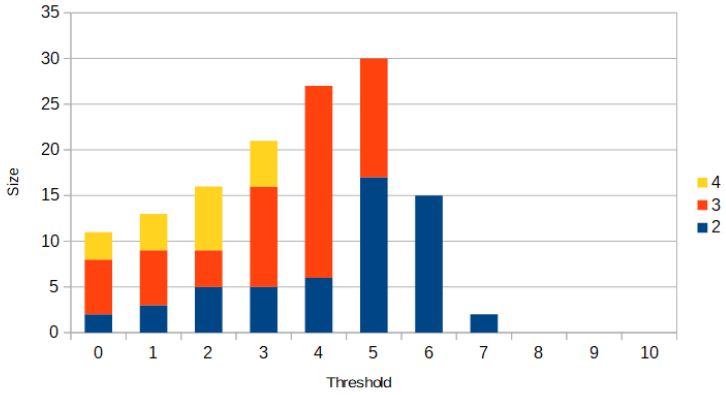


Figure 8. Distribution of tree's depth for different threshold values

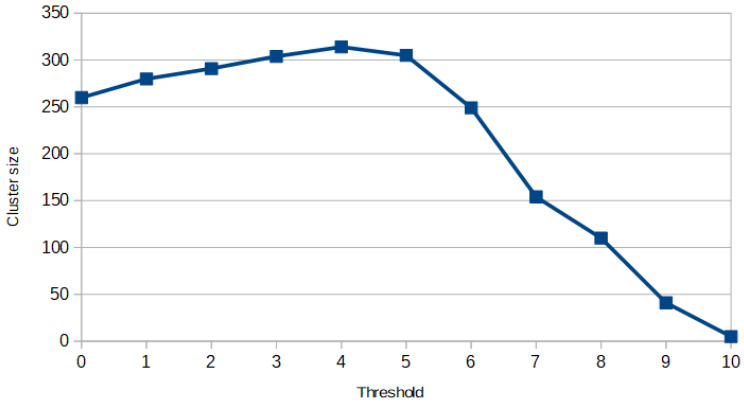


Figure 9. Evolution of cluster size by applying the clustering recursive

5. RELATED WORK

In this section, we present a survey of some approaches in documents clustering. Shah and Mahajan (2012) presented a detailed review of semantic based documents clustering.

Roul (2018) proposed an approach that consists of clustering engine search results and ranking results of each cluster. It follows three steps. The first step aims to select important term, for that, the author creates documents-terms vectors and terms-documents vectors, applies a -means clustering and extracts top terms of each cluster, this step is called clustering-based feature selection (CBFS). The second step is the document clustering, based on the important terms extracted by CBFS, documents are represented in semantic space. Then Euclidean distance is computed between all documents and a graph is constructed, where nodes represent documents and the edges represent the Euclidean distance between two documents. From this min-cut algorithm is used to partition the graph into clusters. As a final step, topics are extracted

for each cluster and for each topics a list of documents is associated. Finally, the documents are ranked.

Hotho et al. (2003) studied the advantages of using an ontology, like WordNet, in document clustering. Authors started by representing a document with vector of words, then they compared three strategies to enrich the vector: add strategy, concept only strategy and replace terms strategy. Add strategy consists to concatenate the vector of terms with the vector of concepts contained in WordNet. Replace terms by concepts strategy is like add strategy, but terms which correspond at least to one concept in WordNet are considered only in the concepts vector. In concept only strategy, a document is described only by its vector of concept. The result obtained in (Hotho et al. 2003) shows that using an ontology improve the clustering, especially when the ontology is well suited to the domain. Punitha and Punithavalli (2012) compared two approaches in documents clustering, they compared a hybrid method based on pattern recognition and semantic driven methods (HSTC) with documents clustering based on an ontology (TCFS). The authors found that TCFS was slightly better than HSTC.

Cosa (Concept Selection and Aggregation) (Staab and Hotho 2003) is a document clustering approach based on ontology. It follows two steps, the first one maps concepts to to documents, to do that, the authors use a tokenizer to extract words from the document, then a lexical analysis is applied, it consists in determining the canonical stems of words and named entity detection. Based on domain lexicon, which contains the mapping between concepts and stems, authors map concepts to a document. After mapping concepts to all documents, an aggregation of concepts is made to replace the too frequent (rarely frequent) concepts into their subconcepts (parent concepts) based on the ontology, the aim of the aggregation is to reduce the size of the used concepts.

Wang and Koopman (2017) propose a new semantic representation of clustering articles. The authors use named entities occurring on the articles to cluster them rather than words. Each entity is described by a vector, which consists of its lexical context. Based on the entity vectors, an article vector is constructed, the article vector is the centroid of all entity vectors occurring in it. The articles are clustered using the article vectors with two approaches, k-means and Louvain community detection.

Romeo et al (2014) propose a framework, named SeMDocT (Segment-based MultiLingual Document Clustering via Tensor Modeling), for documents clustering. SeMDocT decomposes all documents in segments, each segment describes a subtopic and is represented by a vector of word occurrences or a vector of BabelNet synsets. Then, segments are clustered in k clusters. Based on segment clusters, SeMDocT generates document-feature matrix for each cluster. In each cluster, a document is described by a vector of feature in the document-feature matrix, the vector is the sum of all segment vectors of the document present in the cluster. The framework uses the constructed document-feature matrices to build a third-order tensor, where a tensor is a multi-dimensional array. And finally the documents are clustered in K clusters.

6. CONCLUSION

In this paper, we have compared statement approach and document approach using three parameters: node size, cluster size and modularity of generated clusters. The obtained results show that statement approach is better than document approach and 5 seems to be the optimum threshold value for this corpus. Based on these detected semantic relationships, we have

improved clusters extracted in previous work in term of size of the words and the quality of clusters and we built a hierarchical representation for those clusters. We proposed an approach to extract relevant bigrams and trigrams. All these contributions are elements of global work, which consists is automating ontology building by reducing human interactions in the process, we believe word clusters and document clusters will help us to detect concepts. In further works, we will study the impact of bigrams and trigrams in document clustering and the impact of different corpus on our approach. We will also study how detect which concepts behind these word clusters and document clusters and introduce human interaction in this process. For the last point, the introduction of multiagent systems (Hilaire et al. 2003) and the study of hierarchical structures such as holarchies (Rodriguez et al. 2006) may allow the dynamic evolution of hierarchical clusters and the integration of the humans in the process.

REFERENCES

Blondel, V.D., Guillaume, J.L., Lambiotte, R., Mech, E.L.J.S., 2008. Fast unfolding of communities in large networks. *J. Stat. Mech* P10008.

Bollegala, D., Matsuo, Y., Ishizuka, M., 2011. A Web Search Engine-Based Approach to Measure Semantic Similarity between Words. *IEEE Trans. Knowl. Data Eng.* 23, 977–990.

Cilibrasi, R.L., Vitáni, P.M.B., 2007. The Google Similarity Distance.

Firth, J.R., 1957. A synopsis of linguistic theory 1930-55. *Studies in Linguistic Analysis (special volume of the Philological Society)* 1952–59, 1–32.

Hilaire V., Koukam A., Gruer P. (2003) A Mechanism for Dynamic Role Playing. In: Carbonell J.G., Siekmann J., Kowalczyk R., Müller J.P., Tianfield H., Unland R. (eds) Agent Technologies, Infrastructures, Tools, and Applications for E-Services. NODe 2002. *Lecture Notes in Computer Science, vol 2592. Springer*, Berlin, Heidelberg

Hotho, A., Staab, S., Stumme, G., 2003. Ontologies Improve Text Document Clustering, in: *Proceedings of the International Conference on Data Mining — ICDM-2003.* IEEE Press.

Iosif, E., Potamianos, A., 2010. Unsupervised Semantic Similarity Computation between Terms Using Web Documents. *IEEE Trans. Knowl. Data Eng.* 22, 1637–1647.

Jarmasz, M., Szpakowicz, S., 2003. Roget's thesaurus and semantic similarity, in: *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP-03).* Borovets, Bulgaria, pp. 212–219.

Lewis, D.D., 2004. *Reuters-21578 text categorization test collection, Distribution 1.0*. http://www.daviddlewis.com/resources/testcollections/reuters21578/

Mei, J., Kou, X., Yao, Z., Rau-Chaplin, A., Islam, A., Moh'd, A., Milios, E.E., 2015. Efficient Computation of Co-occurrence Based Word Relatedness., in: *Vanoirbeek, C., Genevès, P. (Eds.), DocEng. ACM*, pp. 43–46.

Meng, L., Huang, R., Gu, J., 2013. A review of semantic similarity measures in wordnet. *International Journal of Hybrid Information Technology* 6, 1–12.

Newman, M., 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103, 8577–8582.

Newman, M.E.J., 2003. Fast algorithm for detecting community structure in networks. *Physical Review E* 69.

Newman, M.E.J., Girvan, M., 2004. Finding and evaluating community structure in networks. *Phys. Rev. E* 69, 026113. https://doi.org/10.1103/PhysRevE.69.026113

Punitha, S.C., Punithavalli, M., 2012. Performance Evaluation of Semantic Based and Ontology Based Text Document Clustering Techniques. *Procedia Engineering 30*, 100–106. https://doi.org/10.1016/j.proeng.2012.01.839

Reinse, D., Gantz, J., Rydning, J., 2017. *Data Age 2025*.

Rodriguez S., Hilaire V., Koukam A. (2006) Holonic Modeling of Environments for Situated Multi-agent Systems. In: Weyns D., Van Dyke Parunak H., Michel F. (eds) Environments for Multi-Agent Systems II. E4MAS 2005. *Lecture Notes in Computer Science, vol 3830. Springer*, Berlin, Heidelberg

Romeo, S., Tagarelli, A., Ienco, D., 2014. Semantic-Based Multilingual Document Clustering via Tensor Modeling., in: *Moschitti, A., Pang, B., Daelemans, W. (Eds.), EMNLP. ACL*, pp. 600–609.

Roul, R.K., 2018. An effective approach for semantic-based clustering and topic-based ranking of web documents. *I. J. Data Science and Analytics 5*, 269–284.

Shah, N., Mahajan, S., 2012. Semantic based Document Clustering: A Detailed Review. *International Journal of Computer Applications* 52, 42–52. https://doi.org/10.5120/8202-1598

Smail Sellah, V.H., 2018(b). A document clustering approach for automatic building of ontologies. *The Second International Workshop on Data Science Engineering and its Applications 6*.

Smail Sellah, V.H., 2018(a). Automatic Generation of ontologies: comparison of words clustering approaches. *International Conferences WWW/Internet 2018 and Applied Computing 2018* 269–276.

Spanakis, G., Siolas, G., Stafylopatis, A., 2009. A Hybrid Web-Based Measure for Computing Semantic Relatedness Between Words, in: *2009 21st IEEE International Conference on Tools with Artificial Intelligence. Presented at the 2009 21st IEEE International Conference on Tools with Artificial Intelligence (ICTAI), IEEE*, Newark, New Jersey, USA, pp. 441–448. https://doi.org/10.1109/ICTAI.2009.64

Staab, S., Hotho, A., 2003. Ontology-based Text Document Clustering., in: *Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'03 Conference* Held in Zakopane. pp. 451–452.

Terra, E., Clarke, C.L.A., 2003. Frequency estimates for statistical word similarity measures, in: *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Association for Computational Linguistics, Edmonton, Canada, pp. 165–172. http://dx.doi.org/10.3115/1073445.1073477

Wang, S., Koopman, R., 2017. Clustering articles based on semantic similarity. *Scientometrics 111*, 1017–1031.