Journal of Universal Computer Science, vol. 26, no. 1 (2020), 71-88
submitted: 30/12/18, accepted: 15/11/19, appeared: 28/1/20 CC BY-ND 4.0

Label Clustering for a Novel Problem Transformation in Multi-label Classification

Smail Sellah
(CIAD UMR 7533 Univ. Bourgogne Franche-Comte, UTBM, Belfort, France
and
HTI automobile groupe APSIDE, France
smail.sellah@utbm.fr)

Vincent Hilaire
(CIAD UMR 7533 Univ. Bourgogne Franche-Comte, UTBM, Belfort, France
vincent.hilaire@utbm.fr)

Abstract: Document classification is a large body of search, many approaches were proposed for single label and multi-label classification. We focus on the multi-label classification more precisely those methods that transformation multi-label classification into single label classification. In this paper, we propose a novel problem transformation that leverage label dependency. We used Reuters-21578 corpus that is among the most used for text categorization and classification research. Results show that our approach improves the document classification at least by 8% regarding one-vs-all classification.

Key Words: Classification, Clustering, Feature extraction, Ontology
Category: I.5, I.2.6

1 Introduction

Nowadays, companies are dealing with a large amount of data and these data keeps growing in size, IDC, in its white paper named Data Age 2025, forecasts that, by 2025, almost 60% of The global Datasphere will be produced by companies [Gantz and Rydning, 2017]. Exploiting this amount of data is a challenging task in many domains like information retrieval, data mining and machine learning. Textual data are the most predominant type of data in companies, thus, we focus on it.

Our global work aims to build a semantic representation of documents by using a semantic structure automatically constructed from a corpus of documents. Based on this representation, we aim to define a mechanism that automatically organize documents into clusters of semantically related documents. We believe this organization will ease the design of an information retrieval system which will allow us to exploit document semantic during requests and documents mining.

A classical representation of a document is a vector, where each element of the vector represents the weight of a term respectively to the document. The size of the vector depends on the size of the dictionary used to describe documents. The dictionary can be the set or a subset of all words present in the corpus. The weight of a word can be defined as either its frequency in the document or TF-IDF (term frequency inverse document frequency). The latter definition is more popular, it gives more importance for words occurring several time in the document and less importance to words occurring in many documents. This approach has many limitations, one of these limitations is that two documents describing a same idea with different words (synonyms) are not considered similars.

In order to tackle these limitations, many works were proposed. They can be divided into two categories, supervised and unsupervised learning. In unsupervised learning, some authors proposed a new document representation which consists in: reducing the size of features [Roul, 2018] or taking advantage of semantic in documents [Hotho et al., 2003,Wang and Koopman, 2017,Romeo et al., 2014] or both [Staab and Hotho, 2003,Sellah and Hilaire, 2018b]. In supervised learning, many works try to reduce the dimensionality of the document's features, these works can be organized into two categories, features extraction and features selection. Features extraction transforms the initial high dimensional document vector into a lower dimensional document vector, while features selection selects a subset of initial features set.

In [Sellah and Hilaire, 2018a], we proposed an approach for building an ontology automatically from a corpus of documents, we construct word clusters of semantically close words, we believe that these word clusters will help us to capture concepts or vocabulary related to concepts. In our previous work [Sellah and Hilaire, 2018b] which can be categorized as unsupervised learning, we proposed a new document representation, it's based upon word clusters extracted from documents by using our method presented in [Sellah and Hilaire, 2018a], each word cluster is considered as a feature. Using word clusters as document features has several advantages, first it reduces the dictionary size by keeping only words which have a strong semantic relationship between them. And by clustering words into clusters of semantic close words, it reduces the features size of documents and the word clusters help to leverage semantic of the documents. Works like [Rattinger et al., 2018,Smadi and Qawasmeh, 2018,Cruz et al., 2018] are close in spirit to our global work.

In this paper, which is an extended version of [Sellah and Hilaire, 2018b], we are interested in document classification, while in [Sellah and Hilaire, 2018b], we were interested in document clustering. We are particularly interested to classify document labeled with multiple labels. Our approach consists to transform the initial label set into a new label set, the aim of this transformation is to produce documents labeled only with a single label. Our transformation measures the semantic similarity of labels, then, clusters them into clusters of semantic labels where each cluster of labels represents a new label.

Our main contributions, in this paper, are as follows :

- We propose a novel transformation method, which leverage semantic of labels to create a new set of labels.
- We compare word clusters as feature space to LDA topics as feature space.

We use the popular Reuters-21578 corpus [Lewis, 1997], which provides documents annotated manually by human. We choose this corpus because it is the most popular corpus for documents classification and many results are available. Thus, we can compare our method to the other on the same basis.

The reminder of this paper is as follows : in section 2, we will discuss the related work. in section 3, we introduce an overview of our framework. In section 4, we discuss our experimental results and in section 5, we conclude. Results show that our approach improves the document classification at least by 8% regarding one-vs-all classification.

2 Related Work

Before introducing the related work, we will present a definition multi-label classification, this definition is based on definitions found in [Tsoumakas et al., 2010, Boutell et al., 2004, Tsoumakas and Vlahavas, 2007]. Given a finite set of labels L= l_i, i=1..N where N is the number of labels, multi-label classification is a problem of classifying a document with a set label Y, where Y is a subset of the label set L, the document set D= (d_i,Y_i), i=1..M where M is the number of documents. In single-label classification a document is labeled only with one label, which means the subset label Y has a cardinality $\|Y\| = 1$.

Tsoumakas and Katakis [Tsoumakas and Katakis, 2007] categorized multi-label classification in two main categories : problem transformation methods and algorithm adaptation methods. They define transformation methods as approaches that transform multi-label classification into one or more single-label classification. Authors define algorithm adaptation methods as approaches that extends specific learning algorithms to handle multi-label data directly. In this work, we will focus on the first category.

Binary relevance [Tsoumakas and Katakis, 2007], also known as one-vs-all [Rifkin and Klautau, 2004], defines a classifier for each label. Each classifier i is trained to recognize a label i on a specific data set, the data set is constructed from the transformation of the initial data set, where document labeled with the ith label are labeled as positive examples where the other are labeled as negative examples. Given a new instance, one-vs-all return a set of labels which are predicted positively by their classifier.

Chen et al. [Chen et al., 2007] propose an Entropy-based Label Assignment transformation. It transforms the multi-label classification into a single-label classification by reweighting the features vector. It is based on the intuition that the more a document is tagged with a label, the less it has the probability to belong to these labels. The authors proposed that the probability a document belonging to a label decreases monotonically by its entropy. Thus, they reweight the document features by dividing their weight by $\frac{1}{N}$ where N is the number of document's labels.

Another problem transformation, named label powerset (LP) in [Tsoumakas et al., 2010] and PT3 in [Read, 2008], consists in considering all unique combinations of labels in the data set as a single label. Thus, the multi-label is transformed in a new set of a single label, where each single label refers to a subset of the initial label set. The pruned problem transformation (PPT) method [Read, 2008] is similar to label powerset, but PPT does not consider a combination which occurs less than a certain threshold.

The random k-labelsets (RAkEL) method [Tsoumakas and Vlahavas, 2007] proposes an ensemble of classifiers, where each classifier is trained to recognize a small subset of the label set. It aims to take into account the label correlation and try to avoid the lack of examples present in label powerset. The authors define k-labelset as all possible subsets from the initial label set of size k, the method constructs m classifiers, where each classifier learn to recognize a specific k-lebelset, the m k-labelset are selected randomly, the number of classifiers m and the size of k-lableset k are user defined.

Hüllermeier et al [Hüllermeier et al., 2008] propose an approach of label ranking by pairwise comparisons (RPC). The authors use a transformation called pairwise learning. It consists in learning between each pair of labels which one is well suited for a document. A classifier is constructed for each pair of labels and trained on a dataset constructed from the initial dataset. The new dataset for each pair contains document labeled with at least one of the label of the pair, but not labeled with both. The transformation learns $\frac{m(m-1)}{2}$ classifier where m is the total number of labels. Each classifier prediction is considered as a vote for the label predicted. The label which receive the greatest number of votes is ranked with the first position, the second with the greatest number of votes is ranked with the second position, and so on.

The features size is another limitation for single-label or multi-label classification, to address this limitation, many approaches were proposed to reduce the features size for single-label classification and some can be used for multi-label classification [Tsoumakas et al., 2010, Chen et al., 2007]. These approaches can be organized in two categories : features selection and features extraction [Tsoumakas et al., 2010].

In this paper, we focus on the features extraction approach Latent Dirichlet Allocation (LDA) [Blei et al., 2001], LDA assumes that documents are composed by a mixture of latent topics where each topic contributes in certain percentage of the meaning of a document. Therefore, LDA transforms the bag of words document representation into latent topic representation, the number of topics is assumed to be known. The latent topic representation is a vector where each element i of the vector represents the probability distribution of the ith topic.

3 Overview

Our framework aims to cluster documents into clusters of semantically related documents, it's composed by three components knowledge extraction, semantic document representation and document organization. Our framework relies on these components and a set of documents to organize them into clusters, each component has its objective.

Knowledge extraction aims to build a semantic structure like an ontology, this component takes the set of documents and builds word clusters where each word cluster is composed by semantically related words, word similarity is based on Firth's assumption :*'You shall know a word by its company it keeps.'* [Firth, 1957]. Our assumption is that clustering words into clusters with semantically close words can lead to detect concepts or a vocabulary related to concepts. This component follows three steps tokenization, word similarity and words clustering, this component was presented in our previous work [Sellah and Hilaire, 2018a], we will resume it in subsection 3.1.

Document representation [Sellah and Hilaire, 2018b] component takes word clusters produced by Knowledge extraction and the set of documents, it aims to build a document representation which captures document's semantic, this will help us to organize documents into clusters of related documents. Our document representation relies on word clusters to represent document, instead a bag of words representation, our document representation two advantages, documents expressing the same idea but with differents words will be described with the same concepts (clusters), because they use similar words which are clustered in same clusters. Thus, the documents will be considered as similar. While in the bag of words representation, these documents will not be considered as similar.

Document organization component aims to organize documents into clusters of related documents, it takes the document representation built for each document by the previous component. Based on this representation, this component uses two approaches to organize documents into clusters, supervised method and unsupervised method. The unsupervised method was described in our previous work [Sellah and Hilaire, 2018b] and will be discussed in the subsection 3.3. The supervised approach is introduced in this paper and it will be described in more details in the subsection 3.4. Our framework is illustrated in figure 1.

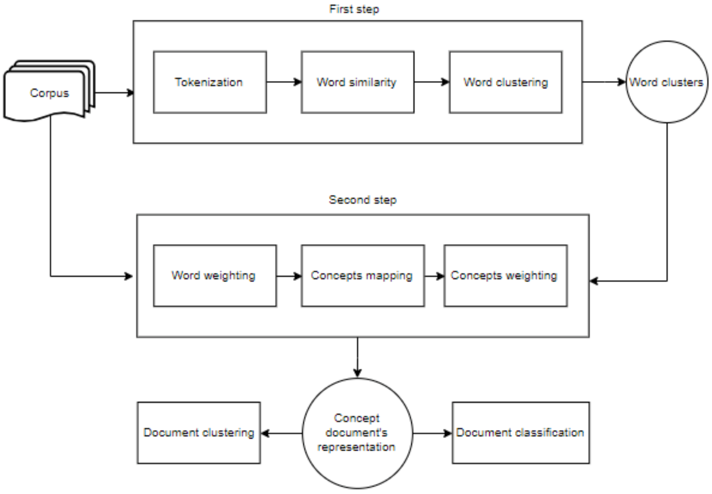

Figure 1: An approach of document clustering and document classification

3.1 Knowledge extraction component

This phase aims to build word clusters where words belonging to a same cluster are semantically close. To build these clusters, our frameworks [Sellah and Hilaire, 2018a] starts 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 for all pairs of words using PMI (Pointwise Mutual Information) [Terra and Clarke, 2003] measure to detect semantic relationship between words. It is computed as follows :

$$PMI(w_1, w_2) = \frac{log_2(\frac{D(w_1,w_2)}{N})}{\frac{D(w_1)}{N} + \frac{D(w_2)}{N}} \tag{1}$$

Where D(w) is the number of documents where the word w occurs, D(w_1,w_2) is the number of documents where words w_1 and w_2 co-occur and N is the size of documents in the corpus.

To cluster words, we build a graph, where nodes represent the words and edges represent PMI distance between words. We use Louvain algorithm to extract clusters (communities) in the graph. It starts with N clusters, where N is the number of nodes in the graph. Clusters are merged until there is no improvement of the modularity. A second step consists in transforming the clusters detected in the first step into a new graph and then applies again the first step. Louvain algorithm alternates these steps until there is no improvement related to the modularity.

The modularity measures the quality of a graph-based clustering. Equation 2 [Newman and Girvan, 2004] shows how it is calculated, where e_{ij} is the fraction of all edges of the network that connect cluster i to cluster j and $a_i = \sum_j e_{ij}$. Modularity ranges between -1 and 1, where clusters with a modularity higher than 0.3 are considered as a good clusters.

$$Q = \sum_i (e_{ii} - a_i^2) \tag{2}$$

3.2 Document representation component

In this phase, we aim to build a representation of documents which consists of three steps. The first step consists in weighting documents' words, a document is represented by a vector V where each element V_i represents the weight of ith word, the vector size is the number of all unique words present in the corpus or a subset of these words.

We use TF-IDF to weight words, we can use a threshold to filter words where words with a weight under a certain threshold will be set to zero, these words are considered as noise for the document, they don't participate to the meaning of the document. TF-IDF of the ith word in jth document is calculated as follows :

$$TF - IDF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} * log(\frac{\|D\|}{d_i}) \tag{3}$$

where n_{ij} is the number occurrences of ith word in the jth document, $\|D\|$ is the number of documents in the corpus and di is the number of the documents where ith word occurs. Thus, words with a high TF-IDF are more important for a document.

The second step consists to map word clusters, which we consider as concepts, to the documents. In this mapping step, we assign a word cluster to the document if at least one word of the word cluster occurs in the document.

The final step which consists to build a concept vector representation of a document, the size of the concept vector is the number of word clusters and each element of the vector represents the weight of a concept, we use TF-IDF to weight concept (word clusters), where TF of a concept is the number of word of word cluster that occur in the document and DF is the number of document where the concept (word cluster) has at least one word that occurs in it. The following example illustrate this concept weighting.

Let be C_1={A, B, C}, C_2={E, F}, C_3={X, Y, Z} three word clusters and D=[D_1, D_2, D_3, D_4] a corpus of documents, where the documents are composed as follows :

- D_1=[A,A,E,B,Z]
- D_2=[E,F,F,Y,Z]
- D_3=[Y,Z,X,A,F]
- D_4=[E,F,C,B]

The concepts mapped to each document are as follows :

- D1=[C_1,C_1,C_1,C_2,C_3]
- D2=[C_2,C_2,C_2,C_3]
- D3=[C_1,C_2,C_3,C_3,C_3]
- D4=[C_1,C_2,C_2,C_2]

The weight of C_1 in D_1 is computed as follow : the frequency of C_1 in D_1 is $\frac{3}{5}$, the document frequency of $C_1 = 3$, which gives TF-IDF of $C_1 = \frac{3}{5} * \log\left(\frac{4}{3}\right)$.

3.3 Document clustering

Document clustering was the subject of our previous paper [Sellah and Hilaire, 2018b] which is extended by this paper. In this phase, we aim to organize all documents in groups of related documents, mainly there is two approaches to organize documents, classification and clustering. In previous work [Sellah and Hilaire, 2018b], we focused on document clustering which is an unsupervised méthod, in this paper, we will focus on the classification méthod which is a supervised method. One aims of our work ([Sellah and Hilaire, 2018b] and this paper) is to see if our semantic document representation works well for both approaches. We want to cover these two aspects of organising documents (clustering and classification) to build a robust framework, because it will be a component of an information retrieval system which we seek to build.

In [Sellah and Hilaire, 2018b], we used 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 V_i is the weight of the i^{th} 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 D_i, we compute the average distance M_i of its distances to other documents. Next, we create an edge between the document Di and all other document where distance between D_i and D_j is less than $\frac{M_i}{2}$. In the case there is no edge created, we create an edge between the document D_i 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.

Document clusters obtained for each approach were evaluated by a quality measure adapted to each one. For vector-based approach, we used Silhouette measure [Rousseeuw, 1987], it measures how close are elements of a clusters and how far they are from other element outside the cluster, it ranges between [-1..1], negative values indicate that the elements are not in the appropriate cluster, while values around 0 indicate overlapping clusters and values over 0.7 indicate well separated clusters [Kaufman and Rousseeuw, 1990]. In this approach, we obtained clusters with Silhouette value around zero which indicates overlapping clusters. For graph-based approach, we used a quality measure named modularity, it's used to measure the quality of clusters extracted from a graph, it ranges between [-1..1]. A good clustering in graph is characterized by a high interconnection between elements within a cluster and a few connection between elements of different clusters. Clusters with a modularity over 0.3 considered as good clusters [Newman and Girvan, 2004]. In our experiment, we connected documents only to their semantically related documents, with this strategy, we obtained clusters with a modularity over 0.6, which indicates a good quality of the clusters.

3.4 Document classification

In this work, we propose a new problem transformation for multi-label classification. Our problem transformation considers dependant labels as a broader single label, it's based on the assumption that dependant labels share some common vocabulary, the bigger the vocabulary is, the bigger is the dependency between these labels. Thus, learning to classify documents into these labels separately is tough because of this common shared vocabulary.

Methods like RAkEL, label powerset and pruned problem transformation try to overcome the labels dependency by transforming all label combinations occurring in corpus into a new single label problem, while pairwise comparison try to overcome this dependency by learning to distinguish between each couple of labels.

Our problem transformation proposes a novel approach to handle labels dependency. Our assumption is that two labels frequently used together to annotate documents are dependent. These labels seems to share a common vocabulary which means that documents annotated only with one of the two labels can be considered to be annotated with the other label. It starts to extract labels dependency of each couple of labels. In order to measure the label dependency, we use equation 1, where D(w) is the number of documents annotated with label w, D(w_1,w_2) is the number of documents annotated with both labels (w_1,w_2) and N represents the number of documents in the corpus. Then it builds a graph where nodes represent labels and edges represent PMI distance between labels and finally we use Louvain algorithm to cluster labels into sets of dependent labels. Once the labels dependency computed, it transforms the initial multi-label classification into single label classification where each set of dependent labels is considered as a new single label and documents are annotated by a new label only if one of the old labels belong to a set of dependent labels. Our problem transformation approach is depicted figure 2.

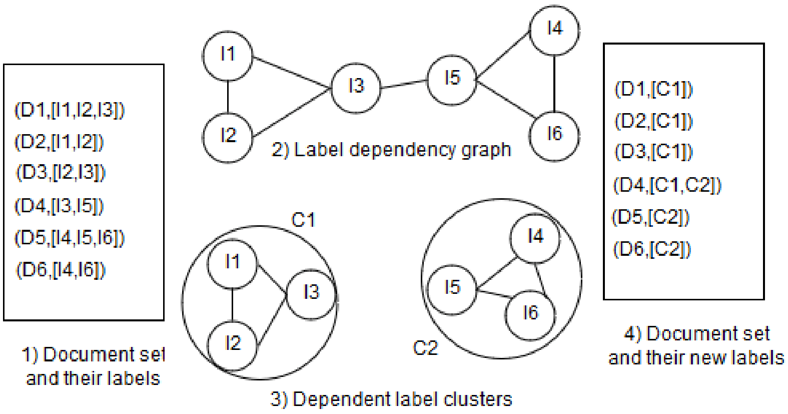

Figure 2: Our problem transformation approach

To best of our knowledge our problem transformation it the first attempt that proposes to leverage labels dependency to transform multi label classification problem into a single label classification.

4 Results and discussion

4.1 Dataset

We use data from Reuters-21578, Distribution 1.0 [Lewis, 1997]. Reuters-21578 contains a collection of articles appeared in Reuters newswire in 1987. This corpus is among the most used for text categorization research. The dataset contains 21578 articles annotated by humains, there is 11367 articles associated at least with one label. The distribution of labels are unequal, the following table shows the number of labels assigned to documents over threshold :

Occurrence threshold	Number of labels
250	10
150	15
50	38

Table 1: Number of labels regarding their number assignation to document

In literature, we can find some statistics measures to characterize a dataset [Tsoumakas et al., 2010, Read, 2008], label cardinality and label density. Label cardinality measures the average label assigned per document and its compute as follows :

$$Cardinality = \frac{1}{m} * \sum_{i=1}^{m} \|Y_i\| \tag{4}$$

Where M represents the document size and Y_i represents the label set associated with the i^{th} document. Label density is computed as follows :

$$Density = \frac{1}{m} * \sum_{i=1}^{m} \frac{\|Y_i\|}{q} \tag{5}$$

Where q represents the size of labels present in the dataset.

	$\|D\|$	$\|L\|$	Cardinality	Density
Dataset	11367	120	1.26	0.01

Table 2: Dataset characteristics

We use one-vs-all approach to test the efficiency of our problem transformation, we split the dataset into 70% for training and 30% for testing, the splitting is as follows, for each label, we sort its documents by their id in ascending order. The first documents are selected for training and remaining documents are used for testing. Concerning the document not annotated with the label, we do the same, we order them by their id in ascending way, then the first documents are used of training and the remaining are used for the testing.

In this paper, we use the implementation of Logistic regression and Linear Support Vector Machine provided by Spark's machine learning library [1]. The parameter are set to default, except number of iteration is set to 100.

[1] https://spark.apache.org/docs/latest/ml-classification-regression.html

4.2 Evaluation mesures

In order to compare the efficiency of our approach, we measure the performance of the classifiers using the well known performance measure precision, recall and f1 measure.

The precision and recall for a given label is defined as follows :

$$Precision(i) = \frac{TP_i}{(TP_i + FP_i)} \tag{6}$$

$$Recall(i) = \frac{TP_i}{(TP_i + FN_i)} \tag{7}$$

Where TP_i represents the total exemples labeled with i which are correctly predicted as belonging to the label i, FP_i represents the total exemples not belonging the label i and predicted as belonging to the label i, FN_i are the total exemples belonging to the label i and predicted as not belonging to the label i. We use the micro average to compute the precision and recall of the classifier, it can be calculated as follows :

$$Precision = \frac{\sum_i TP_i}{\sum_i P_i + \sum_i FP_i} \tag{8}$$

$$Recall = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FN_i} \tag{9}$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{10}$$

4.3 Experimentation

The first phase takes documents with at least labeled with one label in Reuters-21578 corpus. 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, its computes the occurrence for each word and the co-occurrence for each couple of words, we consider two words co-occurring if they are in the same statement. Then, it computes PMI measure between each couple of words and constructs a graph. We use Louvain algorithm to extract clusters from the graph and we measure their quality with the modularity.

Figure 2 shows the evolution of of modularity by setting a threshold in the process of building the graph. Church and Hanks [Church and Hanks, 1989] observed that pair words with a PMI value over 3 tend to have interesting semantic relationship. Thus, Couples of words which have their PMI value under the threshold value are not added to the graph, the aim of this is to filter weak semantic relationship between words. Newman and M. Girvan [Newman and Girvan, 2004] show that a clustering with a modularity over 0.3 is a good clustering.

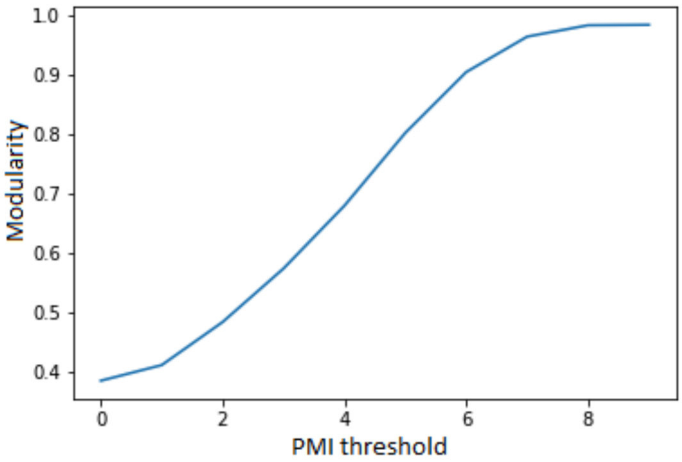

Figure 3: Evolution of modularity for different values of threshold

We see from the graph that the generated clusters are good ones, even for a small threshold value, we start from zero because negative PMI value and zero PMI value indicate words are occurring together or they are independant [Church and Hanks, 1989]. When we compute the PMI value between words, we filter words which occur less than 6 times in the corpus because PMI formula is unstable for small occurrence value [Church and Hanks, 1989]. Therefore, we computed PMI between 9383 words. Figure 3 shows number of words which have a semantic relationship between them, this number decrease when the threshold is set to higher value. Figure 4 shows the evolution of the number of semantic relationship between these words for different threshold value. We can see the number of strong semantic relationship are decreasing with higher threshold value and these strong semantic relationships are between a small number of words. Reuters-21578 top 10 labels are : acq, earn, grain, corn, wheat, trade, interest, money-fx, crude, ship. We apply to these label our proposed problem transformation, first we compute the PMI value between each couple of labels, then, we build a graph and apply Louvain algorithm to organize them into clusters of related labels as we did with words. The resulting clusters are presented in table 4. We can see from the table 4, labels : grain, corn and wheat are related, in other words, these labels are frequently used to annotate same documents, which means that they form a broader label. Table 5 shows the performance for LSVM and Logistic regression, we can see the proposed problem transformation improves the performance of both classifier comparing it to the initial labels.

We do the same but for all labels which have at least one occurrence, ta-

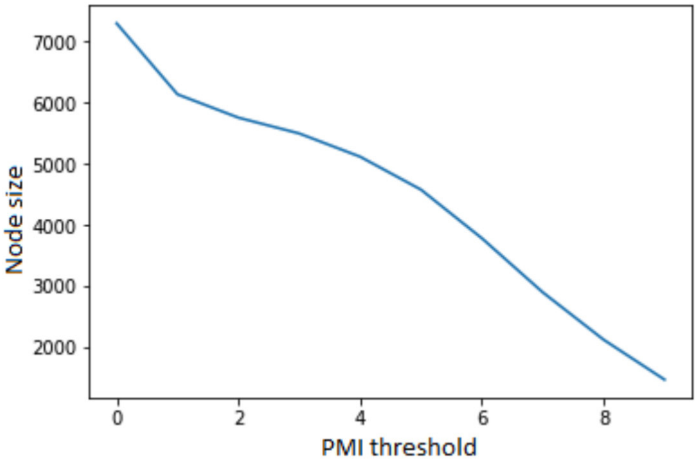

Figure 4: Evolution of nodes size in the graph for different values of threshold

Clusters	New labeling
grain, corn, wheat	0
trade, interest, money-fx	1
crude, ship	2
earn	3
acq	4

Table 3: New labels and their corresponding initial labels

ble 6 shows the obtained label clusters, clusters formed with one label are not represented in the table. New label which have less than 100 exemples are not considered.

Table 8 shows the performance of classifiers with LDA topics as features of documents. We can see, using word clusters as features representation for documents produce better than LDA topics representation.

5 Conclusion

In this paper, we have presented an approach that aims to contribute to the design of an information retrieval system able to deal with large amount of data generated by company workers during their projects. More specifically, the problem of automatic document classification with multiple labels is addressed. The

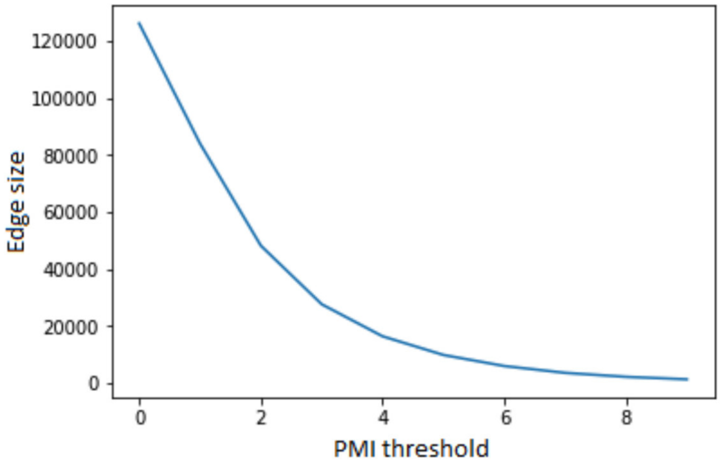

Figure 5: Evolution of number of semantic relationship for different threshold value

Classifier	Labeling	Precision	Recall	F1
Logistic regression	Initial label	0.715	0.858	0.780
Logistic regression	New label	0.863	0.902	0.863
LSVM	Initial label	0.701	0.871	0.777
LSVM	New label	0.829	0.90	0.863

Table 4: Classifiers performance with initial labeling and the new labeling

Clusters	New labeling
copper, zinc, gold, platinum, strategic-metal, lead, silver	0
yen, reserves, money-fx, dlr, interest, stg, dmk, money-supply	1
tea, cocoa, livestock, hog, sugar, coffee, l-cattle, carcass	2
sunseed, soybean, oat, barley, groundnut, rice, soy-oil, wheat, meal-feed, corn, oilseed, cotton, soy-meal, grain, sorghum, rapeseed	3
ipi, trade, iron-steel, jobs, cpi, gnp, bop	4
ship, fuel, gas, nat-gas, crude, heat, pet-chem	5
palm-oil, veg-oil, rape-oil, sun-oil, coconut-oil	6

Table 5: New labels and their corresponding initial labels

Classifier	Labeling	Precision	Recall	F1
Logistic regression	Initial label	0.489	0.781	0.601
Logistic regression	New label	0.634	0.837	0.721
LSVM	Initial label	0.432	0.879	0.579
LSVM	New label	0.595	0.865	0.705

Table 6: Classifiers performance with initial labeling and the new labeling

Dataset	Classifier	K	Perplexity	Precision	Recall	F1
Top 10	Logistic regression	4	7.7056	0.548	0.918	0.686
Top 10	LSVM	4	7.7056	0.549	0.918	0.688
Top 10	Logistic regression	10	7.9038	0.503	0.846	0.631
Top 10	LSVM	10	7.9038	0.500	0.858	0.632
Top 10	Logistic regression	50	9.1837	0.403	0.875	0.552
Top 10	LSVM	50	9.1837	0.400	0.905	0.555

Table 7: Classifiers performance with LDA topics representation and initial labels

idea underlying the presented work is to learn labels dependency and organize them into clusters of dependent labels, each cluster forming a new label.

This approach is detailed and some experiments are done through the Reuters-21578 corpus. Obtained results show that this approach produces a better classification than a classification which consists to learn to classify each label separately, it shows that the f1 measure improved by 8% for a small set of labels and 12% for a larger label set. Using word clusters as feature space produces better results than LDA topics as feature space, results show that our representation led to an improvement at least of 8%. These results are positive and encourage us to continue to develop the underlying concepts.

In further works, we will investigate the pertinence of applying recursively our proposed problem transformation for each label clusters. The concept of holon [Rodriguez et al., 2005] will help us to deal with the multiple levels generated by these clusters. We will also explore approaches to improve our semantic representation of document. The overall results are planned for integration within an intelligent system for knowledge management designed according to multi-agent systems such as [Monticolo et al., 2014].

References

[DBL, 2018] (2018). *Fifth International Conference on Social Networks Analysis, Management and Security, SNAMS 2018, Valencia, Spain, October 15-18, 2018.*

IEEE.

[Blei et al., 2001] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2001). Latent dirichlet allocation. In Dietterich, T. G., Becker, S., and Ghahramani, Z., editors, *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 601–608. MIT Press.

[Boutell et al., 2004] Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771.

[Chen et al., 2007] Chen, W., Yan, J., Zhang, B., Chen, Z., and Yang, Q. (2007). Document transformation for multi-label feature selection in text categorization. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM 2007), October 28-31, 2007, Omaha, Nebraska, USA*, pages 451–456. IEEE Computer Society.

[Church and Hanks, 1989] Church, K. W. and Hanks, P. (1989). Word association norms, mutual information and lexicography. In Hirschberg, J., editor, *27th Annual Meeting of the Association for Computational Linguistics, 26-29 June 1989, University of British Columbia, Vancouver, BC, Canada, Proceedings.*, pages 76–83. ACL.

[Cruz et al., 2018] Cruz, A. F., Rocha, G., and Cardoso, H. L. (2018). Exploring spanish corpora for portuguese coreference resolution. In [DBL, 2018], pages 290–295.

[Firth, 1957] Firth, J. R. (1957). *A synopsis of linguistic theory 1930–1955.* Longman, London.

[Gantz and Rydning, 2017] Gantz, D. R. J. and Rydning, J. (2017). Data age 2025: The evolution of data to life critical don't focus on big data; focus on the data that's big. https://blog.seagate.com/business/data-becomes-life-critical-a-fundamental-change-transforming-the-way-we-live/. Accessed: 09-09-2019.

[Hotho et al., 2003] Hotho, A., Staab, S., and Stumme, G. (2003). Ontologies improve text document clustering. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA*, pages 541–544. IEEE Computer Society.

[Hüllermeier et al., 2008] Hüllermeier, E., Fürnkranz, J., Cheng, W., and Brinker, K. (2008). Label ranking by learning pairwise preferences. *Artif. Intell.*, 172(16-17):1897–1916.

[Kaufman and Rousseeuw, 1990] Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley & Sons.

[Lewis, 1997] Lewis, D. D. (1997). Reuters-21578 text categorization test collection distribution 1.0. http://www.daviddlewis.com/resources/testcollections/reuters21578/.

[Monticolo et al., 2014] Monticolo, D., Mihaita, S., Darwich, H., and Hilaire, V. (2014). An agent-based system to build project memories during engineering projects. *Knowl.-Based Syst.*, 68:88–102.

[Newman and Girvan, 2004] Newman, M. E. J. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113.

[Rattinger et al., 2018] Rattinger, A., Goff, J. L., Meersman, R., and Guetl, C. (2018). Semantic and topological patent graphs: Analysis of retrieval and community structure. In [DBL, 2018], pages 51–58.

[Read, 2008] Read, J. (2008). A pruned problem transformation method for multi-label classification. In *In: Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS*, pages 143–150.

[Rifkin and Klautau, 2004] Rifkin, R. M. and Klautau, A. (2004). In defense of one-vs-all classification. *J. Mach. Learn. Res.*, 5:101–141.

[Rodriguez et al., 2005] Rodriguez, S., Hilaire, V., and Koukam, A. (2005). Holonic modeling of environments for situated multi-agent systems. In Weyns, D., Parunak, H. V. D., and Michel, F., editors, *Environments for Multi-Agent Systems II, Second International Workshop, E4MAS 2005, Utrecht, The Netherlands, July 25, 2005,* *Selected Revised and Invited Papers*, volume 3830 of *Lecture Notes in Computer Science*, pages 18–31. Springer.

[Romeo et al., 2014] Romeo, S., Tagarelli, A., and Ienco, D. (2014). Semantic-based multilingual document clustering via tensor modeling. In Moschitti, A., Pang, B., and Daelemans, W., editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 600–609. ACL.

[Roul, 2018] Roul, R. K. (2018). An effective approach for semantic-based clustering and topic-based ranking of web documents. *Int. J. Data Sci. Anal.*, 5(4):269–284.

[Rousseeuw, 1987] Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53 – 65.

[Sellah and Hilaire, 2018a] Sellah, S. and Hilaire, V. (2018a). Automatic generation of ontologies: Comparison of words clustering approaches. *IADIS*, pages 269–276.

[Sellah and Hilaire, 2018b] Sellah, S. and Hilaire, V. (2018b). A document clustering approach for automatic building of ontologies. In [DBL, 2018], pages 220–225.

[Smadi and Qawasmeh, 2018] Smadi, M. and Qawasmeh, O. (2018). A supervised machine learning approach for events extraction out of arabic tweets. In [DBL, 2018], pages 114–119.

[Staab and Hotho, 2003] Staab, S. and Hotho, A. (2003). Ontology-based text document clustering. In Klopotek, M. A., Wierzchon, S. T., and Trojanowski, K., editors, *Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM'03 Conference held in Zakopane, Poland, June 2-5, 2003*, Advances in Soft Computing, pages 451–452. Springer.

[Terra and Clarke, 2003] Terra, E. L. and Clarke, C. L. A. (2003). Frequency estimates for statistical word similarity measures. In Hearst, M. A. and Ostendorf, M., editors, *Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL 2003, Edmonton, Canada, May 27 - June 1, 2003*. The Association for Computational Linguistics.

[Tsoumakas and Katakis, 2007] Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *IJDWM*, 3(3):1–13.

[Tsoumakas et al., 2010] Tsoumakas, G., Katakis, I., and Vlahavas, I. P. (2010). Mining multi-label data. In Maimon, O. and Rokach, L., editors, *Data Mining and Knowledge Discovery Handbook, 2nd ed.*, pages 667–685. Springer.

[Tsoumakas and Vlahavas, 2007] Tsoumakas, G. and Vlahavas, I. P. (2007). Random *k* -labelsets: An ensemble method for multilabel classification. In Kok, J. N., Koronacki, J., de Mántaras, R. L., Matwin, S., Mladenic, D., and Skowron, A., editors, *Machine Learning: ECML 2007, 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007, Proceedings*, volume 4701 of *Lecture Notes in Computer Science*, pages 406–417. Springer.

[Wang and Koopman, 2017] Wang, S. and Koopman, R. (2017). Clustering articles based on semantic similarity. *Scientometrics*, 111(2):1017–1031.