# Topic Models and Fusion Methods: a Union to Improve Text Clustering and Cluster Labeling

Mohsen Pourvali[1]*, Salvatore Orlando[1], Hosna Omidvarborna[2]

[1] Università Ca' Foscari Venezia, Venezia (Italy)
[2] Politecnico di Torino, Torino (Italy)

Received 4 June 2018 | Accepted 14 November 2018 | Published 14 December 2018

LA UNIVERSIDAD
EN INTERNET

## Abstract

Topic modeling algorithms are statistical methods that aim to discover the topics running through the text documents. Using topic models in machine learning and text mining is popular due to its applicability in inferring the latent topic structure of a corpus. In this paper, we represent an enriching document approach, using state-of-the-art topic models and data fusion methods, to enrich documents of a collection with the aim of improving the quality of text clustering and cluster labeling. We propose a bi-vector space model in which every document of the corpus is represented by two vectors: one is generated based on the fusion-based topic modeling approach, and one simply is the traditional vector model. Our experiments on various datasets show that using a combination of topic modeling and fusion methods to create documents' vectors can significantly improve the quality of the results in clustering the documents.

Keywords

Document Enriching,
Document Clustering,
Cluster Labeling,
Text Mining.

DOI: 10.9781/ijimai.2018.12.007

## I. Introduction

While we are overwhelming by the increasing amount of available texts, we simply do not have the human power to read and study them to provide browsing and organizing experience over such the huge amount of texts. To this end, machine learning researchers have developed probabilistic topic modeling, a suite of algorithms that aim to discover and annotate large archives of documents with thematic information. Topic modeling algorithms are statistical methods which are able to find the themes (topics) running through the text documents by analyzing their words. Using topic models in machine learning and text mining is popular due to its applicability in inferring the latent topic structure of a corpus. In document clustering, a topic model could be directly used to map the original high-dimensional representation of documents (word features) to a low dimensional representation (topic features) and then apply a standard clustering algorithm like k-means in the new feature space, or we can consider each topic as a feature of a document, thus documents with highest proportion of same topic (same feature) are located in the same cluster [1]. Specifically, in the classification problem, topic models can be interpreted as the soft (or fuzzy) classification of the collection of documents into latent classes, which means a document does not belong fully to one class but it has different degrees of membership in several classes. Besides, the results of topic models could be used to produce a hard classifier in which a document can only have one and only one category.

In this work, we present a novel approach to improve the quality of clustering using topic models [2] and fusion methods [3]. The core idea of our approach is to enrich the vectors of the documents in order to improve the quality of clustering. To this end, we apply a statistical approach to discover and annotate a corpus with thematic information represented in form of different proportions over different topics for each document. Our approach is an unsupervised method and the topics, used for enriching, are produced by the unsupervised learning method. Further, final enriched vectors, representing documents, are clustered through kmeans clustering and produced classes are hard classification classes extracted from the Latent Dirichlet Allocation (LDA) results.

We first run topic modeling several times with different parameters over the collection, we then specify a set of topics in each iteration as the special topics for each document. Finally, we combine all the special topics in each iteration to generate a single topic for every document. These generated topics are indeed the vectors which are used later in the clustering of the collection. Furthermore, we use these topics to generate labels for each cluster.

## II. Related Works

In this section, we briefly summarize related works on text representation models for vector-word based text clustering.

The basic text representation model, i.e., Bag of Words (BOW) model, is widely used for text clustering and classification. In this model each term is weighted by various schemes such as TF, TF-IDF [4], and its variants [5]. Using BOW representation is popular but in the short text it generates a sparse vector for the document.

To overcome data sparseness, there are several works that exploit external knowledge (e.g., Wikipedia, WordNet, etc) to extend content of the documents. Banerjee et al. [6] uses Wikipedia knowledge base to enrich document representation vector with additional features, and Hotho et al. [7] uses WordNet knowledge base to enrich the representation vectors. There are some works that use feature selection approaches to reduce the high dimensionality. Revanasiddappa et al. [8] proposed a feature selection method based on Intuitionistic Fuzzy Entropy for text categorization.

* Corresponding author.

E-mail address: pourvali.mohsen@gmail.com

Lu et al. in [1] investigated performance of two probabilistic topic models Probabilistic Latent Semantic Analysis (PLSA) and LDA in document clustering. Authors used the topic models to generate a number1 of topics which are treated as specific features of documents. Therefore, for clustering, documents that have highest probability in a same feature (same topic) are clustered into the same cluster. In a similar way, Yau et al. [9] aims to elaborate on the ability of further other topic modeling algorithms Correlated Topic Model (CTM), Hierarchical LDA, and Hierarchical Dirichlet Process (HDP) to cluster documents. We highlight two main problems here: first, we do not know the exact number of topics running through the corpus, besides, because of frequency-based nature of topic models, we cannot claim the topic with the highest probability for a document is the main topic by which the documents must be clustered. These two problems are considered as our hypothesis in dealing with topics running through the corpus.

The supervised approaches in text classification domain [10, 11, 12] exploit topic models to enrich document representation. Vo and Ock [11] used the LDA model for topic analysis but presented new methods for enhancing features by combining external texts modeled from various types of universal datasets. In other studies [10, 12] their authors propose an approach to learn word vectors together with topics.

There are also some neural embedding methods word2vec [13] and doc2vec [14] that produce vector representations of words and documents by processing a corpus. Word2vec is a two-layer network with the main assumption that words with similar contexts have similar meaning. According to this assumption, word2vec describes semantic correlations between words in the corpus. Doc2vec (or Paragraph Vectors) is an extension of word2vec that requires labels to associate arbitrary documents with the labels. Indeed, Doc2vec learns to correlate labels and words rather than words with other words. These algorithms prefer to describe real semantic information embedded in words, sentences and documents rather than statistical relationships of the term occurrences.

In this paper, we propose a method to enrich document representation vectors to be used in partitional text clustering and cluster labeling. Our method is an unsupervised approach, needless of any external knowledge, with the aim of overcoming the two main problems about sparse vector and traditional LDA representation explained above. Since the main goal of our method is to enrich document vectors according to the statistical relationships of the term occurrences rather than real semantic information embedded in terms, we compared our results with two strong baselines in this domain. To this end, we use two unsupervised baselines: *first baseline*, i.e., BOW text representation with TF-IDF terms weighting, and *second baseline*, i.e., unsupervised usage of LDA in document representation [1, 9].

To the best of our knowledge, our work is the first to suggest a topic modeling solution to improve the quality of clustering and to perform cluster labeling based on the fusion methods.

## III. Preliminary

Before we explain the main approach proposed in this paper, we briefly describe topic models and explain LDA as the topic model that we apply in our approach. We also explain two well-known data fusion methods which are used in this paper.

### A. Topic Models

Topic models are based on the idea that documents are created by a mixture of topics, where a topic is a probability distribution over words. Specifically, a topic model is a statistical model by which we can create all the documents of a collection. Assume that we want to fill up every document of a corpus with the words, topic model says each document contains multiple topics and exhibits the topics in different proportion. Thus, for each document, there is a distribution over topics that according to this distribution, a topic is chosen for every word of that document, and then from that topic (i.e. distribution over vocabulary) a word is drawn [2].

### B. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a topic model widely used in the information retrieval field. Specifically, LDA is a probabilistic model that says each document of a corpus is generated by a distribution over topics, and each topic is characterized by a distribution over words. The process of generating a document defines a joint probability distribution over both observed (i.e. words of corpus) and hidden (i.e. topics) random variables. The data analysis is performed by using that joint distribution to compute the conditional distribution of the hidden variables given the observed variables. Formally, LDA is described as follows:

$$p(\beta_{1:k}, \theta_{1:D}, z_{1:D}, w_{1:D}) = \prod_{i=1}^{k} p(\beta_i) \prod_{d=1}^{D} p(\theta_d) \left( \prod_{n=1}^{N} p\left(z_{d,n} \middle| \theta_d\right) p\left(w_{d,n} \middle| \beta_{1:K}, z_{d,n}\right) \right) \quad (1)$$

where $\beta_{1:k}$ are topics where each $\beta_k$ is a distribution over words of the corpus (i.e. vocabulary), $\theta_{1:D}$ are topic proportions for the $d$th document, $z_d$ are the topic assignments for the $d$th document where $z_{d,n}$ is the topic assignment for the $n$th word in document $d$, which specifies the topic that $n$th word in $d$ belongs to, and $w_d$ are the observed words for document $d$ where $w_{d,n}$ is the $n$th word in document $d$.

### C. Fusion Methods

We now introduce two baseline state-of-the-art data fusion methods, frequently used for various information retrieval tasks, namely the CombSUM and CombMNZ fusion methods [3].

Suppose there are $n$ ranked lists which are created by $n$ different systems over a collection of items D. Each system $S_i$ provides a ranked list of items $L_i = < d_{i1}, d_{i2}, \ldots, d_{im} >$ and a relevance score $s_i(d_{ij})$ is assigned to each of the items in the list. Data fusion techniques use some algorithms to merge these n ranked lists into one [3].

CombSUM uses the following equation:

$$g(d) = \sum_{i=1}^{n} s_i(d) \quad (2)$$

If $d$ does not appear in any $L_i$, a default score (e.g., 0) is assigned to it. According to the global score $g(d)$ the items can be ranked as a new list.

Another method CombMNZ uses the equation:

$$g(d) = m \times \sum_{i=1}^{n} s_i(d) \quad (3)$$

where $m$ is the number of lists in which item $d$ appears.

The linear combination (i.e. general form of CombSUM) uses the equation:

$$g(d) = \sum_{i=1}^{n} w_i \times s_i(d) \quad (4)$$

where $w_i$ is the weight assigned to system $S_i$.

## IV. Our Method

To create an enriched vectorial representation for documents of a corpus, we propose an unsupervised technique, called Fusion- and Topic-based Enriching (FT-Enrich). Let $\mathbb{D} = \{d_1, d_2, \ldots, d_n\}$ be the collection of documents that we wish to be clustered, we run LDA algorithm several times over the collection, every time with different specified number of topics. We used LDA because we want to manually specify and change the number of topics. The intuition behind using different topics in each iteration is to bring in variety of topics being discussed in documents with an ensemble approach. We start with a number of topics close to the number of clusters, for example, assuming $K$ is the number of clusters we wish to have, the beginning number for topics is $I = K \pm \kappa$ where κ is a small integer[1]. The reason of starting with $I$ is to emphasize the topics in an iteration which has a number of topics close to the number of clusters. Finally, for every document $d_i$ of $\mathbb{D}$ there is a set $\mathbb{B} = \{\mathcal{B}_1, \mathcal{B}_2, \dots, \mathcal{B}_m\}$ where $\mathcal{B}_i = \{\beta_1, \beta_2, \dots, \beta_s\}$ shows $s$ topics belonging to iteration $i$, and $m$ indicates the number of iterations. At first $s = I$ which is increased by one in each iteration. Number of iterations depends on the maximum number of topics, i.e., bigger than number of clusters, involving in determining of special topics. It could be an expectation of different topics among the corpus. The clustering results in our experiments are obtained by 25 iterations. Therefore, for clustering a corpus into 4 clusters, sequence of the topics number for 25 iterations with $I = 4 - 1$ is 3,4,5, … ,27.

In every iteration, for each document, we generate a set of topics, namely, special topics, which are selected from the topics within iteration $i$. To generate these topics, we construct a graph $G_i$ comprising the documents of $\mathbb{D}$ and the topics generated in iteration $i$. Fig. 1 shows three examples of graph $G$ in different iterations. Every circular node corresponds to a document of the collection, and the square nodes correspond to the topics generated in that iteration. The connection $\mathcal{X}_{jr}$ between a circular node $d_i$ and a square node $\beta_r$ indicates the proportion of the corresponding topic in the document. Therefore, $\mathbb{P}_i = \{\theta_{1:s}, \theta_{2:s}, \dots, \theta_{n:s}\}$ indicates topic proportions of the documents in iteration $i$ where $\theta_j = \{\mathcal{X}_{j1}, \mathcal{X}_{j2}, \dots, \mathcal{X}_{js}\}$ shows topic proportions for document $j$ in graph $G_i$ where $\sum_{l=1}^{s} \mathcal{X}_{jl} = 1$. Therefore, the elements of special topics for document $d_j$, within iteration $i$, include:

- the topic with highest proportion of $\mathcal{X}_{jx}$ for document $d_j$,
- the topic by which document $d_j$ finds its best couple,
- the topic by which $d_j$ is selected as the best couple for a document.

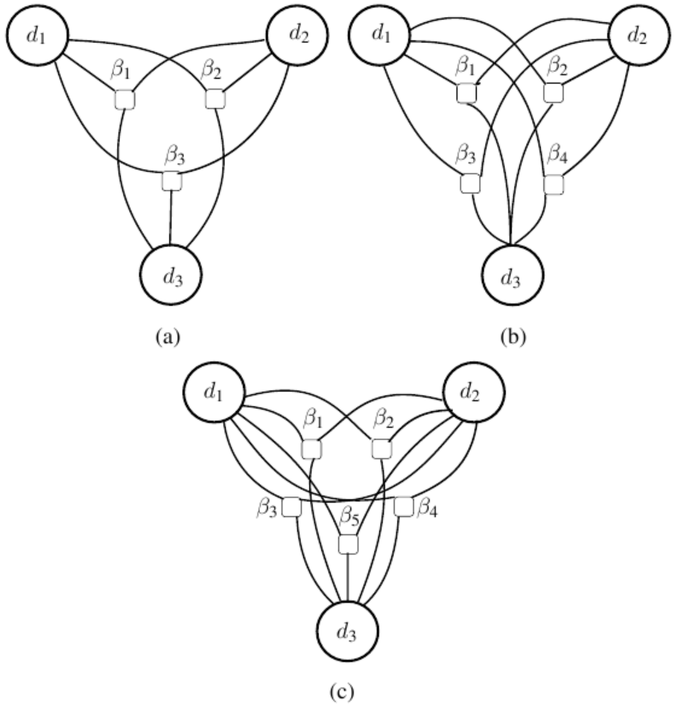


Fig. 1. Three typical graphs of $G$ for $\mathbb{D} = \{d_1, d_2, d_3\}$ in three different iterations with: (a) three topics; (b) four topics; (c) five topics.

Given the topics of iteration $i$th, the best couple for document $d_j$ is a document $d_k$ for which the following equation returns the highest value:

$$Couple(d_j, d_k | \mathcal{B}_i) = \arg \max_{\beta_l \in \mathcal{B}_i} \left( \frac{\mathcal{X}_{jl} \times \mathcal{X}_{kl}}{\mathcal{X}_{jl} - \mathcal{X}_{kl}} \right) \tag{5}$$

where the denominator in case of $\mathcal{X}_{jl} = \mathcal{X}_{kl}$ equals 0.1. Specifically, Equation (5) is to find documents which are similar together in a specific topic, considering their proportion in the topic. Therefore, for each document in a specific iteration, there is a special topics set $ST_i(d_j)$ where $|ST_i| \leq |\mathcal{B}_i|$. We take into account the effect of special topics for each document by combining elements of $ST_i(d_i)$. Our goal is to generate a representing vector for each document to be used in clustering where this vector is a combination of some special topics. We use the data fusion method CombSUM in two phases to generate a single topic (vector) for each document in the corpus.

In the first phase, all the topics within $ST_i(d_i)$ are combined to generate a single vector $\mathcal{V}_{ij}$ for each document $d_j$ in iteration $i$. Formally, let $S_{\beta}^{norm}(b|\mathcal{B}_i)$ denotes $b$'s **normalized** score given in distribution (topic) $\beta$, the general form of CombSUM fusion method then simply sums over the normalized $b$'scores given by various topics in $ST_i(d_j)$.

$$CombSum\left(b \middle| ST_i(d_j)\right) = \sum_{\beta \in ST_i} \mathcal{X}_{j\beta} \times S_{\beta}^{norm}(b|\mathcal{B}_i) \tag{6}$$

where $\mathcal{X}_{j\beta}$ is the proportion of document $j$ in topic $\beta$.

In the second phase, all the single vectors $\mathcal{V}_{ij}$ generated in $m$ iterations are combined to generate a **unique** vector $V_j$ for document $j$. Formally, given $AV(d_j) = \{\mathcal{V}_{1j}, \mathcal{V}_{2j}, \dots, \mathcal{V}_{mj}\}$, let $S_{\mathcal{V}}^{norm}(b|\mathbb{B})$ denotes $b$'s normalized score given in vector $\mathcal{V}$, therefore, the CombSUM fusion method sums over the normalized $b$'s scores given by various vectors in $AV(d_j)$.

$$CombSum\left(b \middle| AV(d_j)\right) = \sum_{\mathcal{V} \in AV} S_{\mathcal{V}}^{norm}(b|\mathbb{B}) \tag{7}$$

Finally, a trade-off between $V_j$ and traditional vector, i.e., a vector generated based on TF-IDF for document $j$, are used to generate the final vector. Which is the representing vector for $j$th document in clustering. Formally:

$$FV_j^{norm} = \alpha \times V_j^{norm} + (1 - \alpha) \times vt_j \tag{8}$$

where $vt_j$ indicates traditional vector for $j$th document, and $\alpha \in [0,1]$.

## V. Cluster Labelling

To label a cluster $C = \{FV_1, FV_2, \dots, FV_c\}$, we use CombMNZ data fusion method which provides good results in combining several ranked lists [3][15]. First, we create $\mathcal{L} = \{L_1, L_2, \dots, L_c\}$ where $L_j$ is a list of terms corresponding to the vector $FV_j$ within $C$, we then rank/sort the terms of $L_j$ based on the scores/probabilities obtained for its corresponding vector $FV_j$ in Equation (8). Therefore, $\mathcal{L}$ is updated with the new ranked lists. We then create candidate labels $L^{[M]}{}_j(C)$ which are Top-M terms within list $L_j$. Therefore, let $\mathcal{L}^{[M]}(C) = \bigcup_{L \in \mathcal{L}} L^{[M]}(C)$ denotes the overall candidate-labels pool which are generated based on the union of all Top-M scored labels selected from $L \in \mathcal{L}$ for cluster $C$. The CombMNZ is to boost label $l$ based on the number of times that $l$ appears in various lists. Formally:

$$CombMNZ\left(l \middle| \mathcal{L}^{[M]}(C)\right) = \#\{l \in L^{[M]}(C)\} \times \sum_{L \in \mathcal{L}} S_L^{norm}(l|C) \tag{9}$$

Finally, Top-N, i.e., $|N| < |M|$, labels of the combination result are selected as the labels of the cluster $C$.

[1] In our experiments $\kappa = 1$

## VI. Experimental Setup

The principal idea of the experiments is to show the efficacy of an ensemble approach of topic modeling on clustering results through a manually predefined categorization of the corpus.

### A. Datasets

We explore the utility of using representation vectors of documents generated by our method in addition to label the clusters. To this end, we used three different datasets:

**Classic4**: This dataset is often used as a benchmark for clustering and co-clustering[2]. It consists of 7095 documents classified into four classes denoted MED, CISI, CRAN and CACM. For our experiments, we extract randomly 500 documents from each class.

**BBC NEWS**: This dataset consists of 2225 documents from the BBC news website corresponding to stories in five topical areas, which are named Business, Entertainment, Politics, Sport and Tech, from 2004-2005 [18].

**20NG**: 20 News Group[3] (20NG) is a collection of documents manually classified into 20 different categories that each one contains about 1000 documents.

### B. Preprocessing

Preprocessing is an essential step in text mining. The first classical preprocessing regards stop words removal and lower case conversion. In addition, we used L2-norm to normalize the topics/vectors generated by MALLET. The normalized vector of $v = (v_1, v_2, .., v_n)$ is a vector with the same direction but with length one. It is denoted by $\hat{v} = \frac{v}{|v|}$, where $|v| = \sqrt{v_1^2 + v_2^2 + .. + v_n^2}$.

### C. Vectors Similarity Measure

For evaluating similarity of two represented vectors, we used comparative traditional measure Cosine Similarity that measures the cosine of the angle between two none zero vectors of an inner product space. Given two vectors of attributes, $A$ and $B$, the cosine similarity, $\cos(\theta)$, is represented as follows:

$$Similarity = \cos(\theta) = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{10}$$

Cosine similarity is a judgment of orientation and not magnitude of two vectors commonly used with text data represented by word counts: its results range from -1 meaning exactly opposite, to 1 meaning exactly the same, with 0 indicating orthogonality, and in-between values indicating intermediate similarity or dissimilarity. In Information Retrieval (IR), since the term frequencies (TF-IDF weights) cannot be negative, the cosine similarity of two represented vectors of two documents will range from 0 to 1.

### D. Clustering Evaluation Measures

We used two external criteria Purity and F1-measure for evaluating the clustering results.

**Purity**: The purity is a simple and transparent evaluation measure which is related to the entropy concept [19]. To compute the purity criterion, each cluster $C$ is assigned to its majority class. Then we consider the percentage of correctly assigned documents, given the set of documents $L_i$ in the majority class:

$$Precision(C, L_i) = \frac{|C \cap L_i|}{|C|} \tag{11}$$

The final purity of the overall clustering is defined as follows:

$$Purity(\mathbb{C}, \mathbb{L}) = \sum_{C_j \in \mathbb{C}} \frac{|C_j|}{N} arg \max_{L_i \in \mathbb{L}} Precision(C_j, L_i) \tag{12}$$

where $N$ is the number of all documents, $\mathbb{C} = \{C_1, C_2, \dots, C_k\}$ is the set of clusters and $\mathbb{L} = \{L_1, L_2, \dots, L_c\}$ is the set of classes.

**F1-measure**: The F1-measure is defined as a harmonic mean of precision $P$ and recall $R$ [20]. Formally, F1-measure is defined as follows:

$$F_1 = \frac{2PR}{P+R} \tag{13}$$

where $P$ (Precision) is defined in Equation (11), and $R$ (Recall) is formally defined as follows:

$$Recall(C, L_i) = \frac{|C \cap L_i|}{|L_i|} \tag{14}$$

where $L_i$ is the majority class.

### E. Labelling Evaluation Measures

For evaluating the quality of cluster labeling, we use the frameworks represented in [21]. Therefore, for each given cluster, its ground truth labels where obtained by manual (human) labeling and are used for the evaluation.

We use **Match@N** (Match at top N results) and **MRR@N** (Mean Reciprocal Rank) measures proposed in [21] to evaluate the quality of the labels. They consider the categories of Open Directory Project (ODP) as the correct labels and then evaluate a ranked list of proposed labels by using the following criteria:

- **Match@N**: It is a binary indicator, and returns 1 if the top N proposed labels contain at least one correct label. Otherwise it returns zero.
- **MRR@N**: It returns the inverse of the rank of the first correct label in the top-N list. Otherwise it returns zero.

A proposed label for a given cluster is considered correct if it is identical, an inflection, or a WordNet synonym of the cluster's correct label [16].

## VII. Experimental Results

### A. Evaluating Results of Clustering

In our experiments, we use the software package CLUTO[4] which is used for clustering low- and high-dimensional datasets. The algorithm adopted for clustering is Partitional, and the measure of the similarity between two vectors is Cosine similarity. Every document of the corpus is represented by two vectors: one is generated based on FT-Enrich method, and one simply is the traditional vector (BOW)–classical TF-IDF weighting of terms–model.

We tested and evaluated clustering with/without applying FT-Enrich, to show the improvements in clustering purity due to a capable combination of fusion and topic modeling approaches. The obtained results of such improvement are shown in Table II and Table IV on two various datasets BBC and Classic[4]. The obtained results in Table II on BBC indicate that representing documents by only using FT-Enrich ( $\alpha = 1$) considerably improve the quality of clustering compared to using traditional TF-IDF method (*first baseline*) shown in Table I. We can see in Table II the best improvement in total purity (%22) and average of F1-measures (%23) are obtained by entirely using FT-Enrich method (α=1). Furthermore, in Table I and Table II, it can be observed in cluster 4 we have about %50 improvement in purity of the cluster.

[2] http://www.dataminingresearch.com/index.php/2010/09/classic3-classic4-datasets/

[3] http://qwone.com/~jason/20Newsgroups/

[4] http://glaros.dtc.umn.edu/gkhome/views/cluto

TABLE I. Clustering Results of Dataset BBC Using Traditional Document Representations (First Baseline) ($\alpha = 0$)

| Cluster | Bus | Enter | Polit | Sport | Tech | F1 | Purity |
|---|---|---|---|---|---|---|---|
| Cluster 0 | 58 | 6 | 254 | 5 | 11 | 0.676 | 0.760 |
| Cluster 1 | 320 | 2 | 15 | 4 | 5 | 0.748 | 0.925 |
| Cluster 2 | 79 | 24 | 52 | 7 | 344 | 0.759 | 0.680 |
| Cluster 3 | 30 | 16 | 15 | 441 | 5 | 0.866 | 0.870 |
| Cluster 4 | 23 | 338 | 81 | 54 | 36 | 0.736 | 0.635 |
| Total Purity | | | | | | | **0.763** |

TABLE II. Clustering Results of Dataset BBC Using Ft-Enrich Method ($\alpha = 1$)

| Cluster | Bus | Enter | Polit | Sport | Tech | F1 | Purity |
|---|---|---|---|---|---|---|---|
| Cluster 0 | 21 | 21 | 401 | 27 | 13 | 0.891 | 0.830 |
| Cluster 1 | 473 | 5 | 9 | 1 | 8 | 0.940 | 0.954 |
| Cluster 2 | 13 | 6 | 3 | 0 | 364 | 0.925 | 0.943 |
| Cluster 3 | 1 | 0 | 1 | 482 | 4 | 0.961 | 0.980 |
| Cluster 4 | 2 | 354 | 3 | 1 | 12 | 0.934 | 0.952 |
| Total Purity | | | | | | | **0.932** |

We investigated the variation of $\alpha$ by considering the amount of dispersion of documents' sizes. Our experiments show that contribution of FT-Enrich method in creating the representation vectors for corpus with low Standard Deviation (SD) with respect to its mean (ME) is major compared to the one with the high SD. Table IV shows the clustering result with $\alpha = 0.1$ on Classic4 for which $SD = 143.34$ and $ME = 158.47$, but on the other hand, the clustering result shown in Table II is obtained by $\alpha = 1$ for which $SD = 123.64, ME = 341.21$.

We also compared our method with the *second baseline*, i.e., unsupervised LDA document representation. To this end, we considered the number of topics for each dataset corpus is equal to the number of classes manually specified for the corpus. For example, for dataset BBC with 5 manually specified classes, we ran LDA topic modeling with 5 topics over the corpus. Therefore, each document of BBC news corpus is represented by 5 different representation vectors/topics. Finally, documents that have highest probability/proportion in a same topic are clustered into the same cluster. The results of the clustering are shown in Table V and Table VI. As it can be observed, clustering using LDA representation alone returns worse result on dataset Classic4 compared to the results obtained by using traditional TF-IDF method (*first baseline*) shown in Table III.

TABLE III. Clustering Results of Dataset Classic4 Using Traditional Document Representations (First Baseline) ($\alpha = 0$)

| Cluster | Cacm | Cisi | Cran | Med | F1 | Purity |
|---|---|---|---|---|---|---|
| Cluster 0 | 323 | 30 | 11 | 21 | 0.730 | 0.839 |
| Cluster 1 | 55 | 17 | 479 | 0 | 0.911 | 0.869 |
| Cluster 2 | 47 | 6 | 4 | 454 | 0.898 | 0.888 |
| Cluster 3 | 75 | 447 | 6 | 25 | 0.849 | 0.808 |
| Total Purity | | | | | | **0.852** |

TABLE IV. Clustering Results of Dataset Classic4 Using Ft-Enrich Method ($\alpha$ =0.1)

| Cluster | Cacm | Cisi | Cran | Med | F1 | Purity |
|---|---|---|---|---|---|---|
| Cluster 0 | 334 | 9 | 2 | 0 | 0.790 | 0.968 |
| Cluster 1 | 71 | 0 | 485 | 0 | 0.918 | 0.872 |
| Cluster 2 | 43 | 0 | 6 | 485 | 0.938 | 0.908 |
| Cluster 3 | 49 | 491 | 7 | 15 | 0.925 | 0.874 |
| Total Purity | | | | | | **0.898** |

TABLE V. Clustering Results by Grouping Documents which Have a Same Topic with Highest Probability (Second Baseline) on the BBC

| Cluster | Bus | Enter | Polit | Sport | Tech | F1 | Purity |
|---|---|---|---|---|---|---|---|
| Cluster 0 | 5 | 12 | 0 | 70 | 285 | 0.737 | 0.766 |
| Cluster 1 | 0 | 348 | 6 | 180 | 5 | 0.753 | 0.646 |
| Cluster 2 | 462 | 9 | 17 | 0 | 10 | 0.917 | 0.928 |
| Cluster 3 | 18 | 12 | 375 | 11 | 10 | 0.889 | 0.880 |
| Cluster 4 | 25 | 5 | 19 | 250 | 91 | 0.555 | 0.641 |
| Total Purity | | | | | | | **0.773** |

TABLE VI. Clustering Results by Grouping Documents which Have a Same Topic With Highest Probability (Second Baseline) on the Classic4

| Cluster | Cacm | Cisi | Cran | Med | F1 | Purity |
|---|---|---|---|---|---|---|
| Cluster 0 | 211 | 386 | 11 | 10 | 0.691 | 0.625 |
| Cluster 1 | 258 | 59 | 0 | 128 | 0.546 | 0.580 |
| Cluster 2 | 25 | 8 | 484 | 0 | 0.952 | 0.936 |
| Cluster 3 | 6 | 47 | 5 | 362 | 0.787 | 0.862 |
| Total Purity | | | | | | **0.745** |

### B. Evaluating Results of Cluster Labeling

We use 20NG benchmark for our experiments in cluster labeling. Therefore, we first show the result of clustering on this dataset using representation vectors generated by our method which indeed are used in cluster labeling. We further compare our result with the clustering result obtained by using the traditional representation vectors. The results of the clustering are shown in Table VII. It shows a remarkable improvement (%68) in the total purity of clustering (TP = 0.64) which leads to achieve significant result in cluster labeling as well.

The cluster labeling method represented in this work is a direct cluster labeling method in which the candidate labels for clusters are directly extracted from content of the clusters without using external sources (e.g. Wikipedia). One of the baseline direct approaches that several clustering systems apply for cluster labeling [17] is to select the top-n terms with maximal weights from the cluster centroid as the candidate labels. In our experiments we use this approach as a baseline for comparison. Specifically, we explore the effectiveness of using candidate labels generated by our approach in addition to the highest weighted terms extracted from cluster centroid provided by: TF-IDF and FT-Enrich method.

As an example of cluster labeling, Table VIII shows top-15 labels produced by the three above explained labeling methods over first cluster of 20News dataset which is labeled "Atheism" by experts. It can be observed in Table VIII that the labels produced by CombMNZ (topic-based) method are more describing a cluster of documents with subject Atheism than other methods. Specifically, first correct proposed label atheist (i.e. inflection for Atheism) is observed with $N = 7$ (Match@7=1, MRR@7=0.143) for CombMNZ (topic-based), whereas for Centroid (topic-based) with $N = 14$ (Match@14=1, MRR@14=0.071), and for Centroid (TF-IDF) with $N = 15$ (Match@15=1, MRR@15=0.067).

Fig. 2 reports on the Match@N and MRR@N scores of each method for increasing values of N. As it can be observed, using the highest weighted terms extracted from clusters' centroids provided by FT-Enrich method is more effective than the ones provided by TF-IDF. It further shows that using fusion method (CombMNZ(FT-Enrich)) on the representation vectors generated by FT-Enrich method provides the best performance for both label quality measures. We can further observe that, for the Match@N measure, baseline method with FT-Enrich based cluster centroid requires at list 18 terms to cover %80 of the clusters with a correct label, while the same effectiveness is

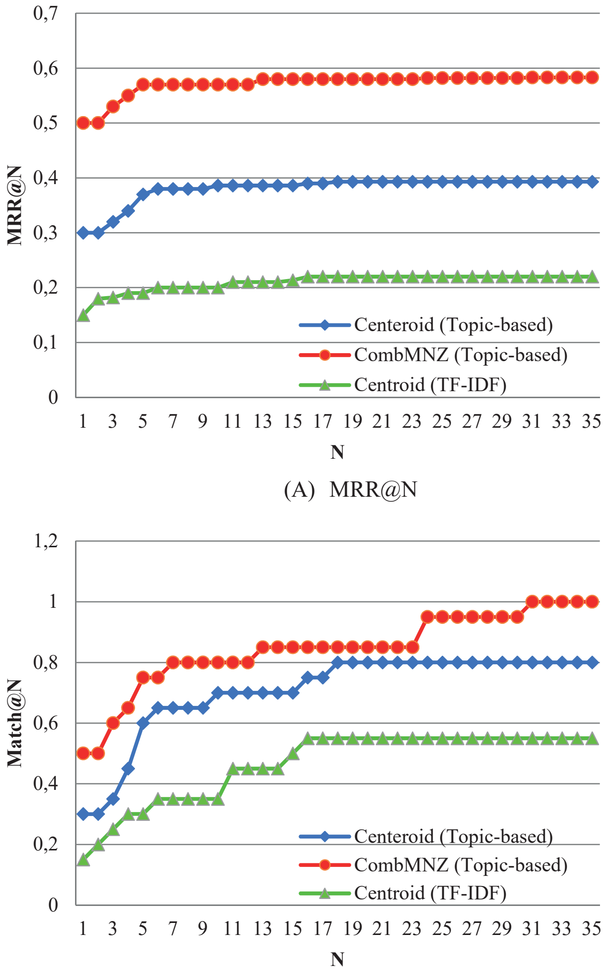
TABLE VII. Clustering Results by Grouping Documents which Have a Same Topic With Highest Probability (First Baseline) on the BBC, Using (**A**) Tf-Idf, and (**B**) Ft-Enrich Methods, Total Purity Indicated with **TP**

| | **Purity of Cluster** | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | TP |
| A | 0.25 | 0.39 | 0.16 | 0.5 | 0.21 | 0.3 | 0.46 | 0.48 | 0.25 | 0.3 | 0.16 | 0.29 | 0.71 | 0.24 | 0.84 | 0.35 | 0.41 | 0.4 | 0.71 | 0.45 | **0.38** |
| B | 1.0 | 0.96 | 0.89 | 0.99 | 0.96 | 0.64 | 0.63 | 0.43 | 0.23 | 0.42 | 0.28 | 0.47 | 0.95 | 0.59 | 0.88 | 0.95 | 0.94 | 0.94 | 0.35 | 0.81 | **0.64** |
| | **F1-measure** | | | | | | | | | | | | | | | | | | | | |
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | |
| A | 0.19 | 0.30 | 0.17 | 0.54 | 0.23 | 0.32 | 0.51 | 0.54 | 0.28 | 0.33 | 0.18 | 0.33 | 0.56 | 0.19 | 0.69 | 0.28 | 0.34 | 0.34 | 0.76 | 0.48 | |
| B | 0.54 | 0.69 | 0.46 | 0.87 | 0.91 | 0.46 | 0.71 | 0.41 | 0.33 | 0.28 | 0.36 | 0.43 | 0.55 | 0.67 | 0.89 | 0.68 | 0.88 | 0.88 | 0.49 | 0.79 | |

TABLE VIII. An example of Top-15 proposed labels, using three different methods; (**A**) Centroid (tf-idf), (**B**) Centroid (topic-based), and (**C**) CombMNZ (topic-based) over first cluster of 20News dataset "Atheism"

| | Labels | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| A | caltech | keith | solntze | livesey | Livesey | sgi | wpd | Schneider | Keith | nuclear | Allan | jon | mathew | Political | Atheist |
| B | system | moral | person | wrong | morality | objective | murder | keith | life | jon | society | innocent | god | atheist | human |
| C | god | moral | person | life | wrong | morality | **atheist** | objective | murder | human | evidence | society | keith | truth | jon |

achieved by a list of 7 terms only using FT-Enrich method. It is also interesting that with $N > 31$ CombMNZ (FT-Enrich) method covers %100 of the clusters with a correct label.



(A) MRR@N

(B) Match@N

Fig 2. Average (A) MRR@N and (B) Match@N values obtained for clusters of 20NG using fusion method over representation vectors generated by FT-Enrich, using top-N terms of cluster centroid weighted by FT-Enrich method, and using top-N terms of cluster centroid weighted by TF-IDF.

## VIII. Conclusion

In this paper, we presented a fusion- and topic-based enriching approach in order to improve the quality of clustering. We applied a statistical approach, namely topic model, to enrich the representation vectors of the documents. To this end, an ensemble topic modeling with using different parameters for each model are represented, and then, using a fusion approach, all the generated results are combined to provide a single vectorial representation for each document. Our experiments on the different datasets show significant improvement in clustering results. We further show that putting such representation vectors in a fusion method provides interesting results in cluster labeling as well.

As a future work, we plane to exploit external sources (e.g. WordNet) in both the clustering and cluster labeling to explore the effectiveness of using topic models as well as the resources in corresponding domains.

## References

[1] Yue Lu, Qiaozhu Mei and ChengXiang Zhai, Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA, Information Retrieval 14, pp. 178–203, April 2011.

[2] David M Blei, Probabilistic topic models, Communications of the ACM 55, pp. 77–84, April 2012.

[3] Shengli Wu, Data fusion in information retrieval, 13, Springer Science & Business Media, 2012.

[4] Youngjoong Ko, A study of term weighting schemes using class information for text classification, in: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 1029–1030, August 2012.

[5] Gerard Salton and Christopher Buckley, Term-weighting approaches in automatic text retrieval, Information processing & management 24, pp. 513–523, January 1988.

[6] Somnath Banerjee, Krishnan Ramanathan and Ajay Gupta, Clustering short texts using wikipedia, in: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 787–788, July 2007.

[7] Andreas Hotho, Steffen Staab and Gerd Stumme, Ontologies improve text document clustering, in: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, IEEE, pp. 541–544, November 2003.

[8] Revanasiddappa M. B. and Harish B. S., A new feature selection method based on intuitionistic fuzzy entropy to categorize text documents, International Journal of Interactive Multimedia and Artificial Intelligence 5(3), pp. 106-117, 2018.

[9] Chyi-Kwei Yau, Alan Porter, Nils Newman and Arho Suominen,

Clustering scientific documents with topic modeling, Scientometrics 100, pp. 767–786, September 2014.

[10] Yang Liu, Zhiyuan Liu, Tat-Seng Chua and Maosong Sun, Topical word embeddings, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI Press, pp. 2418–2424, January 2015.

[11] Duc-Thuan Vo and Cheol-Young Ock, Learning to classify short text from scientific documents using topic models with various types of knowledge, Expert Systems with Applications 42, pp. 1684–1698, February 2015.

[12] Heng Zhang and Guoqiang Zhong, Improving short text classification by learning vector representations of both words and hidden topics, Knowledge-Based Systems 102, pp. 76–86, June 2016.

[13] Tomas Mikolov, Kai Chen, Greg Corrado and Jeffrey Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv: pp. 1301.3781, January 2013.

[14] Quoc Le and Tomas Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, pp. 1188–1196, January 2014.

[15] Haggai Roitman, Shay Hummel and Michal Shmueli-Scheuer, A fusion approach to cluster labeling, in: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, ACM, pp. 883–886, July 2014.

[16] David Carmel, Haggai Roitman and Naama Zwerdling, Enhancing cluster labeling using wikipedia, in: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 139–146, July 2009.

[17] Douglass R Cutting, David R Karger, Jan O Pedersen and John W Tukey, Scatter/gather: A cluster-based approach to browsing large document collections, in: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 318–329, August 1992.

[18] Derek Greene and Pádraig Cunningham, Practical solutions to the problem of diagonal dominance in kernel document clustering, in: Proceedings of the 23rd international conference on Machine learning, ACM, pp. 377–384, June 2006.

[19] Jonathan A Silva, Elaine R Faria, Rodrigo C Barros, Eduardo R Hruschka, André CPLF de Carvalho and João Gama, Data stream clustering: A survey, ACM Computing Surveys (CSUR) 46, October 2013.

[20] Yutaka Sasaki et al., The truth of the F-measure, Teach Tutor mater 1, pp. 1–5, October 2007.

[21] Pucktada Treeratpituk and Jamie Callan, Automatically labeling hierarchical clusters, in: Proceedings of the 2006 international conference on Digital government research, Digital Government Society of North America, pp. 167–176, May 2006.

M. Pourvali

Experienced Lecturer with a demonstrated history of working in the research industry. Skilled in Word Sense Disambiguation, Text Clustering, Text Summarization, Document Enrichment, and generally in Natural Language Processing. Strong education professional with a Doctor of Philosophy (PhD) focused on Computer Science from Ca' Foscari University of Venice in Italy.

S. Orlando

MSc (1985) and PhD (1991), University of Pisa - is a full professor at Ca' Foscari University of Venice. His research interests include data and web mining, information retrieval, parallel/distributed systems. He published over 150 papers in peer reviewed international journals and conferences. He co-chaired conferences, tracks, and workshops, and served in the PC of many premier conferences.

H. Omidvarborna

She received her Bachelor Degree in Electrical Engineering from Razi University, and continued her studies in Computer and Communication Networks at Polytechnic University of Turin. Her research interests include Information retrieval and analysis and visualization of massive data.