CluWords: Exploiting Semantic Word Clusters for Enhanced Topic Modeling

Conference Paper · February 2019

CITATIONS
0

READS
318

6 authors, including:

Washington Cunha
Federal University of Minas Gerais
8 PUBLICATIONS **20** CITATIONS
SEE PROFILE

Felipe Viegas
Federal University of Minas Gerais
16 PUBLICATIONS **110** CITATIONS
SEE PROFILE

Sérgio Canuto
Federal University of Minas Gerais
20 PUBLICATIONS **125** CITATIONS
SEE PROFILE

Thierson Couto
Universidade Federal de Goiás
31 PUBLICATIONS **319** CITATIONS
SEE PROFILE

CluWords: Exploiting Semantic Word Clustering Representation for Enhanced Topic Modeling

Felipe Viegas
UFMG - Brazil
frviegas@dcc.ufmg.br

Sérgio Canuto
IFG - Brazil
sergio.canuto@ifg.edu.br

Christian Gomes
UFSJ - Brazil
christian@ufsj.edu.br

Washington Luiz
UFMG - Brazil
washingtoncunha@dcc.ufmg.br

Thierson Rosa
UFG - Brazil
thierson@inf.ufg.br

Sabir Ribas
SEEK - Melbourne, Australia
sribas@seek.com.au

Leonardo Rocha
UFSJ - Brazil
lcrocha@ufsj.edu.br

Marcos André Gonçalves
UFMG - Brazil
mgoncalv@dcc.ufmg.br

ABSTRACT

In this paper, we advance the state-of-the-art in topic modeling by means of a new document representation based on pre-trained word embeddings for non-probabilistic matrix factorization. Specifically, our strategy, called CluWords, exploits the nearest words of a given pre-trained word embedding to generate meta-words capable of enhancing the document representation, in terms of both, syntactic and semantic information. The novel contributions of our solution include: (i) the introduction of a novel data representation for topic modeling based on syntactic and semantic relationships derived from distances calculated within a pre-trained word embedding space and (ii) the proposal of a new TF-IDF-based strategy, particularly developed to weight the CluWords. In our extensive experimentation evaluation, covering 12 datasets and 8 state-of-the-art baselines, we exceed (with a few ties) in almost cases, with gains of more than 50% against the best baselines (achieving up to 80% against some runner-ups). Finally, we show that our method is able to improve document representation for the task of automatic text classification.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; *Natural language processing*; *Topic modeling*;

KEYWORDS

Data Representation, Topic Modeling, Word Embedding

ACM Reference Format:
Felipe Viegas, Sérgio Canuto, Christian Gomes, Washington Luiz, Thierson Rosa, Sabir Ribas, Leonardo Rocha, and Marcos André Gonçalves. 2019. CluWords: Exploiting Semantic Word Clustering Representation for Enhanced Topic Modeling. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3291032

WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3291032

1 INTRODUCTION

The intuition behind topic models is that each document is comprised of some topics or themes. A topic is understood as a collection of words that represent the topic as a whole. Thus, Topic Modeling is the machine learning task that extracts "implicit" topics from a collection of documents and assigns the most probable ones to each document [4].

Topic Modeling is an important research area, mainly when there is no explicit taxonomy or classification scheme to associate with documents or when such an association (a.k.a., labeling) is very cumbersome or costly to obtain. Important scenarios where Topic Modeling has been demonstrated to be very useful include (i) expansion of the representation for short documents, a problem very common on social computing applications [15, 26, 39], (ii) unsupervised tasks (e.g. clustering) [36] or (iii) supervised tasks (e.g., text and topic classification) [29].

Developments in this research line have exploited not only syntactical variations of the same word (stems) before the generation of the topics, but some types of *semantic similarity* have also been considered [19]. Such semantic similarities are usually computed by means of some type of distance among the words, for instance "hops" in a manually built semantic dictionary [12]. The problem with most of these dictionary-based approaches is that they are manually built, usually for a particular application, and are hard to scale up or to adapt or evolve to new contexts. More recently, some works have tried to automatize this task by exploiting semantic similarities by means of the computation of distances between words, most notably by exploiting their positioning within an embedding space [21, 27]. Such similarities have been previously correlated with *semantic closeness* [1, 18, 32].

A common important issue that is neglected by basically all of these embedding-based approaches, mainly in the context of Topic Modeling, is that, differently from manually-built semantic dictionaries, they do not explicitly consider the relationships between and among words in the whole document vocabulary. Most methods only consider general implicit relationships between words in the context of the documents [19, 32], not taking advantage of

the local information from semantic relationships between a word and its neighbors. Such neighborhood might be susceptible to the presence of noisy terms [23, 38] and therefore, should be carefully considered when exploiting new discriminative information.

In this context, in this paper we present a novel document representation based on pre-trained word embedding for non-probabilistic matrix factorization (NMF)[1] for Topic Modeling. Specifically, our strategy, called *CluWords (for Clusters of Words)*, exploits the nearest words of a given pre-trained word embedding to generate "meta-words" capable of enhancing the document representation, in terms of syntactic and semantic information. The explicit exploitation of similarity between word embeddings to find the nearest words provides fine-grained information about relationships between words. Our strategy combines both traditional syntactic evidence (from the occurrences of words in a document) and the similarity between a word and its neighbors by means of a parameter α that weights the contribution from these sources of evidence. Therefore, we expect to mitigate the potential drawbacks of using the projected space of word embeddings by exploiting only clear similarity evidence and relying on traditional syntactic document representations. Each word of the vocabulary corresponds to one CluWord, which is weighted according to new TF-IDF-based strategy, particularly developed to measure the importance a given CluWord to define a topic of a document. This novel representation is rich and flexible enough to be exploited by any type of Topic Modeling approach (see Section 3).

Our experiments demonstrate that our proposed strategy, when exploited together with NMF, is more robust and present less variability than the state-of-the-art representation for Topic Modeling that also uses NMF and word embeddings – SeaNMF [32]. While SeaNMF tries to obtain generalized evidence about topics from the context of documents by means of matrix factorization, our approach focus on the fine-grained and high-quality information exploited from the similarity between a word and its neighbors. Moreover, instead of a disjoint use of frequencies (syntactic evidence) and word embeddings (implicit semantic evidence) as proposed in SeaNMF, we propose a combined strategy that weights the CluWords with an adaptation of the traditional TF-IDF weighting scheme. The experiments also demonstrate that our strategy outperforms SeaNMF when the discovered topics are explored in an important application: supervised document classification.

In sum our main contributions are:

(1) a novel document representation (CluWords) that exploits, into a unified framework, semantic and syntactic relationships among words in a document collection with the goal of enhancing non-probabilistic Topic Modeling;
(2) the proposal of a new TF-IDF weighting strategy for the CluWords;
(3) an extensive experimental evaluation covering 12 datasets and 8 state-of-the-art baselines, in which our approach excels (with a few ties) in almost cases, with gains of more than 50% against the best baselines (achieving up to 80% against some runner-ups).

[1]We focus on NMF as it produces top-notch state-of-the-art performance without the limitations of probabilistic approaches, such as lack of observations when applied to short texts.

2 BACKGROUND AND RELATED WORK

2.1 Data Representation

The most traditional data representation strategy for textual documents is based on simple term occurrence information, encoded by the so-called TF-IDF score (and its variants). Although this approach is, by far, the most used one (especially considering learning approaches based on vector space models), it lacks useful information such as context.

One simple strategy to overcome this is to use n-grams [6]. In the n-grams approach, a sequence of n co-occurring words (or, simply, a context window of n words) is used instead of single words. The same TF-IDF score is used, but applied to the n-grams. The use of n-grams has already shown to produce significant improvements in learning, although still limited in capturing contextual information observed in non-sequential patterns.

Recently, much has been developed in terms of data representations. Here, we pay special attention to the word embedding models, such as Word2Vec [21], GloVe [27] and FastText [22]. These models are based on co-occurrence statistics of textual datasets. They represent words as vectors so that their similarities correlate with semantic relatedness by exploiting contextual information (e.g., terms adjacent to a target one). As shown in [2], prediction models consistently outperform count models in several tasks, such as concept categorization, synonyms detection, and semantic relatedness, providing strong evidence in favor of the supposed superiority of word embedding models.

Towards the design of a richer data representation, the authors in [21] propose the so-called Word2Vec model — probably the most popular word embedding strategy so far. Similarly to GloVe, the unsupervised Word2Vec strategy aims at estimating the probability of two words occurring close to each other. This is achieved by a neural network trained with sequences of words that co-occur within a window of fixed size, in order to predict the n-th word given words $[1, ..., n-1]$ or the other way around. The output is a matrix of word vectors or context vectors. Differently from other distributional models, both Word2Vec and GloVe are prediction models, in the sense that they aim at predicting word occurrence instead of only relying on co-occurrence patterns to represent data. This usually brings up richer representations that ultimately improve the learning capabilities of downstream models. FastText [22], on the other hand, learns vectors for the sub-words (ie., character n-grams) found within each word, as well as the complete word. At each training step in FastText, the mean of the target word and subword vectors are used for training. The adjustment that is calculated from the error is then used uniformly to update each of the vectors that were combined to form the target. This adds a lot of additional computation cost to the training step. The trade-off is a set of word-vectors that contain embedded sub-word information. In [22], the authors claims that the potential benefits of FastText are: (i) it generates better word embedding for rare words; (ii) The usage of character embedding for downstream tasks have recently shown to boost the performance of those tasks compared to using word embedding like Word2Vec or GloVe. *We exploit the FastText embeddings within our CluWords*, but, as shown in our experiments, its benefits when compared to its costs are not clear, at least in the tested applications and datasets.

2.2 Latent Topic Decomposition

We now turn our attention to algorithms that aim at uncovering abstract topics from data. We start with the probabilistic models. In [5] the authors propose the so-called latent Dirichlet allocation (LDA), which generalizes how $P(w|z)$, the probability distribution over terms w considering documents belonging to the abstract topic z, is estimated. In [8], the authors proposed the Bi-term Topic Model (BTM) method to deal with the data sparsity challenge. BTM uses the concept of bi-terms generated based on co-occurrence statistics of frequent terms.

In [7], the authors deal with incoherent topics through a technique called Lifelong Topic Model (LTM): an iterative method that exploits data from several application domains that usually show some degree of information overlapping in order to produce more coherent and reliable topics. The basic assumption here is that lexical and semantic relationships are key to uncover coherent topics.

In the basic pLSA [13], the word-topic distributions (Φ) and document-topic distributions (Θ) matrices are learned by directly optimizing the log-likelihood of the training dataset $L(\Phi, \Theta)$. In the recently developed Additive Regularization of Topic Models (ARTM) [34] approach, the basic pLSA model is augmented with additive regularizers. More specifically, the Φ and Θ matrices are learned by maximizing a linear combination of $L(\Phi, \Theta)$ and r regularizers $R_i(\Phi, \Theta), \forall i = 1, \cdots, r$, with regularization coefficients τ_i as shown in Equation 1.

$$R(\Phi, \Theta) = \sum_{i=1}^{r} \tau_i R(\Phi, \Theta), \;\; L(\Phi, \Theta) + R(\Phi, \Theta) \rightarrow \max(\Phi, \Theta) \qquad (1)$$

Embedding-based Topic Model (ETM) [28] is another technique which incorporates the external word correlation knowledge into short texts to improve the coherence of topic modeling. ETM not only solves the problem of very limited word co-occurrence information by aggregating short texts into long pseudo-texts, but also utilizes a Markov Random Field regularized model that gives correlated words a better chance to be put into the same topic. *LDA, BTM , LTM, ARTM and ETM*[2] *are used as baselines here.*

The FS method [11] is a strategy used to build topics with sentiment information. It extracts words that co-occur often (a.k.a., bi-grams). Then, it infers the sentiment strength of the extracted bi-grams based on the sentiment score of the documents in which they occurred. To generate the topics, the strategy applies LDA over these sentimental bi-grams. *We use FS as one of our baselines.*

We now consider the non-probabilistic topic modeling, comprising strategies such as matrix factorization since it produces top-notch state-of-the-art performance without the limitations of probabilistic approaches, such as lack of observations when applied to short texts. In this case, a dataset with n documents and m different terms is encoded as a design matrix $A \in \mathbb{R}^{n \times m}$ and the goal is to decompose A into sub-matrices that preserve some desired property or constraint. *Our proposed framework is specifically tailored for non-probabilistic strategies.*

A well-known matrix factorization applicable to topic modeling is the Non-negative Matrix Factorization (NMF) [16]. Under this strategy, the design matrix A is decomposed into two sub-matrices $H \in \mathbb{R}^{n \times k}$ and $W \in \mathbb{R}^{k \times m}$, such that $A \approx H \times W$. In this notation, k denotes the number of latent factors (i.e., topics), H encodes the relationship between documents and topics, and W encodes the relationship between terms and topics. The restriction enforced by NMF is that all three matrices do not have any negative element. When dealing with properly represented textual data, the design matrix usually contains non-negative term scores, such as TF-IDF, with well-defined semantics (e.g., term frequency and rarity). It is natural to expect the extracted factors to be non-negative so that such semantics can be somehow preserved. *We thus consider NMF as our matrix factorization strategy of choice.* As a final note, as with the probabilistic strategies, the non-probabilistic ones can also generate incoherent topics, which is not desirable. We shall revisit this matter in next section.

Recent works have been proposed to improve the construction of topics by means of using word embedding as auxiliary information for probabilistic topic modeling. Das *et. al.* [9] propose an LDA based topic model by using multivariate Gaussian Distribution with word embedding. In [31], the authors propose the STE framework, which can learn word embedding and latent topics in a unified way. Finally, Li et al. [18] propose a model called GPU-DMM, which can promote semantically related words using the information provided by the word embedding within any topics. The GPU-DMM extends the Dirichlet Multinomial Mixture (DMM) model by incorporating the learned word relatedness from word embedding through the generalized Pólya urn (GPU) model [18] in topic inferences. *GPU-DMM is one of our baselines.*

In [32], the authors propose a semantics-assisted non-negative matrix factorization (SeaNMF) model to discover topics for the short texts. Basically, the method incorporates the word-context semantic correlations into the model. The semantic correlations between the words and their contexts are learned from the skip-gram view of the corpus, which was demonstrated to be effective for revealing word semantic relationships. *We consider SeaNMF as a baseline.*

To the best of our knowledge, there is no work that combines the information of word embedding and non-probabilistic models. The main reason for the absence of works like ours is that the introduction of the richer information provided by word embedding representation hampers the topics representation due to the lack of direct correspondence between topics and smaller semantic units (e.g., words) in these richer representations. Moreover, the instability of word embedding may yield to noisy word vectors which makes them difficult to exploit [35]. As we shall see, we propose a new topic extraction strategy to mitigate these problems.

3 PROPOSED STRATEGY

In this section, we describe our strategy to transform the traditional BOW representation of documents to include semantic information related to the terms present in the documents. The semantic context is obtained by means of a pre-trained word representation, such as Word2Vec [21], Fasttext [22]. Our approach consists in transforming each document in a new representation where original words are replaced by a cluster of words that we refer to as *CluWords*. The transformation process is composed of two phases. In the first one, we compute, for each term t of the dataset its corresponding CluWord. A CluWord for a term t is a set of terms in the vocabulary which word vectors are most similar to term t. In the second phase, we compute a modified version of the TF-IDF weighting scheme for the new features (CluWords) so that we can exploit these new

[2]The ETM strategy was trained with Glove [27], as suggested by the authors.

terms as a richer representation of documents of the collection. The first phase of our approach is described in Section 3.1, whereas the second one is presented in Section 3.2.

3.1 CluWord generation

Let $\mathcal{V}$ be the vocabulary of terms present in the set of documents $\mathcal{D}$. Also, let $\mathcal{W}$ be the set of vectors representing each term in $\mathcal{V}$ according to the pre-trained word embedding representation, for instance, Word2Vec or Fasttext. Thus, each term t in $\mathcal{V}$ has a corresponding vector in $\mathcal{W}$. Each vector $u \in \mathcal{W}$ has length l, where l is the dimensionality of the word vector space.

We define the CluWords as a matrix $C \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, where each index $C_{t,t'}$ is computed according to Eq 2.

$$C_{t,t'} = \begin{cases} \omega(t,t') & \text{if } \omega(t,t') \geq \alpha \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\omega(t,t')$ is the cosine similarity defined in Eq. 3 and α is a similarity threshold which controls the inclusion of the value of the similarity between the term t and a term t'. In this notation each CluWord is representated as a row C_t and each column t' in $\mathcal{V}$ may correspond to a component in C_t if the cosine similarity between the vectors for t and t' in the word vector space is greater than or equal to a threshold α. Otherwise, the column t' is equal to zero.

$$\omega(u,v) = \frac{\sum_i^l u_i \cdot v_i}{\sqrt{\sum_i^l u_i^2} \cdot \sqrt{\sum_i^l v_i^2}} \tag{3}$$

The CluWord C_t for term t relates t with its closest words, limiting this relationship with the cutoff value α that filters noisy words (i.e., words that do not have a significant relationship with t) from the CluWord. Since threshold α is a cosine similarity value, it is contained within the interval $[0,1]$. If $\alpha = 0$ the similarities of every term in $\mathcal{V}_T$ are included in C_t, otherwise, if $\alpha = 1$ only the similarity of t to itself (i.e. $\omega(t,t) = 1.0$) is included in C_t. Thus, the appropriate selection of a value for parameter α is an important aspect of generating good CluWords. Note that once we select an appropriate value for α, each CluWord C_t keeps the values of similarities of the terms most similar to t according to the semantics established by the word embeddings.

Table 1 presents an example of the words belonging to a CluWord whose centroid is the word "chat". The Table shows the words we consider, in a very informal analysis, syntactically, semantically or unrelated to the respective centroid.

Our intention is to use the CluWords to replace the original BOW representation of documents. It is important to note that the goal of CluWords is (mainly, but not only) to enrich the BOW representation by adding semantic information, that is, each term t will be replaced by its corresponding CluWord C_t in each document d it belongs. In order to use the CluWords representation, we need to compute the TF-IDF of the CluWords. We describe this weighting scheme in Section 3.2.

3.2 TF-IDF weights for CluWords

Basically, the conventional TF-IDF [30] is a measure of the importance of a term which evaluates two distinct aspects: (i) the relevance of the term in a specific document d (characterized by the TF component of the measure) and (ii) the importance of the term in the collection of documents to be considered (given by the IDF component). TF ($tf(t,d)$) accounts for the frequency of occurrence of term t in document d. IDF measures the importance of term t in a collection of documents. The more documents a term occurs in, the less important it is considered. Thus, the IDF of a term should be inversely related to the number of documents in which the term occurs. The TF-IDF score $tf\_idf(t,d)$ of term t in document d is defined in Eq. 4.

$$tf\_idf(t,d) = tf(t,d) \cdot log(\frac{|\mathcal{D}|}{n_t}) \tag{4}$$

where n_t is the number of documents in $\mathcal{D}$ where t occurs.

The CluWords were created based on semantic similarity of terms, so the conventional TF-IDF metric is not capable of weighting these features while taking full advantage of the information provided by them. Our motivation is to combine the two aspects of the conventional TF-IDF metric with the semantic information of a CluWord. In what follows, we propose a modified version of TF-IDF to score the CluWords to be included in extended BOW representation of document d.

The TF-IDF for CluWords is defined according to Eq. 5.

$$C_{TF-IDF} = C_{TF} \times idf(C) \tag{5}$$

First, the TF ($tf(t,d)$) can be representated as a matrix $T \in \mathbb{R}^{|\mathcal{D}|\times|\mathcal{V}|}$, where each position $T_{d,t}$ regards the frequency of a term t in document d. The TF of the CluWords can be measured as product of matrices as described in Eq. 6.

$$C_{TF} = T \times C \tag{6}$$

The value of $C_{TF_{d,t}}$ corresponds to the sum of the products of the term frequencies $T_{d,t'}$ of each term $t' \in C_{t,t'} \neq 0$ occurring in document d.

To compute the IDF of CluWord C_t we first define the vocabulary $\mathcal{V}_{d,h}$ composed by all terms in document d which have the weight (w_t) not equal to zero in CluWord C_t. This is formally defined in Eq. 7.

$$\mathcal{V}_{d,C_t} = \{t' \in d | C_{t,t'} \neq 0 \text{ in } C_t\} \tag{7}$$

Next, we compute the mean of the values of the weights in CluWord C_t of terms occurring in vocabulary $\mathcal{V}_{d,C_t}$, according to Eq. 8.

$$\mu_{C_t,d} = \frac{1}{|\mathcal{V}_{d,C_t}|} \cdot \sum_{t \in \mathcal{V}_{d,C_t}} w_t \tag{8}$$

Finally we compute IDF CluWord C_t as defined in Eq 9, where $\mathcal{D}$ is the training set.

$$idf(C_t) = log\left(\frac{|\mathcal{D}|}{\sum_{1 \leq d \leq |\mathcal{D}|} \mu_{C_t,d}}\right) \tag{9}$$

4 EXPERIMENTAL EVALUATION

4.1 Experimental Setup

4.1.1 Datasets. The primary goal of our solution is to perform effectively topic modeling so that more coherent topics can be extracted. To evaluate topic model coherence, we consider 12 real-world datasets as a reference. Two of them were created by us, collecting comments from Facebook and Uber Apps in Google Play Store. The others were obtained from previous works in the literature. For all, we performed stopword removal (using the standard SMART list) and removed words such as adverbs, using the VADER lexicon dictionary [14], as the vast majority of the important words for identifying topics are nouns and verbs. These procedures improved both, the efficiency and effectiveness of all analyzed strategies. Table 2 provides a brief summary of the reference datasets,

Table 1: Example of the words belonging to a CluWord with "chat" as centroid.

Centroid: chat	
Semantically similar words	audio, communicate, communication, contact, conversation, conversations, discuss, email, emails, forum, hear, interact, interaction, listen, listening, mail, message, messages, messaging, news, phone, post, reply, socialize, socializing, speak, talk, talking, voice, meeting, networking, room, service
Syntactically similar words	chat, chats, chatted, chatting
Unrelated/Undefined words	access, avatar, buddies, buddy, download, dude, evening, evenings, exchange, gallery, game, gaming, girl, girlfriend, guys, homework, interface, mate, mates, pal, server, sip, sit, smilies, strangers, stuff, telephone, thoughts, twitter, video, wander, wanna, web

reporting the number of features (words), documents, the mean number of words per document (density) and the corresponding references.

Table 2: Dataset characteristics

Dataset	#Feat	#Doc	Density
Angrybirds [11]	1,903	1,428	7.135
Dropbox [11]	2,430	1,909	9.501
Evernote [11]	6,307	8,273	11.002
InfoVis-Vast [3]	6,104	909	86.215
Pinterest [11]	2,174	3,168	4.478
TripAdvisor [11]	3,152	2,816	8.532
Tweets [20]	8,029	12,030	4.450
WhatsApp [11]	1,777	2,956	3.103
20NewsGroup [4]	29,842	15,411	76.408
ACM [33]	16,811	22,384	30.428
Uber	5,517	11,541	7.868
Facebook	5,168	12,297	6.427

4.1.2 Evaluation, Algorithms and Procedures. We compare the topic modeling strategies using representative topic quality metrics in the literature [24, 25]. In general, there are three class of topic quality metrics based on three criteria: (a) coherence, (b) mutual information and (c) semantic representation. In this paper, we focus on (a) and (b) since they are the most used metric in the literature [25, 32]. We also consider three topic lengths (5, 10 and 20 words) under each metric in our evaluation—different lengths may bring different challenges.

Regarding the metrics, *coherence* captures easiness of interpretation by co-occurrence. Words that co-occur frequently in similar contexts in a corpus are easier to correlate since they usually define a more well-defined "concept" or "topic". For coherence, we employ an improved version of regular coherence [25], called TFIDF-Coherence, defined as

$$c_{tf-idf}(t, W_t) = \sum_{w_1, w_2 \in W_t} log \frac{\sum_{d: w_1, w_2 \in d} tf - idf(w_1, d) tf - idf(w_2, d)}{\sum_{d: w_1 \in d} tf - idf(w_1, d)} \quad (10)$$

where the *tf-idf* metric is computed with augmented frequency as

$$tf - idf(w, d) = \left(\frac{1}{2} + \frac{f(w, d)}{max_{w' \in d} f(w', d)}\right) log \frac{|D|}{|\{d \in D : w \in d\}|} \quad (11)$$

and $f(w, d)$ is the number of occurrences of a term w in document d. This skews the metric towards topics with high *tf-idf* scores since the numerator of the coherence fraction has quadratic dependence on the *tf-idf* scores and the denominator only linear.

Another class of topic quality metrics is based on the notion of *pairwise point-wise mutual information (PMI)* between the top words in a topic. It captures how much one "gains" in information given the occurrence of the other word, taking dependencies between words into consideration. Following a recent work [24], we compute a *normalized version of PMI* (NPMI), in which, for a given ordered set of top words $W_t = (w_1, ..., w_N)$ in a topic, NPMI is computed as:

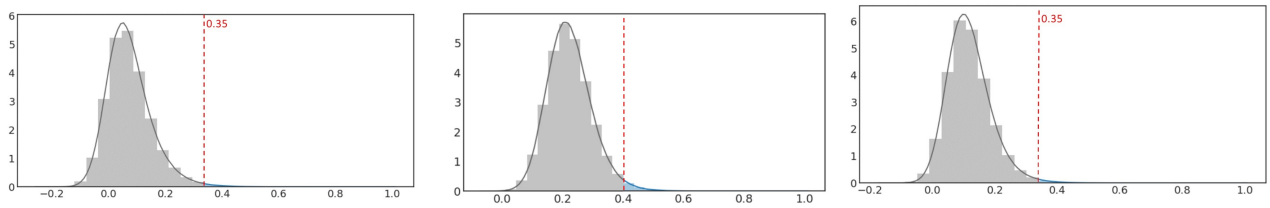
$$NPMI_t = \sum_{i<j} \frac{log \frac{p(w_i, w_j)}{p(w_i)p(w_j)}}{-log\ p(w_i, w_j)} \quad (12)$$

Next, we compare our proposed data representation described in Section 3, as well the best configuration with eight topic model strategies recently proposed, marked in bold in Section 2. In our experiments, we adopt the Non-negative Matrix Factorization (NMF) [16] to evaluate the CluWords, since it is the main non-probabilistic matrix factorization. We discovered 25 topics for all datasets except 20News, ACM and Tweets, where 20, 11 and 6 topics were discovered for these datasets, respectively. The number of topics for the app datasets was defined based on the choice of topics made in [11]. For the 20News, ACM and Tweets datasets, we chose the number of topics equals to the real number of classes. We assess the statistical significance of our results by means of a paired t-test with 95% confidence and Holm-Bonferroni correction to account for multiple tests. In the next section, we present the results of experiments conducted to evaluate the effectiveness of the CluWords using three different pre-trained word embedding spaces, considering NPMI scores. Next, we compare the best word embeddings instantiation with the baseline strategies, regarding the scores of Coherence and NPMI.

4.2 Experimental Results

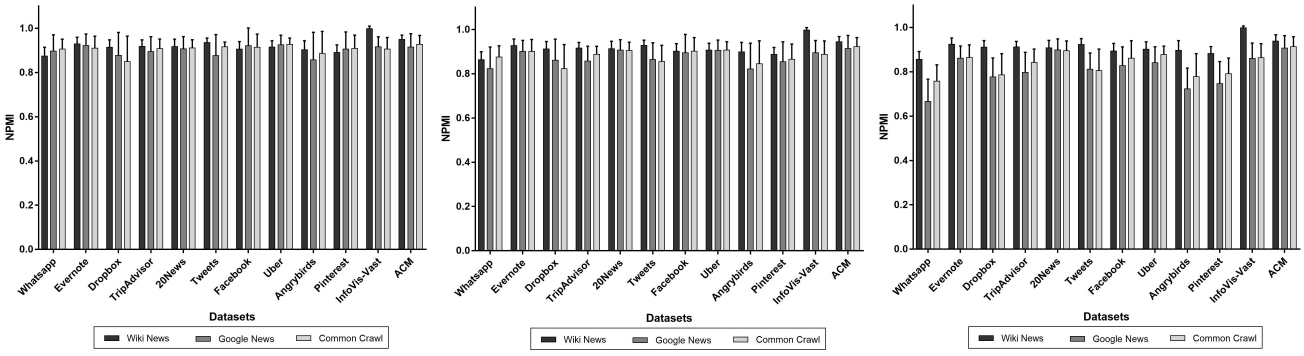
4.2.1 Choosing the best word embedding space. In this section we compare the proposed CluWords with three pre-trained word embeddings spaces: (i) Word2Vec trained with GoogleNews [21]; (ii) FastText trained with WikiNews [22] and (iii) Fasttext trained on Common Crawl [22]. Initially, to build the proposed data representation, as described in section 3, we need to select a cosine similarity threshold α. The idea is to select a threshold α that is restrictive, capable of filtering pairs of noisy words. For this, we need to find the distribution of similarities between the word pairs of the word embedding to infer a similarity threshold. The Figure 1 shows the distribution of similarities of each pre-trained word embedding. We can observe that the distribution of similarities of the three-word vector spaces is quite similar and that the FastText WikiNews

[3] https://www.cc.gatech.edu/gvu/ii/jigsaw/datafiles.html
[4] http://qwone.com/~jason/20Newsgroups/



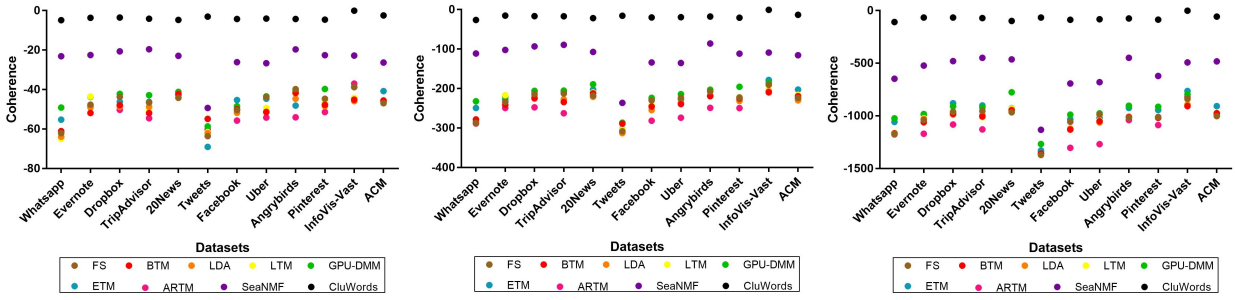
(a) W2V GoogleNews (b) FastText WikiNews (c) FastText Common Crawl

Figure 1: Cosine similarity histogram.



(a) 5 Words (b) 10 Words (c) 20 Words

Figure 2: Evaluation of CluWords exploring different Word Embeddings, in terms of NPMI score.



(a) 5 Words (b) 10 Words (c) 20 Words

Figure 3: Comparing the results achieved by each strategy considering top 5, 10 and 20 words for TFIDF-Coherence.

presents a slightly greater deviation than the other word embeddings. Thus, for our experiments, we chose a threshold α capable of selecting only 2% of the most similar word pairs. We select a high threshold α just to avoid an unexpected pair of terms since the space of the word vectors are not evenly dispersed, according to to [23]. Thus, the threshold selected for FastText WikiNews is $\alpha \geq 0.40$, while for W2V GoogleNews and FastText Common Crawl a threshold of $\alpha \geq 0.35$ has been selected.

Figure 2 contrasts the results of the proposed CluWords on the three evaluated word embedding spaces. The FastText WikiNews always achieves superior results considering all datasets and topic lengths (5, 10 and 20 words). In fact, most of the results (32 out of 36 results) are statistic ties, which suggests that the proposed data representation is capable of performing with the same quality in the three distinct word vector spaces. The CluWords results presented in the next Section were generated using the word embeddings from FastText WikiNews.

We performed an additional quantitative experiment using the Fasttext WikiNews space to reinforce the evidence that CluWords can also capture syntactic information. In the experiment, our goal is to show that in the process of selecting the neighborhood of a CluWord C_t (Section 3.1), a part of the terms closest to term t are variations of the same word (e.g. the word *chats* is a variation of the word *chat*). Thus, given a CluWord C_t, we select each term $t'|C_{t,t'} \neq 0$ and derive t' to its stem form. We measured the proportion of terms affected by the stemming process. Table 4 illustrates

the average affected terms in the CluWords, for the 12 datasets. We can observe that approximately 11% of the terms belonging to a CluWord are the variation of the same word.

4.2.2 Effectiveness Results Against the Baselines. We compare our proposed solution against eight state-of-the-art topic modeling strategies considering the twelve reference datasets. In Figure 3, our strategy achieves statistically significant gains in terms of the quality of the discovered topics in all 12 datasets, considering the Coherence score. Most baselines cannot get even close, with the best baseline (SeaNMF) being worse than Cluwords by more than 33% when it obtains its best performance (in TripAdvisor), considering the three evaluated topic lengths. These are very strong results as SeaNMF is considered **the state-of-the-art** in Topic Modeling (besides being a very recent proposal [32]).

In Table 3, we contrast the results of CluWords and the reference strategies, considering the NPMI metric. The best results, marked with ▲, are statistically superior to others. Statistical ties are represented with ●. As we can see, our strategy achieves the single best results in 7 out of 36 results, tying with SeaNMF in the other 29 as the **best** method in terms of the quality of the discovered topics, considering the NPMI score. Again, the other baselines' results are far below, reinforcing that SeaNMF is the baseline to be beaten.

Table 4: Syntactic information in the CluWords.

Datasets	Syntatic information
Whatsapp	10.37 ± 6.21
Angrybirds	10.76 ± 6.11
20News	14.47 ± 6.54
Dropbox	12.83 ± 6.53
InfoVis-Vast	17.57 ± 7.66
Tweets	13.34 ± 6.37
ACM	15.58 ± 7.54
Evernote	14.02 ± 7.05
Pinterest	12.12 ± 5.99
Uber	12.99 ± 7.05
Facebook	12.62 ± 6.78
Tripadvisor	12.92 ± 6.90

Another perspective of the results can be taken when analyzing the standard deviations of the results. The ones obtained by CluWords are considerably smaller than those of SeaNMF. Figure 4 allows us to more clearly observe the differences between the NPMI deviations, considering topics with 10 words. To better quantify this, we performed two variability tests. Equal variances across samples are also called *homogeneity of variance*. Some statistical tests, for instance, the analysis of variance, assume that variances are equal across groups or samples. Levene's test [17] and Bartlett's test [3] can be used to verify that assumption. The Levene's test is less sensitive than the Bartlett's test to departures from normality. On the other hand, if the data come indeed from a normal, or nearly normal, distribution, then Bartlett's test should perform better. As we cannot assume any of the options, we applied both tests. In these tests, if the resulting p-value is less than some significance level (e.g., p-value < 0.05), the obtained differences in sample variances are unlikely to have occurred based on random sampling from a population with equal variances (i.e. different variances).

Table 5 presents the test for equality of variances with respect to the NPMI scores of both, CluWords and SeaNMF. We marked in ▲ the p-values that present statistically significant differences between the variances and use ● when both strategies have the same variance. The Table 5 shows that in 21 out of 24 tests the CluWords and SeaNMF have indeed different variances. Tweets is the only dataset in which the test showed equivalent variances between the strategies. However, in this dataset, CluWords outperforms SeaNMF considering all three topic lengths Table 3). Thus, we can conclude that our CluWords strategy is capable of generating the best semantically cohesive topics, in terms of TFIDF coherence and NPMI, with less variability in NPMI according to Levene's and Bartlett's tests.

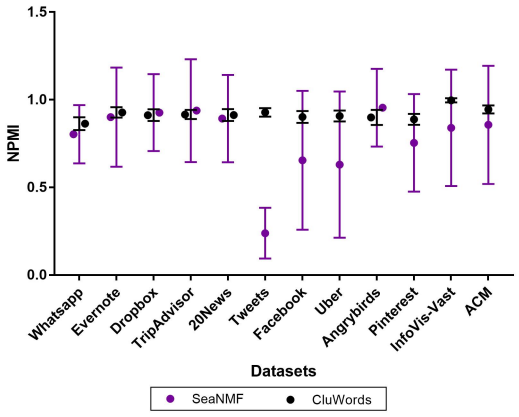


Figure 4: Comparison of NPMI scores for Cluwords and SeaNMF strategies considering 10 words.

4.3 Application: Document Classification

As we have seen, our proposed method is capable of generating more cohesive topics and potentially better document representations, which can potentially help in tasks such as automatic classification and clustering. Due to lack of space, we analyze the suitability of the information our model in the classification task, leaving the analysis of other applications for future work. We consider the ACM and 20News datasets which have a ground truth for topics to evaluate the impact of the use of information exploited by topic modeling strategies in document classification. We compare three kinds of information extracted from topic model: (1) CluWord topics, which are the latent vector information extracted from the topic modeling, (2) the SeaNMF latent topics and (3) the CluWord document representation, described in Section 3. Each kind of information is combined with the original BOW representation, which is also a baseline.

All experiments were executed using a 5-fold cross-validation and the SVM, which is a top-noch method for text classification [10]. The regularization parameter was chosen among eleven values from 2^{-5} to 2^{15} by using 5-fold nested cross-validation within the training set. We assess the statistical significance of our results by means of a paired t-test with 95% confidence and Holm correction to account for multiple tests. This test assures that the best results, marked with ▲, are statistically superior to others.

Table 3: Comparing the results achieved by each strategy considering top 5, 10 and 20 words for NPMI.

Strategies	Whatsapp			Evernote			Dropbox			TripAdvisor		
	5 words	10 words	20 words	5 words	10 words	20 words	5 words	10 words	20 words	5 words	10 words	20 words
FS	0.171 ± 0.051	0.201 ± 0.048	0.230 ± 0.043	0.102 ± 0.052	0.090 ± 0.020	0.100 ± 0.018	0.109 ± 0.042	0.097 ± 0.027	0.107 ± 0.018	0.094 ± 0.037	0.092 ± 0.028	0.104 ± 0.021
BTM	0.201 ± 0.057	0.236 ± 0.038	0.284 ± 0.038	0.118 ± 0.057	0.109 ± 0.029	0.120 ± 0.024	0.155 ± 0.050	0.161 ± 0.043	0.166 ± 0.040	0.130 ± 0.052	0.144 ± 0.044	0.158 ± 0.039
LDA	0.172 ± 0.050	0.230 ± 0.030	0.284 ± 0.042	0.114 ± 0.067	0.114 ± 0.036	0.114 ± 0.019	0.165 ± 0.110	0.149 ± 0.056	0.150 ± 0.037	0.114 ± 0.057	0.122 ± 0.029	0.137 ± 0.028
LTM	0.178 ± 0.052	0.225 ± 0.041	0.269 ± 0.040	0.193 ± 0.051	0.168 ± 0.044	0.158 ± 0.033	0.167 ± 0.072	0.160 ± 0.040	0.175 ± 0.046	0.149 ± 0.059	0.144 ± 0.035	0.161 ± 0.037
GPU-DMM	0.312 ± 0.165	0.327 ± 0.141	0.330 ± 0.131	0.258 ± 0.165	0.270 ± 0.149	0.229 ± 0.076	0.284 ± 0.147	0.267 ± 0.125	0.284 ± 0.129	0.286 ± 0.209	0.253 ± 0.144	0.244 ± 0.122
ETM	0.365 ± 0.171	0.378 ± 0.163	0.399 ± 0.154	0.319 ± 0.138	0.320 ± 0.133	0.331 ± 0.131	0.403 ± 0.094	0.399 ± 0.109	0.398 ± 0.119	0.347 ± 0.151	0.349 ± 0.154	0.355 ± 0.163
ARTM	0.174 ± 0.046	0.248 ± 0.036	0.339 ± 0.042	0.125 ± 0.050	0.118 ± 0.019	0.139 ± 0.013	0.158 ± 0.041	0.183 ± 0.036	0.239 ± 0.030	0.128 ± 0.042	0.168 ± 0.030	0.226 ± 0.030
SeaNMF	0.884 ± 0.256 ●	0.803 ± 0.166 ●	0.576 ± 0.112	0.932 ± 0.293 ●	0.901 ± 0.283 ●	0.780 ± 0.241 ●	0.968 ± 0.222 ●	0.927 ± 0.219 ●	0.784 ± 0.185 ●	0.951 ± 0.292 ●	0.938 ± 0.293 ●	0.816 ± 0.262 ●
CluWords	0.875 ± 0.039 ●	0.864 ± 0.036 ●	0.856 ± 0.036 ▲	0.929 ± 0.031 ●	0.928 ± 0.029 ●	0.924 ± 0.029 ●	0.914 ± 0.034 ●	0.912 ± 0.033 ●	0.912 ± 0.029 ●	0.918 ± 0.030 ●	0.916 ± 0.026 ●	0.912 ± 0.025 ●
Strategies	**20News**			**Tweets**			**Facebook**			**Uber**		
	5 words	10 words	20 words	5 words	10 words	20 words	5 words	10 words	20 words	5 words	10 words	20 words
FS	0.119 ± 0.056	0.110 ± 0.026	0.110 ± 0.022	0.071 ± 0.054	0.066 ± 0.033	0.078 ± 0.005	0.061 ± 0.065	0.054 ± 0.033	0.050 ± 0.014	0.056 ± 0.043	0.045 ± 0.023	0.048 ± 0.016
BTM	0.244 ± 0.117	0.217 ± 0.089	0.192 ± 0.059	0.142 ± 0.061	0.100 ± 0.026	0.095 ± 0.019	0.137 ± 0.063	0.110 ± 0.036	0.118 ± 0.029	0.093 ± 0.044	0.094 ± 0.036	0.094 ± 0.026
LDA	0.218 ± 0.121	0.196 ± 0.084	0.174 ± 0.063	0.083 ± 0.055	0.060 ± 0.028	0.079 ± 0.020	0.115 ± 0.067	0.085 ± 0.028	0.095 ± 0.023	0.094 ± 0.053	0.083 ± 0.030	0.089 ± 0.012
LTM	0.224 ± 0.134	0.196 ± 0.074	0.179 ± 0.049	0.109 ± 0.060	0.084 ± 0.022	0.093 ± 0.017	0.146 ± 0.079	0.113 ± 0.048	0.119 ± 0.027	0.097 ± 0.065	0.088 ± 0.032	0.091 ± 0.022
GPU-DMM	0.421 ± 0.044	0.477 ± 0.044	0.471 ± 0.031	0.090 ± 0.062	0.081 ± 0.051	0.092 ± 0.046	0.326 ± 0.170	0.313 ± 0.164	0.282 ± 0.162	0.322 ± 0.241	0.275 ± 0.199	0.240 ± 0.142
ETM	0.249 ± 0.109	0.262 ± 0.092	0.243 ± 0.066	0.057 ± 0.044	0.071 ± 0.038	0.092 ± 0.041	0.198 ± 0.090	0.186 ± 0.087	0.171 ± 0.095	0.180 ± 0.077	0.173 ± 0.074	0.165 ± 0.096
ARTM	0.281 ± 0.105	0.235 ± 0.076	0.216 ± 0.062	0.091 ± 0.055	0.068 ± 0.031	0.080 ± 0.025	0.079 ± 0.044	0.091 ± 0.023	0.136 ± 0.021	0.075 ± 0.043	0.091 ± 0.020	0.135 ± 0.018
SeaNMF	0.897 ± 0.247 ●	0.893 ± 0.249 ●	0.891 ± 0.254 ●	0.237 ± 0.183	0.239 ± 0.145	0.195 ± 0.056	0.718 ± 0.410 ●	0.655 ± 0.396 ●	0.546 ± 0.312	0.684 ± 0.434 ●	0.630 ± 0.417 ●	0.522 ± 0.343
CluWords	0.917 ± 0.034 ●	0.913 ± 0.034 ●	0.908 ± 0.034 ●	0.935 ± 0.021 ▲	0.928 ± 0.024 ▲	0.923 ± 0.027 ▲	0.906 ± 0.034 ●	0.902 ± 0.034 ●	0.894 ± 0.034 ▲	0.915 ± 0.029 ●	0.907 ± 0.031 ●	0.902 ± 0.033 ▲
Strategies	**Angrybirds**			**Pinterest**			**InfoVis-Vast**			**ACM**		
	5 words	10 words	20 words	5 words	10 words	20 words	5 words	10 words	20 words	5 words	10 words	20 words
FS	0.053 ± 0.036	0.077 ± 0.033	0.124 ± 0.028	0.102 ± 0.077	0.096 ± 0.050	0.112 ± 0.031	0.049 ± 0.039	0.057 ± 0.026	0.056 ± 0.019	0.148 ± 0.107	0.136 ± 0.050	0.128 ± 0.044
BTM	0.132 ± 0.075	0.154 ± 0.034	0.193 ± 0.040	0.148 ± 0.074	0.144 ± 0.043	0.147 ± 0.032	0.193 ± 0.079	0.170 ± 0.071	0.149 ± 0.051	0.176 ± 0.084	0.146 ± 0.055	0.136 ± 0.051
LDA	0.137 ± 0.065	0.154 ± 0.038	0.190 ± 0.044	0.144 ± 0.062	0.135 ± 0.043	0.147 ± 0.039	0.154 ± 0.075	0.153 ± 0.064	0.139 ± 0.051	0.138 ± 0.062	0.122 ± 0.051	0.117 ± 0.046
LTM	0.117 ± 0.061	0.154 ± 0.041	0.189 ± 0.041	0.145 ± 0.061	0.137 ± 0.051	0.143 ± 0.035	0.182 ± 0.092	0.158 ± 0.058	0.131 ± 0.042	0.173 ± 0.095	0.163 ± 0.074	0.143 ± 0.054
GPU-DMM	0.260 ± 0.173	0.286 ± 0.141	0.301 ± 0.142	0.330 ± 0.192	0.322 ± 0.194	0.278 ± 0.146	0.264 ± 0.155	0.259 ± 0.104	0.207 ± 0.103	0.233 ± 0.101	0.208 ± 0.113	0.189 ± 0.095
ETM	0.366 ± 0.089	0.373 ± 0.079	0.385 ± 0.085	0.358 ± 0.114	0.355 ± 0.109	0.370 ± 0.115	0.304 ± 0.163	0.304 ± 0.157	0.313 ± 0.158	0.266 ± 0.086	0.230 ± 0.052	0.201 ± 0.035
ARTM	0.209 ± 0.066	0.262 ± 0.054	0.337 ± 0.051	0.167 ± 0.060	0.198 ± 0.030	0.264 ± 0.027	0.102 ± 0.095	0.084 ± 0.077	0.076 ± 0.045	0.178 ± 0.088	0.147 ± 0.058	0.149 ± 0.047
SeaNMF	0.964 ± 0.238 ●	0.955 ± 0.222 ●	0.808 ± 0.194 ●	0.836 ± 0.311 ●	0.754 ± 0.278 ●	0.552 ± 0.167	0.861 ± 0.321 ●	0.840 ± 0.332 ●	0.768 ± 0.288	0.843 ± 0.336 ●	0.857 ± 0.337 ●	0.860 ± 0.345 ●
CluWords	0.903 ± 0.041 ●	0.899 ± 0.043 ●	0.897 ± 0.044 ●	0.891 ± 0.034 ●	0.888 ± 0.031 ●	0.883 ± 0.031 ▲	0.998 ± 0.012 ●	0.997 ± 0.012 ●	0.998 ± 0.009 ▲	0.950 ± 0.019 ●	0.945 ± 0.023 ●	0.939 ± 0.028 ●

Table 5: Test for Equality of Variances considering 20 Words.

Datasets	Variance		p-value	
	CluWords	SeaNMF	Levene's test	Bartlett's test
20News	0.0013	0.0644	0.004▲	0.0 ▲
ACM	0.0008	0.0741	0.169●	0.018▲
Angrybirds	0.0020	0.0375	0.002▲	0.014▲
Dropbox	0.0009	0.0343	0.006▲	0.006▲
Evernote	0.0009	0.0582	0.015▲	0.000▲
Facebook	0.0012	0.0971	0.000▲	0.000▲
Infovisvast	0.0001	0.0831	0.000▲	0.000▲
Pinterest	0.0010	0.0278	0.002▲	0.001▲
TripAdvisor	0.0007	0.0687	0.004▲	0.000▲
Tweets	0.0009	0.0032	0.112●	0.138●
Uber	0.0011	0.1179	0.000▲	0.000▲
WhatsApp	0.0013	0.0125	0.001▲	0.000▲

Table 6: Average Macro-F1 and Micro-F1 for the classification task using different document representations.

Representation	ACM		20NG	
	MicF1	MacF1	MicF1	MacF1
BOW	69.1(0.4)	57.3(1.64)	89.6(0.5)	89.5(0.5)
CluWords	74.0(0.8)	61.9(1.8)	91.1(0.8)	91.0(0.9)
CluWords Topics	76.0(0.5) ▲	62.8(1.5) ▲	92.4(0.2) ▲	92.2(0.3) ▲
SeaNMF Topics	71.2(0.8)	61.3(1.4)	87.0(0.3)	87.0(0.2)

Table 6 presents the classification effectiveness on MicroF_1 and MacroF_1 measures [37]. In all situations, the use the CluWords latent topics obtained the best classification results, with statistical significance on both evaluated datasets. This indicates that the discriminative information provided by the latent topics can improve the classification results. Moreover, the compact representation of the latent topics – which adds at most k dimensions to the problem, k being the number of chosen topics – can avoid the cost of including other highly-dimensional features, such as the individual CluWords – which can add up to hundrends or thousands of new dimensions. The savings on terms of computational cost are obvious. In fact, when compared to the original Bow, gains of up to 10% in both Micro and MacroF1 were observed (in ACM).

On the other hand, the SeaNMF topics were not able to provide as much relevant discriminative information as the CluWord topics. In fact, there is evidence that SeaNMF included noisy features for classification, since the results on 20News are significantly worse than the BOW representation.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel document representation for Topic Modeling - Cluwords. Our solution can be thought as the "best of all worlds": (i) as manually-built semantic dictionaries, it exploits explicit semantic relationships between words, but without their limitations (scalability, adaptability); (ii) it exploits large word embedding spaces to automatize the process of computing such relationships; (iii) it conjugates into a single representation syntactic and semantic information; and (iv) as the widely used TF-IDF scheme, it proposes a way to measure (i.e., weight) the importance of a given CluWord to express the topics of a document.

Our thorough experimental evaluation (12 datasets, 8 baselines, 2 evaluation metrics, 3 topic lengths) showed that sometimes by large margins, we outperform the best (state-of-the-art) methods for Topic Modeling known in the literature, with a much smaller variability in terms of the quality of the produced topics. In other words, our CluWords results are currently the ones to be beaten, setting a new high standard for the Topic Modeling research area. We also demonstrated that the generated topics have the potential to improve other applications such as automatic text classification.

We envision plenty of future work ahead of us. A lot still needs to be understood in terms of the theoretical properties of the Cluword clusters. For instance, can we exploit other notions of similarity

that do not rely solely on the generated embedding spaces or use other types of similarities more suited to these spaces, avoiding unexpected results[23]? As we have seen, there is also space for filtering out irrelevant topics or strip unrelated words out of the CluWords. We also need to evaluate some type of "recall" measure of the CluWords, for instance: do they contain all the syntactic variations it should for a given centroid? And in terms of semantics, when is it worth or should we "merge" different CluWords with the same or very close centroids? We have seen that in practice, Cluwords with the same centroid do occur. This "merging" aspect is a variant of the classical k-means strategy, but CluWords bring some particular and specific issues. For example, in K-means, no cluster shares the same centroid when the process stops. But it is not clear whether this property should hold for CluWords as merging clusters with the same or close centroids can just bring more noise to the process. And, since we are talking about clusters of words, we should test which strategies, beyond classical k-means, are better suited for them, for example, hierarchical or density-based clustering. Finally, another venue we intend to exploit it to learn (from the data) not only new weighting schemes for the CluWords but also new similarity measures adapted for the particularities of a dataset (density, number of features, etc).

6 ACKNOWLEDGMENTS

This work is partially supported by CAPES, CNPq, Finep, Fapemig, Mundiale, Astrein, projects InWeb and MASWeb.

REFERENCES

[1] Klemens Boehm Abel Elekes, Martin Schaeler. 2017. On the Various Semantics of Similarity in Word Embedding Models. *Digital Libraries (JCDL) 2017 ACM/IEEE Joint Conference* (2017).
[2] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *ACL'14*.
[3] M. S. Bartlett. 1937. Properties of Sufficiency and Statistical Tests. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 160 (1937).
[4] David M. Blei. 2012. Probabilistic Topic Models. *Communications of The ACM* 55, 4 (April 2012), 77–84. https://doi.org/10.1145/2133806.2133826
[5] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* (2003).
[6] William B. Cavnar and John M. Trenkle. 1994. N-Gram-Based Text Categorization. In *SDAIR'94*.
[7] Zhiyuan Chen and Bing Liu. 2014. Topic Modeling Using Topics from Many Domains, Lifelong Learning and Big Data. In *ICML'14*.
[8] X. Cheng, X. Yan, Y. Lan, and J. Guo. 2014. BTM: Topic Modeling over Short Texts. *IEEE TKDE '14* (2014).
[9] Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for Topic Models with Word Embeddings.. In *ACL '15*.
[10] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* (2008).
[11] Emitza Guzman and Walid Maalej. 2014. How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews.. In *Requirements Engineering*.
[12] William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing Domain-Specific Sentiment Lexicons from Unlabeled Corpora. *CoRR* (2016).
[13] Thomas Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *SIGIR '99*.
[14] Clayton J. Hutto and Eric Gilbert. 2014. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *ICWSM'14*.
[15] Ou Jin, Nathan N. Liu, Kai Zhao, Yong Yu, and Qiang Yang. 2011. Transferring Topical Knowledge from Auxiliary Long Texts for Short Text Clustering. In *CIKM*.
[16] Daniel D. Lee and H. Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* (1999).
[17] Howard Levene. 1960. Robust tests for equality of variances. (1960).
[18] Chenliang Li, Yu Duan, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2017. Enhancing Topic Modeling for Short Texts with Auxiliary Word Embeddings. *ACM TOIS* (2017).
[19] Chenliang Li, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2016. Topic Modeling for Short Texts with Auxiliary Word Embeddings. In *SIGIR'16*.
[20] Quanzhi Li, Sameena Shah, Xiaomo Liu, Armineh Nourbakhsh, and Rui Fang. [n. d.]. TweetSift: Tweet Topic Classification Based on Entity Knowledge Base and Topic Enhanced Word Embedding. In *CIKM'16*.
[21] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. (2013).
[22] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *LREC'18*.
[23] David M. Mimno and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *EMNLP*.
[24] Sergey I Nikolenko. 2016. Topic Quality Metrics Based on Distributed Word Representations. In *SIGIR'16*.
[25] Sergey I Nikolenko, Sergei Koltcov, and Olessia Koltsova. 2017. Topic modelling for qualitative studies. *Journal of Information Science* (2017).
[26] G. Pedrosa, M. Pita, P. Bicalho, A. Lacerda, and G. L. Pappa. 2016. Topic Modeling for Short Texts with Co-occurrence Frequency-Based Expansion. In *BRACIS*.
[27] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global Vectors for Word Representation.. In *EMNLP*.
[28] Jipeng Qiang, Ping Chen, Tong Wang, and Xindong Wu. 2017. Topic Modeling over Short Texts by Incorporating Word Embeddings. In *PAKDD*. Springer.
[29] Timothy N. Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. 2012. Statistical Topic Models for Multi-label Document Classification. *Mach. Learn.* 88, 1-2 (July 2012), 157–208. https://doi.org/10.1007/s10994-011-5272-5
[30] Gerard Salton and Christopher Buckley. 1988. Term-weighting Approaches in Automatic Text Retrieval. *Inf. Process. Manage.* 24, 5 (1988), 513–523.
[31] Bei Shi, Wai Lam, Shoaib Jameel, Steven Schockaert, and Kwun Ping Lai. 2017. Jointly Learning Word Embeddings and Latent Topics. In *SIGIR'17*.
[32] Tian Shi, Kyeongpil Kang, Jaegul Choo, and Chandan K. Reddy. 2018. Short-Text Topic Modeling via Non-negative Matrix Factorization Enriched with Local Word-Context Correlations. In *WWW '18*. 1105–1114.
[33] Felipe Viegas, Marcos Gonçalves, Wellington Martins, and Leonardo Rocha. 2015. Parallel Lazy Semi-Naive Bayes Strategies for Effective and Efficient Document Classification. In *CIKM*.
[34] Konstantin Vorontsov and Anna Potapenko. 2015. Additive regularization of topic models. *Mach. Learn.* 101, 1-3 (2015), 303–323.
[35] Laura Wendlandt, Jonathan K. Kummerfeld, and Rada Mihalcea. 2018. Factors Influencing the Surprising Instability of Word Embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*. Association for Computational Linguistics.
[36] Pengtao Xie and Eric P. Xing. 2013. Integrating Document Clustering and Topic Modeling. *CoRR* abs/1309.6874 (2013).
[37] Yiming Yang. 1999. An Evaluation of Statistical Approaches to Text Categorization. *Inf. Ret.* (1999).
[38] Hamed Zamani and W. Bruce Croft. 2017. Relevance-based Word Embedding. In *SIGIR'17*.
[39] Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing Twitter and Traditional Media Using Topic Models. In *ECIR'11*.