Word Embedding-based Topic Similarity Measures

Silvia Terragni[0000−0002−0358−1854], Elisabetta Fersini*[0000−0002−8987−100X], and Enza Messina[0000−0002−4062−0824]

University of Milano-Bicocca, Milan, Italy
s.terragni4@campus.unimib.it,
{elisabetta.fersini, enza.messina}@unimib.it

Abstract. Topic models aim at discovering a set of hidden themes in a text corpus. A user might be interested in identifying the most similar topics of a given theme of interest. To accomplish this task, several similarity and distance metrics can be adopted. In this paper, we provide a comparison of the state-of-the-art topic similarity measures and propose novel metrics based on word embeddings. The proposed measures can overcome some limitations of the existing approaches, highlighting good capabilities in terms of several topic performance measures on benchmark datasets.

Keywords: Topic Modeling · Topic Similarity · Word Embeddings

1 Introduction

Topic models [7,10,24] are a suite of probabilistic models that aim at extracting the main themes (or "topics") from a collection of documents. When a topic model automatically generates a set of topics underlying a given corpus, few of them could be similar while others could be different. For instance, a topic about technology, characterized by the words "card video monitor cable vga", is more similar to the topic "gif image format jpeg color" than one about animals ("cat animal dog cats tiger"). Methods for automatically determining the similarity between topics have several potential applications, such as the validation of the quality of the topic modeling output for determining potential overlaps between pairs of topics [2] and document retrieval based on topic proximity [10].

To estimate the similarity between topics, several metrics have been introduced in the state of the art. Most of them are based on word tokens and usually adopt a list of top-t terms to estimate if two topics are related. On the other hand, few approaches exploit the probability distribution of the words denoting the topics to compute the similarity between themes. These distribution-based measures suffer from the high dimensionality of the vocabulary, generating solutions that do not strongly correlate with human judgment [1]. On the contrary, approaches that focus only on the word tokens of a topic [26,5] ignore that two

* Corresponding author.

words could be lexicographically different but denoting a similar meaning. For instance, the words *cat* and *kitten* should not be considered totally dissimilar. A preliminary investigation that partially addressed the above problems has been introduced in [1]. They represent the words of a topic as vectors in a semantic space constructed from an external source or from the corpus using Pointwise Mutual Information (PMI). However, this approach is computationally expensive, requiring to compute the probability of the co-occurrence for each pair of words in the corpus, and does not take into account the more recent advances in Word Embeddings [18,21,9], that have already proved their benefits in several NLP applications and topic modeling [20,3]. Moreover, this approach does not take into account that the topics extracted are actually ranked lists of words, where the rank provides useful insight. In particular, if two topics contain the same words but at different ranking positions, this aspect should be considered when evaluating the similarity of the generated solution.

We therefore propose new topic similarity metrics that exploit the nature of word embeddings and take into consideration topics as ranked lists of words. We demonstrate in the experimental evaluation that these metrics can discover semantically similar topics, also outperforming the state-of-the-art topic similarity metrics.

The paper is organized as follows. In section 2, the main state-of-the-art topic similarity measures are described. In section 3, we present the proposed metrics, which are based on Word Embeddings. In section 4, the experimental investigation is detailed. In section 5, we outline the conclusions and future work.

2 Topic similarity/distance measures: state of the art

The goal of topic modeling is to extract K topics from a document corpus, where each topic is represented as a multinomial distribution over the vocabulary, usually referred to as *word-topic distribution*. Researchers usually consider the top-t most probable words (from the word-topic distribution) to represent a topic. This top-t ranked list of words is usually called *topic descriptor* [4]. The word-topic distribution and topic descriptors are the two key elements that can be exploited to estimate the similarity between two themes. In what follows, we will review the most relevant topic similarity measures that have been proposed in the literature.The topic descriptor of a topic i will be referred to as t_i, represented by its top-t most likely words, i.e. $t_i = \{v_0, v_1, \ldots, v_{t-1}\}$, where v_k is a word of the vocabulary V. We will refer to the word distribution of a topic i as β_i, which is a multinomial distribution over the vocabulary V. In particular, $\beta_i(v)$ represents the probability of the word v in the topic i.

We will introduce in the following subsections the metrics already available in the state of the art, by roughly dividing them into metrics that are based on the counts of the shared word tokens and metrics that are based on the probability distributions.

2.1 Measures based on Shared Word Tokens

A simple way to compute the topic similarity is based on the number of words that two topics share. These measures ignore that two words may be different in their lexicographic representation but semantically similar.

Average Jaccard Similarity (JS). The ratio of common words in two topics can be measured by using Jaccard Similarity [13].[1] The Jaccard Similarity (JS) between t_i and t_j is defined as follows:

$$JS(t_i, t_j) = \frac{|t_i \cap t_j|}{|t_i \cup t_j|} \tag{1}$$

This measure varies between 0 and 1, where 0 means that the topics are completely different, and 1 means that topics are similar to each other.

Rank-biased Overlap (RBO). To consider the ranking of the words, one can use Rank-Biased Overlap (RBO) [27], exploited in Bianchi et al. [5] in the topic modeling context. It is based on a probabilistic model in which a user compares the overlap of two ranked lists (that in our case correspond to two topics) at incrementally increasing depth. The user can stop to examine the lists at a given rank position according to the probability p, enabling therefore the metric to be top-weighted and consequently giving more weight to the top words of a topic. The smaller p, the more top-weighted the metric is. When $p = 0$, only the top-ranked word is considered. The metric ranges from 0 (completely different topic descriptors) to 1 (equal topic descriptors).

RBO is based on the concept of *overlap at depth* h between two lists, which is the number of elements that the lists share when only the first h words are considered. For example, the overlap at depth 2 between the lists $l_1 = \{cat, animal, dog\}$ and $l_2 = \{animal, kitten, animals\}$ is 1. The average overlap is defined as the proportion of the overlap at depth h over h. Therefore, the RBO measure when evaluating two topics is computed as the expected value of the average overlap that the user observes when comparing two lists.

Average Pairwise Pointwise Mutual Information (PMI) In [1], the authors present a similarity metric based on Pointwise Mutual Information (PMI). The authors adapt the PMI coherence to measure topic similarity by computing the average pairwise PMI between the words belonging to two topics. More formally, the PMI between the topics i and j is defined as:

$$PMI(t_i, t_j) = \frac{1}{t^2} \sum_{u \in t_i} \sum_{v \in t_j} PMI(u, v) \tag{2}$$

where t is the number of words of each topic.

[1] This approach has been used in [26] to compute the distance between topics.

2.2 Measures based on Probability Distributions

Instead of considering the top-words, we can consider the word-topic distribution to compute the distance between metrics. However, these metrics may be sensitive to the high dimensionality of the vocabulary [1].

Average Log Odds Ratio (LOR) In [11], the topic similarity is computed using the average log odds ratio (LOR) that is defined as follows:

$$LOR(\beta_i, \beta_j) = \sum_{v \in V} \mathbb{1}_{\mathbb{R}_{\neq 0}}(\beta_i(v))\mathbb{1}_{\mathbb{R}_{\neq 0}}(\beta_j(v))|\log(\beta_i(v) - \beta_j(v)| \tag{3}$$

where $\mathbb{1}_A(x)$ is an indicator function defined as 1 if $x \in A$ and 0 otherwise. This metric computes the distance between the distributions associated with two topics, so it is a dissimilarity metric.

Kullback-Leibler Divergence (KL-DIV). A widely used measure to determine the similarity between two topics is the Kullback-Leibler Divergence [22,2,25], which measures the distance from a given topic's distribution over words to another one. It is defined as follows:

$$KL - DIV(\beta_i, \beta_j) = \sum_{v \in V} \beta_i(V) \log \frac{\beta_i(v)}{\beta_j(v)} \tag{4}$$

Notice that this metric is not symmetric and its domain ranges from 0 (when two distributions are identical) to infinity. In fact, this metric represents a dissimilarity score. Other metrics based on computing the distance between distributions include the Jensen Shannon Divergence and the cosine similarity [1].

3 Word Embedding-based Similarity

To overcome the absence of semantics in the traditional similarity measures available in the state of the art, one can resort to the use of word embeddings to capture conceptual relationships between words. In the word embedding spaces, the vector representations of the words appearing in similar contexts tend to be close to each other [18]. We can therefore exploit the nature of word embeddings and define new metrics to estimate how much two topic descriptors are similar.

Word Embedding-based Centroid Similarity (WECS). The most simple strategy, originally designed in [6] for a cross-lingual task, consists of computing the centroids of two topic descriptors t_i and t_j and then estimating their similarity. Let be $\overrightarrow{t_i}$ the vector centroid of the topic descriptor t_i computed as the average of word embeddings considering all the words belonging to the topic i.

The Word Embedding-based Centroid Similarity between two topics is estimated as $WECS(t_i, t_j) = sim(\overrightarrow{t_i}, \overrightarrow{t_j})$, where sim is a measure of similarity between vectors, i.e. cosine similarity.

Word Embedding-based Pairwise Similarity (WEPS). An alternative to WECS consists of averaging the pairwise similarity between the embedding vectors of the words composing the topic descriptors. We define the similarity between two topic descriptors t_i and t_j as follows:

$$WEPS(t_i, t_j) = \frac{1}{t^2} \sum_{v \in t_i} \sum_{u \in t_j} sim(w_v, w_u) \tag{5}$$

where t represents the number of words of each topic, and w_v and w_u denote the word embeddings associated with words v and u respectively.

Word embedding-based Weighted Sum Similarity (WESS). A simple way to combine the probability distributions and the word embeddings is to compute the sum of the word embeddings of the words in the vocabulary, where the sum is weighted by the probability of each term in the topic. Then, we compute the similarity between the resulting word embeddings.

More formally, let be $b_i = \sum_{v \in V} \beta_i(v) \cdot w_v$ the weighted sum of the word embeddings of the vocabulary for the topic i. Therefore, the WESS for the topic i and j is defined as $sim(b_i, b_j)$.

Word Embedding-based Ranked-Biased Overlap (WERBO). We can extend RBO and define a new metric of similarity that is top-weighted and makes use of word embeddings. Given the lists $l_1 = \{cat, animal, dog\}$ and $l_2 = \{animal, kitten, animals\}$, the words *cat* and *kitten* are similar, even though they are lexicographically different. It follows that their overlap at depth 2 should be higher than 1. We therefore generalize the concept of overlap to handle word embeddings instead of simple word tokens.

Algorithm 1 Calculate generalized overlap at depth h

Input: t_i, t_j topic descriptors composed of n words; h depth of the list, where $h \leq n$

for $u := 1, \ldots, h$ **do**
  for $v := 1, \ldots, h$ **do**
    $sim[w_u^i, w_v^j] := similarity(w_u^i, w_v^j)$
  end for
end for
$overlap := 0$
while sim is not empty **do**
  $max\_value := max(sim)$
  $w_u^i, w_v^j := get\_indices(max\_value)$
  remove all entries of w_u^i and w_v^j from sim
  $overlap := overlap + max\_value$
end while
return $overlap$

Algorithm 1 shows how to compute the generalized overlap between two topic descriptors t_i and t_j. First of all, we compute the similarity between all the pairs of word embedding vectors w_u^i and w_v^j belonging to the two topics i and j (line 1-5). The associative array sim (line 3) is indexed by the tuple (w_u^i, w_v^j) and contains all the computed similarities. Subsequently (line 7-12), we process the associative array sim to get the words that are the most similar, to then update the overlap variable. In particular, the algorithm searches for the tuple (w_u^i, w_v^j) that has the highest similarity in sim (line 8), removes from sim all the entries containing w_u^i or w_v^j (line 9-10) and finally updates the overlap by adding the highest similarity value corresponding to the tuple (w_u^i, w_v^j) (line 12). For example, let us compute the generalized overlap at depth 3 of the word lists $l_1 = \{cat, animal, dog\}$ and $l_2 = \{animal, kitten, animals\}$. The result will be $sim(animal, animal) + sim(cat, kitten) + sim(animals, dog)$, because $(animal, animal)$ are identical vectors and should be summed first, then $(cat, kitten)$ are the second most similar vectors, and finally $(animals, dog)$ are the remaining vectors and should be summed at last.

In the proposed algorithm, $similarity(w_u^i, w_v^j)$ is the angular similarity between the vectors associated with the word embeddings related to the words u and v respectively[2]. Notice that this approach is based on a greedy strategy that estimates the overlapping by considering first the most similar embeddings of the words available in the top-h list. We will then refer to this approach as **WERBO-M**. Instead of computing the similarity between each word embedding, an alternative metric can compute the centroid of the embeddings at depth h. In this way, the overlap at depth h is just defined as $similarity(\overrightarrow{t_i}, \overrightarrow{t_j}) \cdot h$, where $\overrightarrow{t_i}$ and $\overrightarrow{t_j}$ are the centroids of the topics t_i and t_j respectively. We will refer to this metric as **WERBO-C**.

Weighted Graph Modularity (WGM) We can rethink two topic descriptors in the form of a graph. Each word represents a node in the graph, while the edges denote the similarity between the words. Considering two topics composed of their own words (nodes), the intra-topic similarity connections should be higher than the extra-topic similarity connections with any other topic. We can express this idea by using the measure of modularity, which estimates the strength of division of a graph into modules (in our case, topics).

Let $G = (U, E)$ be a fully connected graph, where U is the words related to t_i and t_j and E are weighted edges denoting the similarity between pairs of word embeddings. In particular, an edge weight is defined as $A_{uv} = sim(w_v, w_u)$, where $(u, v) \in E$, $v, u \in U$ and $sim(\cdot, \cdot)$ is the angular similarity between two word embeddings. Given the graph G, originating from two topic descriptors t_i and t_j, the Weighted Graph Modularity (WGM) can be estimated as:

$$WGM(t_i, t_j) = \frac{1}{2m} \sum_{v,u \in U(G)} [A_{vu} - \frac{k_v k_u}{2m}] \mathbb{1}_{vu} \tag{6}$$

[2] We use the angular similarity instead of the cosine because we require the overlap to range from 0 to 1.

where k_v and k_u denote the degrees of the nodes v and u respectively, m is the sum of all of the edge weights in the graph, and $\mathbb{1}_{vu}$ is an indicator function defined as 1 if v and u are words belonging to the same topic, 0 otherwise. Modularity ranges from -1/2 (non-modular topics) to 1 (fully separated topics). Therefore, it should be considered as a dissimilarity score.

4 Experimental Investigation

4.1 Experimental Setting

Compared measures. Before proceeding with the description of the validation strategy and the performance measures adopted for a comparative evaluation, we summarize the investigated measures. In particular, in Table 1 we provide details about all the metrics, reporting their main features:

- TD, which denotes if the metric considers the top-t words of the descriptors;
- PD, that reports if the metric considers the topic probability distribution;
- WE, which indicates if the metric overcomes the limitation of the discrete representation of words by using Word Embeddings;
- TW, that identify if the metric is top-weighted, i.e. the words at the top of the ranked list are more important than the words in the tail.

The implementations of the measures are integrated into the topic modeling framework OCTIS [23], available at `https://github.com/mind-lab/octis`.

Similarity/Distance Measure	TD	PD	WE	TW
Jaccard Similarity (JS) [26]	✓			
Rank-biased Overlap (RBO) [27]	✓			✓
Pointwise Mutual Information (PMI) [1]	✓			
Average Log Odds Ratio (LOR) [11]		✓		
Kullback-Leibler Divergence (KL-DIV) [22]		✓		
Word embedding-based Centroid Similarity (WECS)	✓		✓	
Word Embedding Pairwise Similarity (WEPS)	✓		✓	
Word Embedding-based Weighted Sum Similarity (WESS)		✓	✓	
Word Embedding-based RBO - Match (WERBO-M)	✓		✓	✓
Word Embedding-based RBO - Centroid (WERBO-C)	✓		✓	✓
Weighted Graph Modularity (WGM)	✓		✓	

Table 1: Summary of the characteristics of the metrics presented in this paper. The newly proposed metrics are reported in bold.

Validation strategy. To validate the proposed similarity measures, and compare them with the state-of-the-art ones, we selected the most widely adopted topic model to produce a set of topics to be evaluated. In particular, we trained Latent Dirichlet Allocation (LDA) [8] on two benchmark datasets, i.e. BBC news

[16] and 20 NewsGroups.[3], originating 50 different topics per dataset.[4] For the pre-processing, we removed the punctuation and the English stop-words[5], and we filtered out the less frequent words, obtaining a final vocabulary of 2000 terms.

Given the topics extracted by LDA, we disregarded those with a low value of topic coherence, measured by using Normalized Pointwise Mutual Information (NPMI) [17] on the dataset itself as a reference corpus. Then we randomly sampled 100 pairs of topics (for each dataset) that have been evaluated by three annotators, by considering the top-10 words. In particular, the annotators have rated if two topics were related to each other or not, using a value of 0 (not related topics) and 1 (similar topics). The final annotation of each pair of topics has been determined according to a majority voting strategy on the rates given by the three annotators.

For the metrics that are based on the topic descriptors, we considered the top-10 words of each topic. Regarding the metrics that are based on word embeddings, we used Gensim's[6] Word2Vec model to compute the embedding space on the corpus with the default hyperparameters. The co-occurrence probabilities for the estimation of PMI have been computed on the training dataset. For the metrics that represent dissimilarity scores, such as KL-DIV, the LOR and WGM metrics, we considered their inverse.

Performance Measures. We evaluated the capabilities of all the topic similarity metrics, both the ones available in the state of the art and the proposed ones, by measuring Precision@k, Recall@k and F1-Measure@k.

In particular, Precision@k (P@k) is defined as the fraction of the number of retrieved topics among the top-k retrieved topics that are relevant and the number of retrieved topics among the top-k retrieved topics. Recall@k (R@k) is defined as the fraction of the number of retrieved topics among the top-k retrieved topics that are relevant and the total number of relevant topics. F1-Measure@k (F1@k) is defined the harmonic mean between P@k and R@k, i.e. $F1@k = 2(P@k \cdot R@k)/(P@k + R@k)$.

4.2 Experimental Results

Table 2 shows the results for the BBC News dataset in terms of P@k, R@k and F1@k by varying k for 1 to 5. As a first remark, we can see that the metrics that are based on the shared word tokens only, i.e. the Jaccard Distance (JD) and Rank-biased Overlap (RBO), achieve the lowest performance. KL-DIV and LOR, which are based only on the topic-word probability distributions, outperform the baselines JD and RBO, but they are not able to outperform the proposed measures that consider the word embeddings similarities. The most

[3] `http://people.csail.mit.edu/jrennie/20Newsgroups/`

[4] We trained LDA with the default hyperparameters of the Gensim library.

[5] We used the English stop-words list provided by MALLET: `http://mallet.cs.umass.edu/`

[6] `https://radimrehurek.com/gensim/`

		State-of-the-art metrics					Proposed metrics					
	k	JD	RBO	PMI	LOR	KL-DIV	WESS	WEPS	WECS	WERBO-M	WERBO-C	WGM
P@K	1	0.818	0.864	0.955	0.846	0.909	0.909	0.955	**1.000**	**1.000**	**1.000**	0.818
	2	0.727	0.705	**0.864**	0.769	0.750	0.795	0.841	0.841	**0.864**	**0.864**	0.795
	3	0.652	0.667	0.803	0.667	0.652	0.742	0.788	0.773	**0.818**	0.788	0.773
	4	0.557	0.557	0.705	0.596	0.602	0.682	0.705	0.693	**0.716**	**0.716**	0.693
	5	0.482	0.491	0.573	0.492	0.536	0.573	**0.582**	**0.582**	**0.582**	**0.582**	0.573
	avg	0.647	0.657	0.706	0.674	0.690	0.740	0.774	0.778	**0.796**	0.790	0.730
R@K	1	0.348	0.364	0.417	0.423	0.402	0.409	0.417	**0.439**	**0.439**	**0.439**	0.379
	2	0.545	0.534	**0.663**	0.641	0.587	0.614	0.648	0.648	0.659	**0.663**	0.621
	3	0.697	0.712	0.871	0.776	0.716	0.803	0.856	0.833	**0.879**	0.845	0.833
	4	0.784	0.784	0.977	0.885	0.848	0.951	0.977	0.966	**0.989**	**0.989**	0.966
	5	0.867	0.879	0.989	0.910	0.932	0.985	**1.000**	**1.000**	**1.000**	**1.000**	0.989
	avg	0.648	0.655	0.783	0.727	0.697	0.752	0.780	0.777	**0.793**	0.787	0.758
F1@K	1	0.456	0.479	0.539	0.521	0.517	0.524	0.539	**0.570**	**0.570**	**0.570**	0.480
	2	0.589	0.574	**0.708**	0.651	0.617	0.650	0.689	0.689	0.705	**0.708**	0.656
	3	0.644	0.660	0.798	0.675	0.645	0.734	0.783	0.765	**0.809**	0.777	0.765
	4	0.627	0.627	0.786	0.677	0.673	0.762	0.786	0.775	**0.797**	**0.797**	0.775
	5	0.595	0.605	0.698	0.610	0.654	0.697	**0.709**	**0.709**	**0.709**	**0.709**	0.698
	avg	0.582	0.589	0.706	0.627	0.621	0.673	0.701	0.701	**0.718**	0.712	0.675

Table 2: Precision@K, Recall@K and F1-Measure@k on the BBC News dataset.

competitive metric with respect to the proposed ones is the PMI, which obtains comparative results to the word-embedding metrics for $k = 2$. These results suggest that considering a richer representation of topical words helps in retrieving semantically similar topics to a given target topic. In particular, WERBO-M and WERBO-C reach the highest scores in most of the cases. This means that not only the meaning of the words are important when evaluating the similarity of two topics, but also the position of each word in the topic matters. In fact, WERBO-M and WERBO-C outperform the metrics WEPS and WECD that do not take into consideration the rank of the words.

Table 3 reports the results on the 20NewsGroups dataset. Here, the obtained results are similar to the previous dataset. All the word embedding-based metrics outperform the state-of-the-art ones. In particular, WERBO-C outperforms the other metrics or obtain comparable results in most the cases. Even if WESS is the similarity metric that obtains the best performance on average, the results obtained by WERBO-C and WERBO-M are definitely comparable. Also on this dataset PMI seems to be the most competitive metric, however the word-embedding metrics metrics outperform it in most of the cases.

We report in Table 4 two examples of topics evaluated by the considered similarity/distance measures. The first example reports two topics, that clearly represent two distinct themes, likely *religion* and *technology*. In this case, all the proposed metrics can capture the diversity of the two topics as well as the measure of the state of the art. On the other hand, the second example reports two related topics about *technology*. We can easily notice that while all the

		State-of-the-art metrics					Proposed metrics					
	k	JD	RBO	PMI	LOR	KL-DIV	WESS	WEPS	WECS	WERBO-M	WERBO-C	WGM
P@K	1	0.833	0.833	**1.000**	0.833	0.833	**1.000**	**1.000**	0.958	0.958	0.917	0.958
	2	0.646	0.667	0.813	0.792	0.792	**0.833**	0.813	**0.833**	0.813	**0.833**	**0.833**
	3	0.569	0.569	0.681	0.653	0.667	0.694	0.694	0.694	**0.708**	**0.708**	0.694
	4	0.458	0.458	0.583	0.563	0.583	0.583	0.583	0.583	**0.604**	**0.604**	0.583
	5	0.408	0.408	0.492	0.492	0.492	**0.500**	**0.500**	**0.500**	**0.500**	**0.500**	**0.500**
	avg	0.583	0.587	0.714	0.666	0.673	**0.722**	0.718	0.714	0.717	0.713	0.714
R@K	1	0.424	0.424	**0.542**	0.375	0.396	**0.542**	**0.542**	0.500	0.500	0.459	0.500
	2	0.581	0.591	0.758	0.667	0.737	**0.779**	0.758	**0.779**	0.758	0.772	**0.779**
	3	0.705	0.701	0.869	0.793	0.848	0.890	0.890	0.890	**0.904**	**0.904**	0.890
	4	0.734	0.734	0.950	0.866	0.950	0.950	0.950	0.950	**0.974**	**0.974**	0.950
	5	0.807	0.807	0.974	0.946	0.974	**0.988**	**0.988**	**0.988**	**0.988**	**0.988**	**0.988**
	avg	0.650	0.651	0.819	0.730	0.781	**0.830**	0.825	0.821	0.825	0.819	0.821
F1@K	1	0.522	0.522	**0.653**	0.487	0.501	**0.653**	**0.653**	0.612	0.612	0.570	0.612
	2	0.566	0.580	0.727	0.681	0.706	**0.748**	0.727	**0.748**	0.727	0.744	**0.748**
	3	0.587	0.585	0.709	0.670	0.692	0.725	0.725	0.725	**0.739**	**0.739**	0.725
	4	0.527	0.527	0.674	0.640	0.674	0.674	0.674	0.674	**0.696**	**0.696**	0.674
	5	0.510	0.510	0.610	0.607	0.610	**0.621**	**0.621**	**0.621**	**0.621**	**0.621**	**0.621**
	avg	0.542	0.545	0.675	0.617	0.637	**0.684**	0.680	0.676	0.679	0.674	0.676

Table 3: Precision@K, Recall@K and F1-Measure@k on 20 NewsGroups.

measures of the state of the art suggest that the two topics are completely different because of their low values (e.g. JS = 0.053 and KL-DIV = -4.415), the proposed metrics can capture their actual similarity.

Topic 1	Topic 2	Metrics	Topic 1	Topic 2	Metrics
god	ftp	JS=0	tiff	window	JS=0.053
christian	fax	RBO=0	gif	application	RBO=0.057
christianity	pub	PMI=-0.042	image	manager	PMI=0.327
religion	graphics	LOR=-3.204	format	display	LOR=-2.110
faith	computer	KL-DIV=-4.36416	jpeg	color	KL-DIV=-4.415
christ	software	WESS=-0.145	formats	widget	WESS=0.787
sin	version	WEPS=-0.0941	color	mouse	WEPS=0.402
people	mail	WECS=-0.183	images	screen	WECS=0.565
view	gov	WERBO-M=0.472	complex	button	WERBO-M=0.651
paul	mit	WERBO-C=0.120	resolution	user	WERBO-C=0.170
		WGM=-0.102			WGM=-0.015
Ground Truth = unrelated topics			Ground Truth = similar topics		

Table 4: Qualitative comparison of the considered measures. Since KL-DIV, LOR and WGM represent dissimilarity scores, they are reported as their inverse.

5 Conclusions

In this paper, we investigated and compared several topic similarity metrics. These measures are particularly useful for data analysis tasks [10,19], i.e. when

a user may want to identify topics that are similar for the theme of interest. We proposed several metrics that exploit word embeddings and take into account the ranking of words in the topic descriptors. We experimentally proved that the proposed metrics outperform the state-of-the-art ones. We believe that these metrics should be considered in topic modeling visualization tools [11,12,22,15,23] for improving their performance and allow a user to obtain relevant results. As future work, different word embeddings methods could be investigated, also considering the word embeddings deriving from the state-of-the-art contextualized language models, e.g. BERT [14].

References

1. Aletras, N., Stevenson, M.: Measuring the similarity between automatically generated topics. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. pp. 22–27 (2014)
2. AlSumait, L., Barbará, D., Gentle, J., Domeniconi, C.: Topic significance ranking of lda generative models. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 67–82. Springer (2009)
3. Batmanghelich, K., Saeedi, A., Narasimhan, K., Gershman, S.: Nonparametric spherical topic modeling with word embeddings. In: Proceedings of the conference. Association for Computational Linguistics. vol. 2016, p. 537 (2016)
4. Belford, M., Namee, B.M., Greene, D.: Ensemble topic modeling via matrix factorization. In: Proceedings of the 24th Irish Conference on Artificial Intelligence and Cognitive Science, AICS 2016. vol. 1751, pp. 21–32 (2016)
5. Bianchi, F., Terragni, S., Hovy, D.: Pre-training is a hot topic: Contextualized document embeddings improve topic coherence. In: Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021). Association for Computational Linguistics (2021)
6. Bianchi, F., Terragni, S., Hovy, D., Nozza, D., Fersini, E.: Cross-lingual contextualized topic models with zero-shot learning. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2021. pp. 1676–1683 (2021)
7. Blei, D.M.: Probabilistic topic models. Communications of the ACM **55**(4), 77–84 (2012)
8. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. **3**, 993–1022 (2003)
9. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics **5**, 135–146 (2017)
10. Boyd-Graber, J.L., Hu, Y., Mimno, D.M.: Applications of topic models. Found. Trends Inf. Retr. **11**(2-3), 143–296 (2017)
11. Chaney, A.J., Blei, D.M.: Visualizing topic models. In: Proceedings of the 6th International Conference on Weblogs and Social Media. The AAAI Press (2012)
12. Chuang, J., Manning, C.D., Heer, J.: Termite: visualization techniques for assessing textual topic models. In: International Working Conference on Advanced Visual Interfaces, AVI 2012. pp. 74–77. ACM (2012)

13. Deng, F., Siersdorfer, S., Zerr, S.: Efficient jaccard-based diversity analysis of large document collections. In: Proceedings of the 21st ACM international conference on Information and knowledge management. pp. 1402–1411 (2012)
14. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019. pp. 4171–4186 (2019)
15. Gardner, M.J., Lutes, J., Lund, J., Hansen, J., Walker, D., Ringger, E., Seppi, K.: The topic browser: An interactive tool for browsing topic models. In: Nips workshop on challenges of data visualization. vol. 2, p. 2 (2010)
16. Greene, D., Cunningham, P.: Practical solutions to the problem of diagonal dominance in kernel document clustering. In: Proc. 23rd International Conference on Machine learning (ICML'06). pp. 377–384. ACM Press (2006)
17. Lau, J.H., Newman, D., Baldwin, T.: Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014. pp. 530–539 (2014)
18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. pp. 3111–3119 (2013)
19. Newman, D.J., Block, S.: Probabilistic topic decomposition of an eighteenth-century american newspaper. J. Assoc. Inf. Sci. Technol. **57**(6), 753–767 (2006)
20. Nguyen, D.Q., Billingsley, R., Du, L., Johnson, M.: Improving topic models with latent feature word representations. Transactions of the Association for Computational Linguistics **3**, 299–313 (2015)
21. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
22. Sievert, C., Shirley, K.: Ldavis: A method for visualizing and interpreting topics. In: Proceedings of the workshop on interactive language learning, visualization, and interfaces. pp. 63–70 (2014)
23. Terragni, S., Fersini, E., Galuzzi, B.G., Tropeano, P., Candelieri, A.: OCTIS: comparing and optimizing topic models is simple! In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, EACL 2021. pp. 263–270 (2021)
24. Terragni, S., Fersini, E., Messina, E.: Constrained relational topic models. Inf. Sci. **512**, 581–594 (2020)
25. Terragni, S., Nozza, D., Fersini, E., Messina, E.: Which matters most? comparing the impact of concept and document relationships in topic models. In: Proceedings of the First Workshop on Insights from Negative Results in NLP, Insights 2020. pp. 32–40 (2020)
26. Tran, N.K., Zerr, S., Bischoff, K., Niederée, C., Krestel, R.: Topic cropping: Leveraging latent topics for the analysis of small corpora. In: Research and Advanced Technology for Digital Libraries - International Conference on Theory and Practice of Digital Libraries, TPDL 2013. vol. 8092, pp. 297–308. Springer (2013)
27. Webber, W., Moffat, A., Zobel, J.: A similarity measure for indefinite rankings. ACM Trans. Inf. Syst. **28**(4), 20:1–20:38 (2010)