Applied Intelligence (2022) 52:17829–17844
https://doi.org/10.1007/s10489-022-03388-5

Probabilistic topic modeling for short text based on word embedding networks

Marcelo Pita[1,2] · **Matheus Nunes**[1] · **Gisele L. Pappa**[1]

Accepted: 14 February 2022 / Published online: 6 April 2022

Abstract

Uncovering topics in short texts can be an arduous task. The inadequacy of general-purpose topic models for handling short documents may be explained by the difficulty in dealing with scarce context information. A variety of strategies have been proposed to address this problem, such as using application-specific information, generating larger pseudo-documents, or modeling a single topic per document. This paper introduces a novel strategy to solve this problem named *Vec2Graph Topic Model (VGTM)*. It creates a graph-based representation for the analyzed corpus using word embeddings, named *Vec2Graph*, and infers topics from overlapping communities patterns on this graph. *Vec2Graph* leverages the semantics of word embeddings to create a dense similarity graph of words, mitigating the lack of context in short text documents. Experiments evaluating topic coherence quality in four benchmarks and two real-world datasets show that VGTM achieves the best overall results (obtaining statistically better results in 10 out of 18 experiments) in comparison with standard and state-of-the-art short text topic models. We also analyze the relationship between one of our evaluated metrics – NPMI – and structural patterns in the Vec2Graph representation. We found that networks that present a strong community structure tend to present higher NPMI values, suggesting the possibility of direct measurement and potential control of topic coherence through these network features.

Keywords Topic modeling · Short text · Word embeddings · Complex networks

1 Introduction

Topic modeling is the task of finding a useful structure in unstructured collections of text by discovering patterns of word use in documents [1]. In a world where unstructured text is becoming naturally shorter – e.g., microblogging messages (tweets), short message service (SMS), questions and answers (Q&A) – topic modeling methods, initially designed to deal with long text documents, are being adapted to deal with new challenges related to the scarcity of context in shorter texts.

Short text document collections are defined in the literature as those where documents have only a few words but a large corpus vocabulary, producing highly sparse document-term matrices [2]. In practice, this low word co-occurrence in documents makes tasks such as classification, clustering, and topic discovery much more challenging [3].

Regarding topic modeling, the most well-known methods in the literature, including Latent Dirichlet Allocation (LDA) [4] and Non Negative Matrix Factorization (NMF) [5], present limitations when dealing with short texts [6]. These shortcomings were addressed by proposing adaptations to the original methods, which include modifications of the original probabilistic graphical model [7], restrictions in the number of topics per document [8], or the use of additional information to enhance document context [9, 10].

However, one of the most successful trends in topic modeling for short text is to take advantage of the semantic representation provided by word embeddings [11–13]. Word embeddings are vector representations that

✉ Marcelo Pita
marcelo.pita@dcc.ufmg.br

Matheus Nunes
mhnnunes@dcc.ufmg.br

Gisele L. Pappa
glpappa@dcc.ufmg.br

[1] Computer Science Department, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

[2] Serviço Federal de Processamento de Dados, Brasília, Brazil

encode the semantics of words in a vector space [14, 15], where operations with word vectors are expected to be equivalent to the semantic manipulation of words. As word embeddings capture the semantics of words by analyzing their *context* – which is an essential but most of the time absent information in short text documents – the idea is to enrich the topic modeling process with context information brought by this type of representation.

Among the methods for short text topic modeling that explore word vectors we highlight Latent-Feature LDA (LF-LDA) [16], the distributed representation expansion method (DREx) [11], Generalized Pólya Urn Dirichlet Multinomial Mixture (GPU-DMM) [17], Semantics-assisted Non-negative Matrix Factorization (SeaNMF) [18], and Clu-Words (Clusters of Words) [19]. However, none of these methods explicitly exploit global patterns that can emerge from the semantic relationship among word embeddings representing the entire corpus.

In this direction, this paper introduces VGTM (Vec2Graph topic model), a method that performs topic analysis at the corpus level – an approach introduced by classical short text topic models [7, 20] as an attempt to overcome the problem of context scarcity in small documents – instead of the document level and infers the context of words by using word embeddings.

VGTM presents a novel abstraction for the discovery of patterns in semantic word graphs by using the concept of graph communities. It is grounded on the idea that, given a word embedding graph, *communities* or clusters of words in this graph are good topic discriminators. In the proposed word embedding graph, nodes represent words of the corpus vocabulary, and any arbitrary edge between two nodes (words) w_1 and w_2 represents a semantic relationship between them. Further, edges are weighted according to the similarity between the word vector representations of w_1 and w_2. However, the graph is pruned by a weight threshold parameter, which tunes graph connectivity according to the application.

VGTM can be seen more as a framework than as a method, allowing the usage of any word embedding representation and similarity metric to generate the semantic graph. For example, in the experiments reported in this paper, we used Skip-Gram [21] and cosine similarity. As usual, topics are modeled as word clusters. However, as topics can share words, VGTM detects communities that can overlap [22].

A standard and objective way of evaluating topics is through topic coherence[1]. Experiments show that VGTM obtains competitive NPMI (normalized pointwise mutual information) and topic coherence (C_P) results when compared to seven state-of-the-art baseline methods in a variety of benchmarks and two real-world application datasets.

The main contributions of this paper are:

- a graph representation for a document corpus, where node edges are weighted according to distances in the word vector space;
- a topic modeling method that explores the idea of overlapping communities to extract topics from the proposed graphs, where we formalize the posterior probabilities of topics per word, words per topic, topics, and topics per document;
- an analysis of the structural properties of the corpus graphs and their correlation with a topic modeling quality metric;
- results that are competitive with or better than the methods considered state-of-the-art in 10 out of 18 experiments in six datasets.

This paper is organized as follows. Section 2 reviews the main contributions in the field of topic modeling for short text. Sections 3 and 4 introduce Vec2Graph and VGTM, respectively. Section 5 details the experiments carried out and results obtained, both in topic coherence quality and its correlation with structural properties of the corpus graphs. Finally, Section 6 summarizes the contributions, reports conclusions, and discusses future work.

2 Related work

Topic modeling methods for short text can be organized into three major categories: (i) probabilistic graphical models; (ii) methods that generate larger pseudo-documents from short text to artificially add context to the documents; and (iii) methods based on matrix factorization. Also note that we are not including deep learning topic models in these categories [24–27], since the well-established and state-of-the-art methods reported in the literature are still based on variations of traditional latent space models and the generation of pseudo-documents [13].

Probabilistic graphical models Methods in the first category are mainly modifications of the popular LDA [4] algorithm with specializations for short text. LDA is a Bayesian network designed to discover topics on corpora by assuming a simple generative process for document writing. In LDA, topics are defined as distributions of probability over words, documents are mixtures of topics, and words are selected from topics. Only words are observable variables and priors are Dirichlet distributions. The main objective of the algorithm is to infer the hidden (or latent) variables, such as topics proportions of documents and words distributions per

[1] There are other approaches, such as visual qualitative analysis, indirect evaluation (e.g., document classification) and topic diversity [23].

topic. Aiming to overcome the difficulty of LDA with short text [6], other LDA-based methods have been proposed for short text, such as BTM [7], LF-LDA [16], SATM [28], LTM [29], PTM [9], and GPU-DMM [17].

Among the aforementioned methods, Generalized Pólya Urn Dirichlet Multinomial Mixture (GPU-DMM) is considered state-of-the-art among probabilistic graphical models. It combines the Dirichlet multinomial mixtures (DMM) model [8] with the GPU (generalized Pólya urn) mechanism [30]. It is a specialization of LDA that models a single topic per document. The GPU mechanism is incorporated into the Gibbs sampling process of DMM, enhancing the likelihood of selecting other words that are semantically related to the sampled ones. GPU-DMM uses word vectors for this purpose.

Apart from the methods listed above, most of the recent works in topic modeling for short text take advantage of word embeddings. Das et al. [31], for example, propose a method based on LDA that represents each document as a sequence of word vectors. Shi et al. [32] propose a method that learns both word embeddings and the topic model simultaneously. Li et al. [33], in turn, propose Relational Biterm Topic Model (R-BTM), an extension of BTM that links short texts with the cosine similarity of word embeddings. Tuan et al. [34] proposes the Bag of Biterms Model (BBM), which represents documents as a bag of biterms, expanding the length of the document and providing more context for topic models such as LDA-B (a variation of LDA that handles biterms).

Pseudo-document-based models The second category of topic models for short text is composed of methods for generating large pseudo-documents. Hong et al. [20] and Mehrotra et al. [35] first introduced pseudo-document methods for Twitter topic modeling that use different aggregation schemes, such as pooling by author and hashtag. Still in this category, Zuo et al. [10] presented WNTM, which creates an undirected graph of words whose edges are weighted by word co-occurrence. This means that words that co-occur at least once in a user-defined sliding window will have an edge. It is important to note that this process maps the domain of documents to the domain of words. In this word domain, WNTM creates a pseudo-document for each word in the graph. This document is formed by all words in the neighborhood of the central word it represents.

Bicalho et al. [11] introduced a distributed representation expansion method called DREx, which uses word embeddings to expand documents. DREx expands documents by probabilistic adding words that have the most similar vector representations to the bigrams in the original document. Qiang et al. [36], in contrast, proposed Embedding-based Topic Model (ETM), which aggregates short texts using word embeddings generated from an external large corpus and then infers latent topics with a Markov Random Field Regularized (MRF) model [37]. A similar approach was proposed by Gao et al. [38], generating the Conditional Random Field regularized Topic Model (CRFTM).

Matrix factorization-based models The third category of methods of short text includes those based on matrix factorization. Among these methods we highlight the Semantics-assisted Non-negative Matrix Factorization (SeaNMF) method [18]. SeaNMF is based on NMF and factorizes both document-term frequency and term-term co-occurrence matrices to discover topics.

Within this same category, Rashid et al. [39] explored principal component analysis and a fuzzy approach (fuzzy c-means) over local and global term frequencies to overcome the sparsity problem of short text. Viegas et al. [19], in turn, proposed a word embedding-based document representation named CluWords, which uses the similarity between pairs of word vectors previously learned from a large external dataset (e.g. Wikipedia). Each word in the dataset vocabulary V is mapped to a CluWord, a vector of size $|V|$ whose values are the cosine similarity between the correspondent word vectors above a threshold α (otherwise, it is zero). CluWords are combined with NMF for topic modeling.

Our contribution As previously mentioned, when compared to other state-of-the-art methods for short text topic modeling, VGTM performs topic analysis at the level of the corpus – as BTM and WNTM –, exploits word embedding-based semantics – as DREx, GPU-DMM, SeaNMF, and CluWords –, but it introduces a novel way, to the best of our knowledge, for discovering patterns in semantic word graphs. It uses the concept of overlapping graph communities to find topics in short texts. For that, a complete framework for obtaining the required posterior probabilities of documents, topics, words, and their combinations is formalized.

3 Vec2Graph

Word vectors carry information about the semantics of words in a certain learning context, given by the dataset they are trained on [14, 15]. The cosine of the angle θ_{ij} between two word vectors $\mathbf{v}_i$ and $\mathbf{v}_j$, with $\mathbf{v}_i, \mathbf{v}_j \in \mathcal{W}$ – the set of word vectors for the corpus C – is a standard estimator of the degree of similarity between words i and j [14]. Based on this estimator, we propose a graph-based text representation named Vec2Graph, which induces a word embedding network $\mathcal{G}_C$ for a corpus C from the cosine of the angle ($\cos(\theta_{ij})$) between every pair of words $i, j \in C$.

Table 1 Notation used to describe Vec2Graph and VGTM

Symbol	Description
C	corpus
V	corpus vocabulary
d	a document in C
w	a word in V
$\mathcal{W}$	word embeddings for C
$\mathbf{v}_i$	vector representation for word i
θ_{ij}	angle between two word vectors $\mathbf{v}_i$ and $\mathbf{v}_j$
$\mathcal{G}_C$	a graph representing the corpus C
S	edges in $\mathcal{G}_C$
s_{ij}	an edge between words i and j
w_{ij}	weight of s_{ij}, equals to $\cos\theta_{ij}$
σ	similarity threshold
k	Number of topics
t	a topic
$\mathbf{G}$	$\|V\| \times \|V\|$ adjacency matrix of $\mathcal{G}_C$
$\mathbf{X}$	$\|V\| \times k$ topic affinity matrix
$\mathbf{A}$	$k \times \|V\|$ topic-word affinity matrix

Graph-based text representations [40] have been proposed in the literature and applied to a variety of machine learning problems, including document classification [41], text summarization [42], and information retrieval [43, 44]. In Vec2Graph, nodes represent words and edges semantic relationships between nodes [45]. Documents can be seen as sub-graphs of this larger vocabulary graph.

Before we start to describe the proposed algorithms, Table 1 summarizes the notation used.

Let $\mathcal{G}_C(V, S)$ be a corpus graph, where V is the corpus vocabulary, represented in the graph by the set of nodes – each node labeled by a unique word in V – and S the set of edges – each edge s_{ij} indicating a semantic relationship between nodes i and j. Edges are weighted by the cosine similarity between the correspondent word vectors $\mathbf{v}_i$ and $\mathbf{v}_j$, as shown in (1).

$$w_{ij} = \cos\theta_{ij} = \cos(\mathbf{v}_i \cdot \mathbf{v}_j / \{||\mathbf{v}_i|| \cdot ||\mathbf{v}_j||\}) \quad (1)$$

Vec2Graph[2] imposes a restriction on the graph edges:

$$s_{ij} \in S \iff w_{ij} \geq \sigma, \quad (2)$$

where σ is a similarity threshold that defines the smallest acceptable similarity between a pair of words. The use of σ avoids the generation of a fully connected graph, which makes this new representation computationally cheaper to any posterior analysis and ultimately regulates the clustering level of the network. The clustering level is an important property explored by the proposed topic model described in the next section. The parameter σ must be optimized for each corpus and target application.

Nevertheless, the restriction in (2) produces a graph $\mathcal{G}_C$ that is potentially disconnected, i.e., with more than one connected component. In this case, we relax this restriction by restoring edges that were previously eliminated (i.e., edges with weight bellow σ) but that are necessary to make $\mathcal{G}_C$ connected. The edges restored are those with the highest weights and that connect components. This property ensures that there is a path between every pair of words in $\mathcal{G}_C$, avoiding that distances between disconnected words are infinite or unrepresented distances. The process followed by Vec2Graph to generated $\mathcal{G}_C$ is summarized in Algorithm 1.

Algorithm 1 VEC2GRAPH.

Require: $V, \mathcal{W}, \sigma$
1: $S \leftarrow \emptyset$ ▷ Set of edges (initially empty)
2: **for** $(\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{W}, i \neq j \wedge i, j \in V$ **do** ▷ Pairs of word vectors
3: $w_{ij} = \cos\theta_{ij}$ ▷ (1)
4: **if** $w_{ij} \geq \sigma$ **then** ▷ (2)
5: $S \leftarrow S + \{w_{ij}\}$
6: $\mathcal{G}_C \leftarrow$ BuildGraph(V, S) ▷ Resulting $\mathcal{G}_C$ graph

Figure 1 shows the structure of a corpus graph generated by Vec2Graph for the Sanders dataset (tweets related to IT companies)[3], with Skip-Gram word vectors of 1,000 dimensions and $\sigma = 0.4$ (similarity threshold)[4].

4 Vec2Graph topic model

Vec2Graph Topic Model (VGTM)[5] leverages the community structure of the corpus graph $\mathcal{G}_C$ by learning a probabilistic topic model based on its structural patterns. It is grounded on the hypothesis that communities [46] in the Vec2Graph structure are discriminators of topics. The rationale behind this approach is that, while it is natural that in social network analysis we understand the link between two "agents" (usually humans) in a graph as an indicator of social affinity, in a network of words derived from word embeddings, links can be naturally interpreted as *semantic affinity*.

Community detection is the problem of finding high level affinity patterns among nodes in any complex network. Therefore, in VGTM communities are groups of semantically

[2]Code available at: https://github.com/marcelopita/vec2graph_paper (2021/01/01).

[3]Dataset available at: https://github.com/marcelopita/datasets/blob/master/sanders.csv (2021/01/01)

[4]An interactive version of this graph as available at: https://homepages.dcc.ufmg.br/~marcelo.pita/vec2graph/corpus_graph.html (2021/01/01)

[5]Code available at: https://github.com/marcelopita/vgtm (2021/01/01).

Fig. 1 Corpus graph generated by Vec2Graph for the Sanders dataset. Colors are mapped to topics (in this case, represented by the labels from the dataset), shown in each node as slices of a pie chart

related words, used as an approximation for latent topics in a corpus. Algorithm 2 describes the main steps of VGTM. It receives as input the corpus graph $\mathcal{G}_C$, the number k of communities/topics to be found, the vocabulary V and the corpus C. It then runs a community detection algorithm which looks for overlapping communities in the adjacency matrix of $\mathcal{G}_C$, named G (line 3). The community detection algorithm returns a matrix A of size $k \times |V|$, where each cell in the matrix shows the membership of a word (column) in a topic/community (row).

VGTM then interprets communities as discriminators for topics, which in turn are described as probability distributions over all words in the corpus vocabulary. Hence, based on a normalized version of matrix A, VGTM provides the posterior probability distribution of words per topic (lines 10 to 12 of Alg. 2) and topics per document (lines 13 to 14 of Alg. 2), as detailed next.

Before going into details in VGTM, it is interesting to point out that Skianis et al. [47] interpret unsupervised methods that discover structural patterns in text representations (e.g. clusters of words) as regularizers, since group information can be used to avoid overfitting and increase the quality of the results of machine learning tasks, such as text classification. The authors discriminate four types of structured regularizers for NLP: (1) statistical regularizer, which uses word co-occurrence information; (2) latent space regularizer[6], which considers methods that explore latent semantic spaces, such as LDA, LSI and NMF; (3) word embedding regularizer, based on clustering of word vectors; and (4) graph-of-words regularizer, which discovers clusters in word graphs using community detection algorithms.

According to the aforementioned taxonomy, VGTM is essentially a graph-of-words regularizer, as it explores the community structure in the word graph for the task of topic modeling. Nevertheless, the graph-based text representation model used, Vec2Graph, is actually a graph of word embedding similarities, and the community detection method is the NMF algorithm. Therefore, VGTM also inherits characteristics of word embedding and latent space regularizers.

Algorithm 2 VGTM.

Require: $\mathcal{G}_C, k, V, C$
1: $\mathbf{G} \leftarrow \text{AdjacencyMatrix}(\mathcal{G}_C)$
2: $\mathbf{G} \leftarrow \text{ReplaceZeros}(\mathbf{G})$
3: $(\_, \mathbf{A}) \leftarrow \text{Find Communities}(\mathbf{G})$ ▷ Symetric non-negative matrix factorization
4: **for** $w \in V$ **do**
5: $\quad ||\mathbf{A}_w|| \leftarrow [\sum_t |\mathbf{A}_{tw}|]$ ▷ L1-norm of $\mathbf{A}_w$ (column)
6: $\quad$ **for** $t \in \{1, \cdots, k\}$ **do**
7: $\quad\quad P(t|w) \leftarrow ||\mathbf{A}_w||_t$ ▷ Probability of topic t for word w
8: **for** $t \in \{1, \cdots, k\}$ **do**
9: $\quad ||\mathbf{A}_t|| \leftarrow [\sum_w |\mathbf{A}_{tw}|]$ ▷ L1-norm of $\mathbf{A}_t$ (row)
10: $\quad$ **for** $w \in V$ **do**
11: $\quad\quad P(w|t) \leftarrow ||\mathbf{A}_t||_w$ ▷ Probability of word w for topic t
12: $\quad P(t) \leftarrow \frac{\sum_w P(t|w)}{\sum_j \sum_w P(j|w)}$ ▷ (3)
13: $\quad$ **for** $d \in C$ **do** ▷ Document d in corpus C
14: $\quad\quad P(t|d) = \frac{\sum_{w \in d} P(t|w)}{\sum_j \sum_{w \in d} P(j|w)}$ ▷ (4)

4.1 From graph communities to topics

As already explained, VGTM provides a probabilistic interpretation of topics. For this reason, methods able of detecting overlapping communities are preferable, as they ensure all words have a degree of membership to all communities (or topics), and not only a binary indicator of membership.

Most methods for detecting overlapping communities produce communities with little overlapping [49]. One alternative to them are NMF-based methods for overlapping

[6]The original label in [47] for this type of regularizer is *semantic regularizer*, but we consider *latent space regularizer* a more appropriate nomenclature, as used in [48].

communities detection [50, 51]. In particular, VGTM uses the weighted version of the SNMF (symmetric non-negative matrix factorization) proposed by Wang et al [50], where edges are symmetric (i.e., undirected) and weighted by similarities among the correspondent word vectors. SNMF can be replaced by any other overlapping community detection method[7].

Let $\mathbf{G}$ be the symmetric $|V| \times |V|$ adjacency matrix of the undirected weighted graph $\mathcal{G}_C$ (corpus graph), where V is the corpus vocabulary. The matrix $\mathbf{G}$ is usually sparse, since most words in $\mathcal{G}_C$ have a limited number of neighbors due to the restriction imposed by the σ threshold. In order to prevent the frequent appearance of zero probabilities, all zeros in $\mathbf{G}$ are replaced by a very small number (in our implementation, 10^{-4}), as indicated in line 2 of Alg. 2.

The NMF of $\mathbf{G}$ with rank k (number of communities or topics) is the approximation $\mathbf{G} \approx \mathbf{XA}$, where $\mathbf{X}$ contains the basis vectors and $\mathbf{A}$ is the $k \times |V|$ community-word (or topic-word) affinity matrix. The matrix $\mathbf{A}$ provides the affinity between the $|V|$ words in the vocabulary with the k communities or topics. From these values, VGTM infers the following probabilities:

- The posterior probability of topics per word, $P(t|w)$;
- The posterior probability of words per topic, $P(w|t)$;
- The posterior probability of topics, $P(t)$;
- The posterior probability of topics per document, $P(t|d)$.

4.2 Probability of topics per word

The matrix A returned by NMF indicates the affinity between the $|V|$ words in the vocabulary and the k topics. VGTM first normalizes the columns in matrix $\mathbf{A}$ (words) using the L1-norm such that $\sum_t \mathbf{A}_{tw} = 1$, i.e., the affinity of topics to a word w sums 1. Therefore, each column of $\mathbf{A}$ can be interpreted as the probability of word w belonging to each topic.

Figure 1 shows an example of the probability of topics per word for the corpus graph $\mathcal{G}_C$ generated for the Sanders dataset. VGTM was applied to this graph to discover 5 topics[8]. As previously explained, nodes in $\mathcal{G}_C$ are represented by tiny pie charts with colored slices indicating the proportion of topics. It is possible to observe regions dominated by specific topics (topics 2, 3 and 4), as well as mixture of topics (topics 1 and 5). With the exception of topic 3 (Technology), all other topics in this example capture the notion of idiom (Spanish, Dutch and English).

4.3 Probability of words per topic

Next, to define the probability of words per topic, the rows of matrix $\mathbf{A}$ are also L1-normalized such that $\sum_w \mathbf{A}_{tw} = 1$, i.e. the affinity of all words to a topic t sums 1. Therefore, each row of $\mathbf{A}$ is interpreted as a probability distribution over V. Figure 2 shows the probability distribution of the top-10 words for the topics 1, 2 and 3 illustrated in Fig. 1 (topic colors remain the same).

4.4 Probability of topics

Topics appear at different proportions in the dataset. We infer the posterior probability of a topic t, $P(t)$, from $\mathbf{A}$ according to (3). For each word w, we add up the proportion of t in W and normalize it by the sum of the probabilities of all other topics in the words. Figure 3 shows the proportion of topics for the Sanders corpus graph shown in Fig. 1 (topic colors remain the same).

$$P(t) = \frac{\sum_w P(t|w)}{\sum_j \sum_w P(j|w)} \tag{3}$$

4.5 Probability of topics per document

Finally, documents are represented as a probability distribution over topics. In VGTM, the probability of a topic t for a document d, $P(t|d)$, is inferred by using the rows in the topic-word affinity matrix $\mathbf{A}$ that match the words in d, and their correspondent topic column. $P(t|d)$ is given by the partial conditional probability shown in (4).

$$P(t|d) = \frac{\sum_{w \in d} P(t|w)}{\sum_j \sum_{w \in d} P(j|w)} \tag{4}$$

4.6 Time and space complexity analysis

Vec2Graph creates a corpus graph with $|V|$ vertices, where V is the corpus vocabulary, demanding $|V|(|V| - 1)/2$ word vector cosine distance operations, that is, $\mathcal{O}(|V|^2)$. VGTM internally runs NMF with rank k (number of topics) for overlapping community detection over the $|V| \times |V|$ matrix $\mathcal{G}_C$. We use an implementation based on Févotte and Idier [52], which is $\mathcal{O}(k \times |V|^2)$. The posterior probabilities are calculated over the $k \times |V|$ matrix $\mathbf{A}$. $P(w|t)$ and $P(t)$ are both obtained in $\mathcal{O}(k)$, while $P(t|w)$ in $\mathcal{O}(|V|)$. The probability of topics per document, $P(t|d)$, depends on the number of words per document – $|V|$ in the worst case – and k, resulting in a complexity of $\mathcal{O}(|V| + k)$. The overall time complexity of VGTM is $O(|V|^2 + k|V|^2 + 3|k| + 2|V|) = \mathcal{O}(k|V|^2)$.

[7] It is important to note that the use of NMF as an overlapping community detector is essentially different from the traditional use of NMF as a topic discovery method. In our case, we have a word-word adjacency matrix derived from $\mathcal{G}_C$, while in the traditional case we usually have a document-term TF-IDF matrix.

[8] An interactive version of this graph with topic information is available at: https://homepages.dcc.ufmg.br/~marcelo.pita/vgtm/sanders.html (2021/09/15)

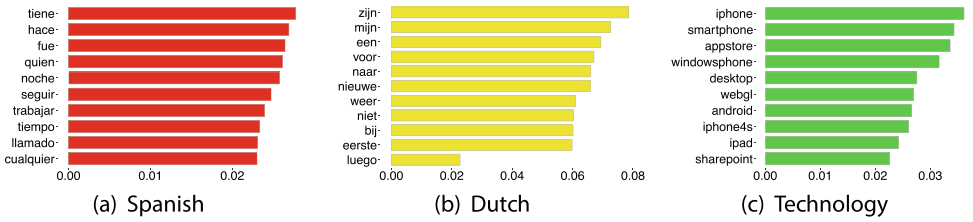

Fig. 2 Probability distribution of top-10 words per topic ("Spanish", "Dutch" and "Technology") for the Sanders dataset inferred by VGTM when using 5 topics

The space complexity of VGTM is dominated by the Vec2Graph representation, which is $\mathcal{O}(|V|^2)$, since it stores – in the worst case – the similarities between every pair of words.

5 Experimental analysis

We evaluated the topics generated by VGTM when applied to four short text benchmark datasets, as well as two real-world industry short text datasets. We start the experimental analysis by assessing the impact of the σ parameter (Vec2Graph similarity threshold) on the results. We then compared the best results found with consolidated topic models and state-of-the-art topic models for short text. Next, structural patterns of the corpus graph were analyzed to help understanding which factors actually correlate with the coherence of topics in VGTM.

5.1 Datasets

Four short text benchmark corpora and two real industry short text datasets were used in the experiments[9]:

1. 20 Newsgroups (20nshort): benchmark corpus partitioned across 20 different public email newsgroups, considering only emails with at most 20 words [16].
2. Tweets Sanders (Sanders): benchmark dataset with tweets related to four IT companies, namely, Google, Apple, Microsoft and Twitter [53].
3. Web snippets (Snippets): benchmark collection of web search snippets, which are summaries of queries generated by web search engines, related to 8 domains [54].
4. Tag My News (TMN): benchmark corpus of titles of RSS news items grouped into 7 categories [55].
5. Courses' Objectives (Courses): a real-world collection of Portuguese texts describing the objectives of 1,706 courses offered by SERPRO (Serviço Federal de Processamento de Dados), a Brazilian Federal Government company. The courses subjects are related to 25 categories of business subjects. Texts were trimmed to have up to 51 words. This value was defined after we considered as short text the minimum amount of text needed to express a single thought of the writer, which coincides with the definition of a paragraph [56], with varies from 3 to 5 sentences. Based on this information, we performed an analysis of the size of sentences in documents in the English and Portuguese Wikipedia dumps, where each sentence has an average of 17 and 16 words, respectively. From there, we defined as short text a document with at most 3 sentences of 17 words each, in a total of at most 50 words.
6. Customer Service Messages (CSM): a set of real-world messages received by a customer service at a Brazilian IT company. The dataset is anonymized and has 17,438 texts obtained by phone call transcriptions or directly through e-mails and Web forms. Messages are related to a business domain and are mapped to 13 categories of technical service. Texts were trimmed to have up to 50 words.

Datasets were preprocessed with the following standard steps: texts were converted to lower case, and non-alphabetic characters, stop words (i.e., words with low semantic value, including prepositions, articles and some usual verbs), and words with less than 3 characters were removed. Non-alphabetic characters include numbers, punctuation and special symbols. Numbers are usually related to quantities,

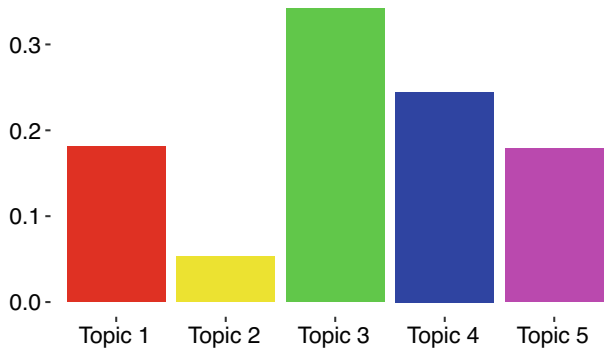

Fig. 3 Posterior probability distribution of topics for VGTM applied to the Sanders dataset

[9]Benchmark datasets available at: https://github.com/marcelopita/datasets/ (2021/01/01)

can vary a lot and have their semantics distributed among many symbols. Because of that, word vector are not, in general, good at representing numbers.

Table 2 shows basic statistics for the datasets. Note that all of them have, on average, very few words per document (column w/doc) and unique words per document (column unique w/doc). The average number of unique words per document ranges from 4.9 (TMN) to 26.6 (CSM). TMN is the largest dataset considering the number of documents. In terms of vocabulary size, CSM is the largest, while 20nshort is the smallest both in terms of number of documents and vocabulary size. All datasets have class labels, and the Courses and CSM datasets were manually labeled by experts.

5.2 Topic coherence evaluation metrics

Topic evaluation is not a simple task and, in an ideal scenario, a human-expert would be responsible for that. There is lot of discussion on topic evaluation in the literature [57, 58], and a few objective metrics are usually used by the community, including variations of PMI (pointwise mutual information), such as NPMI (normalized PMI) and UCI, UMass, C_V, and C_P. Among these metrics, [57] showed that NPMI and C_P have a high correlation with human-evaluation. These metrics are defined as follows.

NPMI Normalized Pointwise Mutual Informantion (NPMI) [59] is the normalized version of the PMI metric [60], which verifies the likelihood of co-occurrence of the most probable words in a topic relative to the co-occurrence of words for a 10-word sliding window on a large external dataset[10].

Let W_{10}^t be the 10 most representative words for topic t. The NPMI score for t is calculated according to (5) and (6):[11]

$$\texttt{NPMI-Score}(t; W_{10}^t) = mean\{\texttt{NPMI}(w_i, w_j) \mid w_i, w_j \in W_{10}^t, w_i \neq w_j\} \quad (5)$$

$$\texttt{NPMI}(w_i, w_j) = \frac{\ln \frac{p(w_i, w_j)}{p(w_i)p(w_j)}}{-\ln p(w_i, w_j)} \quad (6)$$

C_P This coherence metric calculates the likelihood of the most probable words of each topic co-occur in a large external dataset, taking into account a sliding window (in our case, 10-word) and the order that words occur [57]. The C_P score for t is calculated according to (7):[12]

$$\texttt{C}_P\texttt{-Score}(t; W_{10}^t) = mean\{\texttt{C}_P(w_i, w_j) \mid w_i, w_j \in W_{10}^t, i > j\} \quad (7)$$

where $\texttt{C}_P(w_i, w_j)$ is the Fitelson's coherence, which verifies the occurrence of w_j in each subset of the remaining words in $W_{10}^t, i > j$. The metric showed a high correlation with human judgement on a 10-word sliding window [57].

As these metrics have a high computational cost, we use only NPMI to evaluate parameter tuning. For the final experiments and comparisons with baselines, we compare the values of the two metrics described in this section.

5.3 Experimental setup

As previously mentioned, the first experiment we made was to evaluate the impact of VGTM parameters. After that, we compare the results of the method with the best results with 7 other methods. Among them, two are widely used topic modeling methods, namely LDA[13] and BTM[14]. We also considered the state-of-the-art methods for short text DREx[15], GPU-DMM[16], SeaNMF[17], and CluWords[18]. WNTM was also included as a baseline, since it is a graph-based topic model[19]. A recent short text topic modeling survey shows that BTM is still a very effective and robust method when compared to more recent contributions, such as SATM and PTM [13], which is why we did not consider these latest models.

For all these methods, the number of topics k assumed the values 20, 50 and 100, similar to values used in original papers [7, 11]. Regarding LDA parameters, the values of α and β (Dirichlet prior distributions) were optimized using the fixed point iteration method proposed by Minka [62]. The algorithm ran for 2,000 iterations, a large enough number to observe convergence. BTM also executed for 2,000 iterations, with $\beta = 0.005$ and $\alpha = 50/k$, as indicated in [7].

DREx, GPU-DMM, CluWords and VGTM use pre-trained word vectors. For English datasets (20nshort,

[10] In this case, a sample of 15 million documents from the WMT11 news corpus, available at http://www.statmt.org/wmt11/training-monolingual.tgz (2021/09/10)

[11] The NPMI score was calculated using the Palmetto tool [57]. Code available at: https://github.com/dice-group/Palmetto (2021/09/10).

[12] Code available at: https://github.com/dice-group/Palmetto (2021/09/10).

[13] Code available at: https://github.com/gabrielmip/LDAOpt (2021/01/01).

[14] Code available at: https://github.com/xiaohuiyan/BTM (2021/01/01).

[15] Code available at: https://github.com/marcelopita/drex_published (2021/01/01).

[16] GPU-DMM implementation of the STTM tool [13, 61]. Code available at: https://github.com/qiang2100/STTM (2021/01/01).

[17] Code available at: https://github.com/tshi04/SeaNMF (2021/01/01).

[18] Code available at: https://github.com/feliperviegas/cluwords (2021/01/01).

[19] Implemented according to the original paper [10].

Table 2 Statistics for the short text datasets. Columns indicate (left to right): dataset identifier; number of documents; vocabulary size (number of different terms); number of classes or categories; average number of words per document; and average number of unique (distinct) words per document

Dataset	#docs	Vocab. size	#classes	w/doc	unique w/doc
20Nshort	1723	964	20	8.2 (±3.5)	7.1 (±2.9)
Sanders	3770	1311	4	6.1 (±2.7)	5.8 (±2.5)
Snippets	12117	4677	8	14.3 (±4.4)	10.3 (±3.1)
TMN	30376	6314	7	4.9 (±1.5)	4.9 (±1.5)
Courses	1706	4791	25	17.3 (±11.2)	16.1 (±9.5)
CSM	17438	19679	13	30.1 (±15.3)	26.6 (±12.5)

Sanders, Snippets and TMN), word vectors were obtained from a English Wikipedia dump dated from 2015/06/02[20], using the Skip-Gram model with 1,000 dimensions, context window of size 10, negative sampling (NS) and initial learning rate (LR) of 0.025, as in [11]. Pre-trained Portuguese word vectors, used for the datasets Courses and CSM, were obtained from the STIL Corpora 2017 [63][21], using the fastText [64] Skip-Gram model with 1,000 dimensions. The reason of using a large number of dimensions for word vectors is because the more dimensions the vectors have, the greater the amount of semantics they carry. In other words, all models are using high quality word embeddings.

DREx enriches short texts using correlated words to produce longer pseudo-documents. However, it is not a topic modeling algorithm, so it must be used in combination with a general purpose topic model for long text. We used it with LDA, producing LDA-DREx. DREx was ran with $M = 60$ (target expanded document size), as suggested in [11].

The GPU-DMM model used the default parameters proposed in [17], with $\alpha = 0.1$ and $\beta = 0.01$. The algorithm ran for 2,000 iterations, just as in the other probabilistic graphic topic models analyzed. SeaNMF also used the default parameters proposed in [18], with $\alpha = 1.0$ and $\beta = 0.0$, i.e., disabling the sparsity constraint.

CluWords used the parameter $\alpha = 0.4$, as indicated in [19], meaning that only word cosine similarities above this threshold were considered by the method. WNTM used a sliding context window of size 10 to create the co-occurrence graph of words, as suggested by the authors [10]. A summary of the main parameter settings for each topic model is shown in Table 3.

Experiments were repeated 25 times and results were statistically validated with the non-parametric Wilcoxon signed-rank test with 0.05 of significance level over means.

[20]Data extracted from a XML file containing 8,102,107 articles and 2,120,659 words.

[21]Data extracted from 17 Brazilian and European Portuguese corpora in a total of 1,395,926,282 words. Available at: http://www.nilc.icmc.usp.br/embeddings (2021/01/01).

5.4 Tuning the similarity threshold

We started by assessing the impact of the parameter σ (similarity threshold) on the quality of the topics produced by VGTM. It is important to notice that σ has a direct influence on the connectivity patterns of the Vec2Graph corpus graph $\mathcal{G}_C$. Since VGTM uses $\mathcal{G}_C$ as the fundamental data structure for topics inference, different patterns can emerge by varying σ, which in our experiments ranges from 0.2 to 0.9 in 0.1 intervals.

Figure 4 shows the NPMI values (Y-axis) for the VGTM method applied to the datasets listed in Section 5.1 with different values of σ (X-axis) and k (number of topics). We observe that the best values of σ for the datasets 20nshort, Sanders, Snippets and TMN fall in the range of $[0.2, 0.4]$, with the overall best results for $\sigma = 0.4$. For the Courses and CSM datasets, our choice is $\sigma = 0.7$, as it presents overall higher NPMI values. The values of NPMI shown in the graphs are listed in the supplementary material.

In general, the values of NPMI for the 20nshort, Sanders, Snippets and TMN datasets increase monotonically with the values of σ until a global maximum is achieved, and the values start to decrease. However, the unstable behavior of Snippets, TMN, Courses and CSM datasets deserve a more detailed study, since it indicates that σ and k are not enough to explain the observed variations in NPMI.

For all datasets, in general, models with 20 topics present higher NPMI absolute values, which could be attributed to the proximity of this value to the number of class labels in the data: less than 25 for all datasets (see column #classes of Table 2).

5.5 Analysing the effect of Vec2Graph

Vec2Graph takes advantage of the semantic and relational properties of word embeddings to create a similarity graph of words. VGTM, in turn, exploits communities in this word graph to infer topics. This section analyses to which extent the word embeddings help VGTM to retain relevant semantic information. It replaces the VGTM Vec2Graph

Table 3 Main parameter settings for the topic models

	k	α	β	#Iter	M	Window
LDA	20, 50, 100	optimal	optimal	2,000	–	–
BTM		$50/k$	0.005	2,000	–	
DREx		optimal	optimal	2,000	60	–
GPU-DMM		0.1	0.001	–	–	–
CluWords		0.4	–	–	–	–
VGTM		–	–	–	–	–
SeaNMF		1.0	0.0	–	–	–
WNTM		–	–	–	–	10

structures, i.e. word similarity graphs, by a simple words co-occurrence graph.

In the words co-occurrence graph, nodes are words and edges are weighted by co-occurrence frequency. Co-occurrence frequencies are normalized in the interval [0, 1]. Thresholds of minimum word co-occurrence are explored in the interval $[0.2, 0.9]$ at 0.1 steps. Inferred topics are evaluated using the NPMI metric. The mean values of all variations of σ are listed in the supplementary material. We compare the best results of this naive approach, namely, $\sigma = 0.3$ and $\sigma = 0.9$, with the best results of VGTM, $\sigma = 0.4$ and $\sigma = 0.7$, as shown in Table 4.

The positive results for VGTM shown in Table 4 indicate that by incorporating word embedding semantics into the word graph (i.e. the Vec2Graph algorithm) we provide VGTM with information that is discriminative for topics, in comparison with an approach based on simple word co-occurrence. At first glance, incorporating the co-occurrence

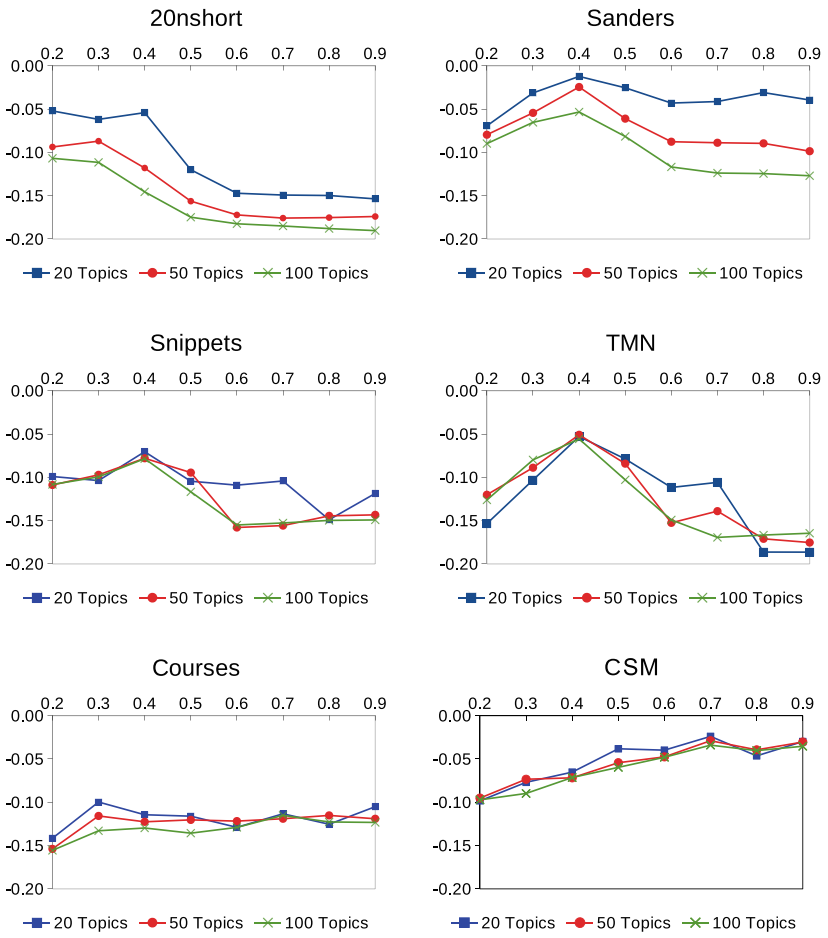

Fig. 4 Plots of NPMI values (Y-axis) for the VGTM method with different values of σ (X-axis), considering 20, 50 and 100 topics

Table 4 Comparison of NPMI results of VGTM with Vec2Graph ($\sigma = 0.4$ and $\sigma = 0.7$) versus a words co-occurrence graph ($\sigma = 0.3$ and $\sigma = 0.9$) for short text datasets

	20 topics	50 topics	100 topics	20 topics	50 topics	100 topics
	20nshort			*Sanders*		
Naive-0.3	-0.227	-0.243	-0.244	-0.140	-0.133	-0.152
Naive-0.9	-0.223	-0.238	-0.243	-0.143	-0.138	-0.155
VGTM-0.4	**-0.056**	**-0.118**	**-0.146**	**-0.012**	**-0.025**	**-0.053**
VGTM-0.7	-0.150	-0.175	-0.185	-0.040	-0.089	-0.124
	Snippets			*TMN*		
Naive-0.3	-0.164	-0.176	-0.193	-0.207	-0.208	-0.224
Naive-0.9	-0.166	-0.174	-0.198	-0.208	-0.215	-0.222
VGTM-0.4	**-0.072**	**-0.078**	**-0.079**	**-0.052**	**-0.051**	**-0.055**
VGTM-0.7	-0.103	-0.155	-0.153	-0.108	-0.139	-0.169
	Courses			*CSM*		
Naive-0.3	-0.237	-0.204	-0.184	-0.056	-0.045	-0.063
Naive-0.9	-0.232	-0.204	-0.182	-0.030	-0.052	-0.072
VGTM-0.4	**-0.114**	**-0.123**	**-0.130**	-0.065	-0.072	-0.071
VGTM-0.7	**-0.113**	**-0.119**	**-0.115**	**-0.024**	**-0.029**	**-0.034**

Bold values indicate the best results in the column for a dataset and K

between words in the model seems to be a good choice given the problem of scarce context in short texts for topic modeling. However, results show that the NPMI of inferred topics benefit more from a sophisticated language model – the word embeddings – than the simple word co-occurrence model.

5.6 Comparison with baselines

This section analyses the coherence of topics generated by VGTM, with $\sigma = 0.4$ and $\sigma = 0.7$, when compared 7 baselines, namely LDA, BTM, WNTM, LDA-DREx, GPU-DMM, SeaNMF and CluWords. The results of NPMI and C_P are shown in Tables 5 and 6, respectively.

Regarding the number of times a method was statistically the best method or presented no statistical difference to other methods, VGTM was the best in 10 out of 18 experiments, with a high agreement between both NPMI and C_P coherence metrics. VGTM was statistically the best in 8 out of 18 experiments (NPMI and C_P). CluWords and LDA-DREx also achieved good performance in NPMI, both being the best method or presenting no statistical difference to others in 6 and 7 experiments, respectively. For C_P, LDA-DREx and WNTM achieved the second best performance, both being the best method or presenting no statistical difference to others in 7 and 3 experiments, respectively. In general, LDA, BTM and GPU-DMM obtained the worse results.

For the real-world datasets Courses and CSM, VGTM with $\sigma = 0.7$ had consistently statistically significant better results than the other methods.

5.7 Analysing structural patterns of the corpus graphs

The properties of the graph generated using Vec2Graph can directly influence the topic modeling task (see Section 5.4). This section presents an extensive analysis of the structural properties of the graphs generated by Vec2Graph, using a set of complex network metrics (e.g., degree assortativity and transitivity). We analyzed the networks generated by Vec2Graph using all values of σ, for all datasets and number of topics. In total, 144 networks were considered (8 σ values, for 6 datasets and 3 variations of number of topics). After calculating a set of complex network metrics, the statistical correlation between network characteristics and the performance of the topic modeling algorithm (measured by NPMI) was measured. It is important to note that this correlation study was done after the networks were already formed, and not with the goal of optimizing the networks to improve topic modeling performance.

This analysis found interesting relationships between the network metrics of degree assortativity and transitivity and NPMI values. Figure 5 presents the values for the Pearson correlations, where the color indicates if the correlation is negative (closer to red) or positive (closer to blue). The ellipses represent the format of the plot (scatterplot) between every pair of variables. All correlations shown are statistically significant, with a significance level (*p-value*) under 0.01.

Degree Assortativity (r) measures the preference of nodes that have similar degrees to attach to each other [65], and its values fall within $[-1, 1]$. A high value ($r \approx 1$)

Table 5 Comparison of NPMI results of VGTM ($\sigma = 0.4$ and $\sigma = 0.7$) and baselines for short text datasets

	20 topics	50 topics	100 topics	20 topics	50 topics	100 topics
	20nshort			*Sanders*		
LDA	-0.188	-0.189	-0.203	-0.085	-0.113	-0.120
BTM	-0.211	-0.208	-0.205	-0.088	-0.098	-0.114
WNTM	-0.196	-0.202	-0.204	-0.093	-0.103	-0.118
LDA-DREx	**-0.045**	**-0.073**	**-0.099**	**-0.024**	-0.047	-0.063
GPU-DMM	-0.216	-0.203	-0.204	-0.077	-0.091	-0.101
SeaNMF	-0.21	-0.227	-0.228	-0.122	-0.145	-0.143
CluWords	-0.137	-0.162	-0.184	**-0.014**	-0.052	-0.086
VGTM-0.4	**-0.056**	-0.118	-0.146	**-0.012**	**-0.025**	**-0.053**
VGTM-0.7	-0.150	-0.175	-0.185	-0.040	-0.089	-0.124
	Snippets			*TMN*		
LDA	-0.076	-0.092	-0.107	-0.061	-0.058	-0.079
BTM	-0.055	-0.083	-0.087	-0.048	-0.054	-0.064
WNTM	-0.047	-0.061	-0.077	-0.039	-0.042	-0.057
LDA-DREx	**-0.023**	**-0.037**	**-0.051**	-0.061	-0.046	-0.053
GPU-DMM	-0.077	-0.09	-0.11	-0.051	-0.063	-0.073
SeaNMF	**-0.006**	-0.057	-0.087	-0.034	-0.075	-0.111
CluWords	-0.045	**-0.042**	**-0.058**	**0.006**	**0.002**	**-0.037**
VGTM-0.4	-0.072	-0.078	-0.079	-0.052	-0.051	-0.055
VGTM-0.7	-0.103	-0.155	-0.153	-0.108	-0.139	-0.169
	Courses			*CSM*		
LDA	-0.258	-0.243	-0.236	-0.164	-0.161	-0.159
BTM	-0.248	-0.242	-0.241	-0.151	-0.151	-0.145
WNTM	-0.273	-0.257	-0.246	-0.146	-0.151	-0.159
LDA-DREx	-0.200	-0.185	-0.181	-0.150	-0.146	-0.144
GPU-DMM	-0.233	-0.238	-0.238	-0.151	-0.148	-0.136
SeaNMF	-0.183	-0.175	-0.189	-0.099	-0.080	-0.077
CluWords	-0.159	-0.167	-0.177	-0.252	-0.264	-0.257
VGTM-0.4	**-0.114**	**-0.123**	**-0.130**	-0.065	-0.072	-0.071
VGTM-0.7	**-0.113**	**-0.119**	**-0.115**	**-0.024**	**-0.029**	**-0.034**

Bold values indicate the best results in the column for a dataset and k

indicates that nodes that have a high degree are very likely to be connected to each other, meaning that the network is composed of a core of interconnected high degree nodes, and a periphery of lower degree nodes. A low value ($r \approx$ -1, also viewed as a high value for disassortativity), in turn, indicates that high degree nodes are more likely to be connected with low degree nodes, which means that star-like structures are more likely to appear.

In the context of this work, a high value of degree assortativity means that words in the corpus graph form tight-knit communities, inside which words are highly similar between themselves (therefore connected). Our analysis showed that *low σ (similarity threshold) values produce networks with high degree assortativity*, which makes sense because when σ is low, the networks tend to be denser (closer to a complete graph). We also observed that *high assortativity values correlate to high NPMI values:* $r = 0.41$, *p-value* $= 5.83 \times 10^{-5}$. We hypothesize that this correlation happens due to the fact that a closely knit community of words in the graph semantically corresponds to words associated to the same topic.

Graph transitivity is defined as the fraction of closed triads over all possible triads in a graph [66]. A *triad* in a graph $G = (V, E)$ includes three vertices $\{u, v, w\} \in V$ connected pairwise, *i.e.* $\{(u, v), (v, w)\} \in E$. A triad is considered closed (triangle) if there exists an edge between the vertices on its ends $(u, w) \in E$. The intuition for interpreting transitivity comes from social networks studies, when it is interesting to know the fraction of friendships that began due to a close acquaintance (the idea is that "friends of friends are also friends").

In our corpus graph, transitivity translates into the fraction of similar words w_1 and w_2 that have another similar word w_3 as a neighbor. It gives a general idea of

Table 6 Comparison of C_P results of VGTM ($\sigma = 0.4$ and $\sigma = 0.7$) and baselines for short text datasets

	20 topics	50 topics	100 topics	20 topics	50 topics	100 topics
	20nshort			*Sanders*		
LDA	-0.735	-0.721	-0.761	-0.492	-0.551	-0.565
BTM	-0.770	-0.776	-0.762	-0.515	-0.519	-0.564
WNTM	-0.753	-0.766	-0.765	-0.502	-0.523	-0.563
LDA-DREx	**-0.524**	**-0.587**	**-0.619**	-0.311	-0.385	**-0.417**
GPU-DMM	-0.761	-0.759	-0.744	-0.497	-0.538	-0.536
SeaNMF	-0.793	-0.810	-0.796	-0.476	-0.570	-0.592
CluWords	-0.630	-0.672	-0.696	-0.314	-0.396	-0.447
VGTM-0.4	**-0.528**	-0.633	-0.695	**-0.248**	**-0.305**	**-0.413**
VGTM-0.7	-0.688	-0.734	-0.744	-0.393	-0.496	-0.580
	Snippets			*TMN*		
LDA	-0.415	-0.504	-0.526	-0.508	-0.506	-0.552
BTM	**-0.337**	-0.474	-0.508	-0.474	-0.502	-0.526
WNTM	-0.354	-0.463	-0.498	**-0.449**	**-0.482**	**-0.508**
LDA-DREx	**-0.349**	**-0.422**	**-0.471**	-0.550	-0.515	-0.547
GPU-DMM	-0.406	-0.480	-0.531	-0.486	-0.515	-0.534
SeaNMF	**-0.324**	-0.536	-0.603	-0.482	-0.598	-0.672
CluWords	-0.443	-0.481	-0.534	-0.484	**-0.483**	-0.562
VGTM-0.4	-0.497	-0.540	-0.569	-0.602	-0.606	-0.629
VGTM-0.7	-0.574	-0.664	-0.685	-0.646	-0.712	-0.751
	Courses			*CSM*		
LDA	-0.847	-0.793	-0.775	-0.638	-0.641	-0.616
BTM	-0.791	-0.801	-0.800	-0.617	-0.600	-0.585
WNTM	-0.860	-0.822	-0.794	-0.604	-0.596	-0.607
LDA-DREx	-0.702	-0.613	-0.597	-0.608	-0.533	-0.500
GPU-DMM	-0.818	-0.797	-0.800	-0.643	-0.624	-0.576
SeaNMF	-0.691	-0.678	-0.727	-0.453	-0.404	-0.378
CluWords	-0.573	-0.587	-0.611	-0.781	-0.839	-0.798
VGTM-0.4	**-0.434**	**-0.470**	**-0.469**	-0.209	-0.239	-0.247
VGTM-0.7	**-0.447**	**-0.467**	**-0.469**	**-0.060**	**-0.111**	**-0.149**

Bold values indicate the best results in the column for a dataset and k

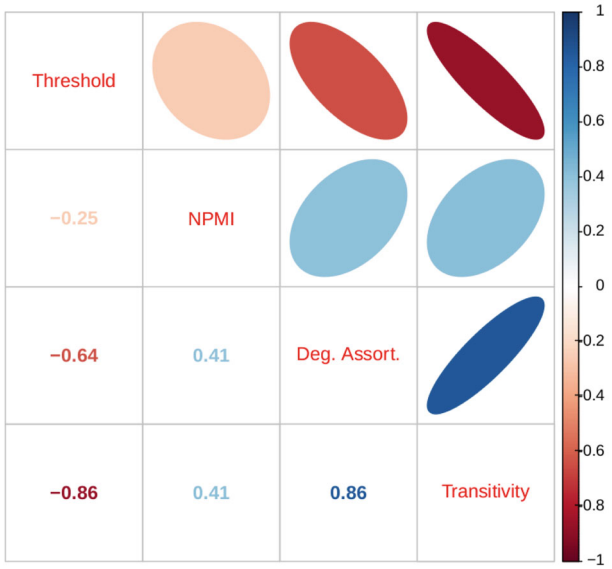

Fig. 5 Network metrics which achieved significant correlation values with NPMI

how dense the network is, and how tightly-knit are the communities of similar words.

Our analysis showed that *the similarity threshold is strongly negatively correlated to transitivity* ($r = -0.86$, with *p-value* $< 2 \times 10^{-16}$), which is plausible because the higher σ is, the more strict the connectivity constraints are (two words, or nodes, will only share an edge if their cosine similarity is very high). As σ increases, the graph tends to have fewer edges, and is more prone to being disconnected. However, *transitivity correlates positively with NPMI* ($r = 0.41$, with *p-value* $= 0.0046$), which makes sense due to the fact that a high transitivity value for the graph indicates that words connect to other similar words very easily, therefore forming groups more easily. In these highly connected networks, it is easier for VGTM to find communities of words that represent a coherent topic, resulting in a high NPMI value. Both correlations are statistically significant, with *p-values* under 0.01.

6 Conclusions and future work

This paper proposed a new representation for document corpora named Vec2Graph, which captures patterns of semantic similarity between words in a corpus using the cosine similarity of word vectors. This new representation was exploited to create a graph-based probabilistic topic model for short text, named Vec2Graph Topic Model (VGTM). VGTM combines the main characteristics of previous methods proposed for short text topic modeling – such as using word embeddings and the adoption of aggregation strategies – but also introduces a new way to find topics by using the connectivity patterns of words using an overlapping community detection method. We argue that the good results obtained by the method are due to the combination of these characteristics, which were able to address the lack of context information available in short text with high quality relationships between words.

The results obtained by VGTM in terms of NPMI and C_p were compared to seven other methods for short text topic modeling, including the state-of-the-art method GPU-DMM and the word graph-based topic model WNTM. VGTM obtained the best overall results among the compared methods. They were particularly expressive for the real-world application datasets. Although the method also provides vector representation for documents as posterior probabilities over topics, in this work we focused on the evaluation of topics. The evaluation of topical document representation remains for the future (e.g. through document classification tasks).

It is important to say that Vec2Graph is general enough to be also used in other tasks, such as text summarization and query expansion. Also, a deeper investigation on how document length and the number of documents in the corpus relate to VGTM performance (NPMI and C_P) can provide insights into how the method would behave for longer text topic modeling.

Finally, as any other method, VGTM has limitations. The graph representation is highly dependent on the quality of word embeddings, which in turn depends on a set of parameters, including the dimensions of word representations. The use of the similarity threshold (σ) is also not ideal, as it has a strong impact on the structure of the graph. In this direction, in the future VGTM could be combined with an optimization mechanism, which can help finding corpus graphs that optimize structural metrics that are highly correlated with topic coherence, such as transitivity and degree assortativity. This could replace the use of σ.

Acknowledgements The authors would like to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) and Serviço Federal de Processamento de Dados (SERPRO), for their financial support.

Funding Gisele L. Pappa was supported by FAPEMIG (grant no. CEX-PPM-00098-17), MPMG (through the project Analytical Capabilities), and CNPq (grant no. 310833/2019-1). Marcelo Pita was supported by SERPRO (through the Graduate Incentive Program). Matheus Nunes was supported by CNPq (studentship grant).

Availability of Data and Material Links to the datasets appear as footnotes in the main body of the paper. There are two datasets (Courses and CSM) that can be made available under request, as they may include sensitive information.

Code Availability The Vec2Graph code is available at https://github.com/marcelopita/vec2graph_paper. The VGTM code is available at https://github.com/marcelopita/vgtm.

Declarations

Conflicts of Interest None declared.

References

1. Boyd-Graber JL, Hu Y, Mimno D et al (2017) Applications of topic models. Now Publishers Incorporated, 11
2. Rosso P, Errecalde M, Pinto D (2013) Analysis of short texts on the web: introduction to special issue. Lang Resour Eval 47(1):123–126
3. Zhang H, Zhong G (2016) Improving short text classification by learning vector representations of both words and hidden topics. Knowl-Based Syst 102:76–86
4. Blei D, Ng A, Jordan M (2003) Latent Dirichlet allocation. JMLR 3:993–1022
5. Xu W, Liu X, Gong Y (2003) Document clustering based on non-negative matrix factorization. In: SIGIR, ACM, pp 267–273
6. Tang J, Meng Z, Nguyen X, Mei Q, Zhang M (2014) Understanding the limiting factors of topic modeling via posterior contraction analysis. In: ICML, pp 190–198

7. Yan X, Guo J, Lan Y, Cheng X (2013) A biterm topic model for short texts. In: WWW, ACM, pp 1445–1456
8. Yin J, Wang J (2014) A dirichlet multinomial mixture model-based approach for short text clustering. In: KDD, ACM, pp 233–242
9. Zuo Y, Wu J, Zhang H, Lin H, Wang F, Xu K, Xiong H (2016) Topic modeling of short texts: A pseudo-document view. In: KDD, ACM, pp 2105–2114
10. Zuo Y, Zhao J, Xu K (2016) Word network topic model: a simple but general solution for short and imbalanced texts. Knowl Inf Syst 48(2):379–398
11. Bicalho P, Pita M, Pedrosa G, Lacerda A, Pappa GL (2017) A general framework to expand short text for topic modeling. Inf Sci 393:66–81
12. Nguyen HT, Duong PH, Cambria E (2019) Learning short-text semantic similarity with word embeddings and external knowledge sources. Knowl-Based Syst 182:104842
13. Qiang J, Qian Z, Li Y, Yuan Y, Wu X (2020) Short text topic modeling techniques, applications, and performance: A survey. TKDE, pp 1–1
14. Mikolov T, Corrado G, Chen K, Dean J (2013) Efficient Estimation of Word Representations in Vector Space. In: ICLR, pp 1–12
15. Pennington J, Socher R, Manning CD (2014) GloVe: Global Vectors for Word Representation. In: EMNLP, pp 1532–1543
16. Nguyen DQ, Billingsley R, Du L, Johnson M (2015) Improving topic models with latent feature word representations. TACL 3:299–313
17. Li C, Wang H, Zhang Z, Sun A, Ma Z (2016) Topic modeling for short texts with auxiliary word embeddings. In: SIGIR, ACM, pp 165–174
18. Shi T, Kang K, Choo J, Reddy CK (2018) Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In: WWW, pp 1105–1114
19. Viegas F, Canuto S, Gomes C, Luiz W, Rosa T, Ribas S, Rocha L, Gonçalves MA (2019) Cluwords: exploiting semantic word clustering representation for enhanced topic modeling. In: WSDM, pp 753–761
20. Hong L, Davison BD (2010) Empirical study of topic modeling in twitter. In: KDD Workshops, ACM, pp 80–88
21. Mikolov T, Chen K, Corrado G, Dean J (2013) Distributed Representations of Words and Phrases and their Compositionality. In: NeurIPS, pp 1–9
22. Xie J, Kelley S, Szymanski BK (2013) Overlapping community detection in networks: The state-of-the-art and comparative study. ACM Computer Surveys 45(4):43
23. Dieng AB, Ruiz FJR, Blei DM (2020) Topic modeling in embedding spaces. Transactions of the Association for Computational Linguistics 8:439–453
24. Srivastava A, Sutton C (2017) Autoencoding variational inference for topic models. In: ICLR, pp 1–12
25. Zhang H, Chen B, Cong Y, Guo D, Liu H, Zhou M (2020) Deep autoencoding topic model with scalable hybrid bayesian inference. IEEE TPAMI
26. Zhang H, Chen B, Guo D, Zhou M (2018) WHAI: Weibull hybrid autoencoding inference for deep topic modeling. In: ICLR
27. Gupta P, Chaudhary Y, Buettner F, Schütze H (2019) Document informed neural autoregressive topic models with distributional prior. In: AAAI, vol 33, pp 6505–6512
28. Quan X, Kit C, Ge Y, Pan SJ (2015) Short and sparse text topic modeling via self-aggregation. In: AAAI, pp 2270–2276
29. Li X, Li C, Chi J, Ouyang J (2018) Short text topic modeling by exploring original documents. Knowl Inf Syst 56(2):443–462
30. Mahmoud H (2008) Pólya urn models. Chapman and Hall/CRC
31. Das R, Zaheer M, Dyer C (2015) Gaussian lda for topic models with word embeddings. In: AACL-IJCNLP, pp 795–804
32. Shi B, Lam W, Jameel S, Schockaert S, Lai KP (2017) Jointly learning word embeddings and latent topics. In: SIGIR, pp 375–384
33. Li X, Zhang A, Li C, Guo L, Wang W, Ouyang J (2019) Relational biterm topic model: Short-text topic modeling using word embeddings. The Computer Journal 62(3):359–372
34. Tuan AP, Bach TX, Nguyen TH, Linh NV, Than K (2020) Bag of biterms modeling for short texts. Knowl. Inf. Syst. 62(10):4055–4090
35. Mehrotra R, Sanner S, Buntine W, Xie L (2013) Improving lda topic models for microblogs via tweet pooling and automatic labeling. In: SIGIR, pp 889–892
36. Qiang J, Chen P, Wang T, Wu X (2017) Topic modeling over short texts by incorporating word embeddings. In: PAKDD, Springer, pp 363–374
37. Xie P, Yang D, Xing E (2015) Incorporating word correlation knowledge into topic modeling. In: NAACL, pp 725–734
38. Gao W, Peng M, Wang H, Zhang Y, Xie Q, Tian G (2019) Incorporating word embeddings into topic modeling of short text. Knowl Inf Syst 61(2):1123–1145
39. Rashid J, Shah SMA, Irtaza A (2019) Fuzzy topic modeling approach for text mining over short text. Information Processing & Management 56(6):102060
40. Osman AH, Barukub OM (2020) Graph-based text representation and matching: A review of the state of the art and future challenges. IEEE Access 8:87562–87583
41. Rousseau F, Kiagias E, Vazirgiannis M (2015) Text categorization as a graph classification problem. In: AACL-IJCNLP, pp 1702–1712
42. Meladianos P, Tixier A, Nikolentzos I, Vazirgiannis M (2017) Real-time keyword extraction from conversations. In: EACL, pp 462–467
43. Blanco R, Lioma C (2012) Graph-based term weighting for information retrieval. Information retrieval 15(1):54–92
44. Rousseau F, Vazirgiannis M (2013) Graph-of-word and tw-idf: new approach to ad hoc ir. In: CIKM, pp 59–68
45. Malliaros FD, Vazirgiannis M (2017) Graph-based text representations: Boosting text mining, nlp and information retrieval with graphs. In: EMNLP
46. David E, Jon K (2010) Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, New York, NY, USA
47. Skianis K, Rousseau F, Vazirgiannis M (2016) Regularizing text categorization with clusters of words. In: EMNLP, pp 1827–1837
48. Yang L, Cao X, Jin D, Wang X, Meng D (2014) A unified semi-supervised community detection framework using latent space graph regularization. Transactions on Cybernetics 45(11):2585–2598
49. Amelio A, Pizzuti C (2014) Overlapping community discovery methods: a survey. In: Social Networks: Analysis and Case Studies. Springer, pp 105–125
50. Wang F, Li T, Wang X, Zhu S, Ding C (2011) Community discovery using nonnegative matrix factorization. DMKD 22(3):493–521
51. Zhang Y, Yeung D-Y (2012) Overlapping community detection via bounded nonnegative matrix tri-factorization. In: KDD, ACM, pp 606–614
52. Févotte C, Idier J (2011) Algorithms for nonnegative matrix factorization with the β-divergence. Neural computation 23(9):2421–2456
53. Sanders NJ (2011) Sanders-twitter sentiment corpus. Sanders Analytics LLC 242:1–4
54. Phan X-H, Nguyen L-M, Horiguchi S (2008) Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: WWW, ACM, pp 91–100
55. Vitale D, Ferragina P, Scaiella U (2012) Classification of short texts by deploying topical annotations. In: ECIR, Springer, pp 376–387

56. The Writing Center, University of North Carolina at Chapel Hill: Paragraphs (2019)
57. Röder M, Both A, Hinneburg A (2015) Exploring the space of topic coherence measures. In: WSDM, ACM, pp 399–408
58. Doogan C, Buntine W (2021) Topic model or topic twaddle? re-evaluating semantic interpretability measures. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 3824–3848
59. Bouma G (2009) Normalized (pointwise) mutual information in collocation extraction. GSCL, pp 31–40
60. Newman D, Noh Y, Talley E, Karimi S, Baldwin T (2010) Evaluating topic models for digital libraries. In: JCDL, pp 215–224
61. Qiang J, Li Y, Yuan Y, Liu W, Wu X (2018) STTM: A tool for short text topic modeling. CoRR abs/1808.02215
62. Minka T (2000) Estimating a dirichlet distribution. Technical report, MIT
63. Hartmann NS, Fonseca ER, Shulby CD, Treviso MV, Rodrigues JS, Aluísio SM (2017) Portuguese word embeddings: Evaluating on word analogies and natural language tasks. In: STIL. SBC, pp 122–131
64. Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. TACL 5:135–146
65. Newman MEJ (2003) Mixing patterns in networks. Phys Rev E 67(2):026126
66. Newman M (2018) Networks. Oxford university press

Marcelo Pita is head of the data science technology division at SERPRO, Brazil. He holds a PhD in Computer Science from the Universidade Federal de Minas Gerais (UFMG), Brazil. His current research activities and interests are related to machine learning methods for natural language processing. He has published works in the fields of short text topic modeling, agent-based simulation for social complex systems and e-Government.

Matheus Nunes is a Software Engineer at Google. He holds a Master's and a Bachelor's degree in Computer Science, both from the Universidade Federal de Minas Gerais (UFMG), in Brazil. His academic interests are centered around the interaction of graphs and machine learning. His master's thesis focused on the applicability of Automated Machine Learning (AutoML) methods to Graph Neural Networks (GNNs).

Gisele L. Pappa is an Associate Professor in the Computer Science Department at UFMG, Brazil. She has an extensive publication record in the intersection of the machine learning and evolutionary computation areas and has also been working with text data, embedded representations and topic modelling. Currently, she is interested in the applications of machine learning in both health data and fraud detection.