Knowledge-Based Short Text Categorization Using Entity and Category Embedding

Rima Türker[1,2](✉), Lei Zhang[1], Maria Koutraki[1,2,3], and Harald Sack[1,2]

[1] FIZ Karlsruhe – Leibniz Institute for Information Infrastructure,
Karlsruhe, Germany
{rima.tuerker,lei.zhang,maria.koutraki,harald.sack}@fiz-karlsruhe.de
[2] Karlsruhe Institute of Technology, Institute AIFB, Karlsruhe, Germany
{rima.tuerker,maria.koutraki,harald.sack}@kit.edu
[3] L3S Research Center, Leibniz University of Hannover, Hannover, Germany
koutraki@l3s.de

Abstract. Short text categorization is an important task due to the rapid growth of online available short texts in various domains such as web search snippets, etc. Most of the traditional methods suffer from sparsity and shortness of the text. Moreover, supervised learning methods require a significant amount of training data and manually labeling such data can be very time-consuming and costly. In this study, we propose a novel probabilistic model for Knowledge-Based Short Text Categorization (KBSTC), which does not require any labeled training data to classify a short text. This is achieved by leveraging entities and categories from large knowledge bases, which are further embedded into a common vector space, for which we propose a new entity and category embedding model. Given a short text, its category (e.g. *Business*, *Sports*, etc.) can then be derived based on the entities mentioned in the text by exploiting semantic similarity between entities and categories. To validate the effectiveness of the proposed method, we conducted experiments on two real-world datasets, i.e., AG News and Google Snippets. The experimental results show that our approach significantly outperforms the classification approaches which do not require any labeled data, while it comes close to the results of the supervised approaches.

Keywords: Short text classification · Dataless text classification · Network embeddings

1 Introduction

Short text categorization is gaining more and more attention due to the availability of a huge number of text data, which includes search snippets, short messages as well as text data generated in social forums [1,17,18]. Although, traditional text classification methods perform well on long text such as news article, yet, by considering short text, most of them suffer from issues such as data sparsity and insufficient text length, which is no longer than 200 characters [13]. In other words, simple text classification approaches based on bag of words (BOW) cannot properly

P. Hitzler et al. (Eds.): ESWC 2019, LNCS 11503, pp. 346–362, 2019.
https://doi.org/10.1007/978-3-030-21348-0_23

represent short text as the semantic similarity between single words is not taken into account [21]. Also, approaches that utilize word embeddings for classification perform better when dealing with longer text, where ambiguities can be resolved based on the provided context information within the given text. In the case of short text, where the available context is rather limited and each word obtains significant importance, such approaches often lead to inaccurate results.

Another characteristic of existing approaches is that they all require a significant amount of labeled training data and a sophisticated parameter tuning process [24]. Manual labeling of such data can be a rather time-consuming and costly task. Especially, if the text to be labeled is of a specific scientific or technical domain, crowd-sourcing based labeling approaches do not work successfully and only expensive domain experts are able to fulfill the manual labeling task. Alternatively, semi-supervised text classification approaches [8,22] have been proposed to reduce the labeling effort. Yet, due to the diversity of the documents in many applications, generating small training set for semi-supervised approaches still remains an expensive process [4].

To overcome the requirement for labeled data, a number of *dataless text classification* methods have been proposed [2,14]. These methods do not require any labeled data as a prerequisite. Instead, they rely on the semantic similarity between a given document and a set of predefined categories to determine which category the given document belongs to. More specifically, documents and categories are represented in a common semantic space based on the words contained in the documents and category labels, which allows to calculate a meaningful semantic similarity between documents and categories. The classification process depends on this semantic similarity. However, the most prominent and successful dataless classification approaches are designed for long documents.

Motivated by the already mentioned challenges, we propose a novel probabilistic model for Knowledge-Based Short Text Categorization (KBSTC), which does not require any labeled training data. It is able to capture the semantic relations between the entities represented in a short text and the predefined categories by embedding them into a common vector space using the proposed network embedding technique. Finally, the category of the given text can be derived based on the semantic similarity between entities present in the given text and the set of predefined categories. The similarity is computed based on the vector representation of entities and categories. Overall, the main contributions of the paper are as follows:

- a new paradigm for short text categorization, based on a knowledge base;
- a probabilistic model for short text categorization;
- a new method of entity and category embedding;
- an evaluation using standard datasets for short text categorization.

The rest of this paper is structured as follows: Sect. 2 discusses related work. In Sect. 3, the proposed approach for short text categorization is explained. Section 4 presents the joint entity and category embeddings used in this paper, while Sect. 5 describes the experimental setup for the evaluation as well as the applied baselines. It further illustrates and discusses the achieved results. Last, Sect. 6, concludes the paper with a discussion of open issues and future work.

2 Related Work

The aim of this work is to categorize (e.g. *Business*, *Sports*, etc.) a given short text by utilizing entity and category embeddings without requiring any labeled data for training. Thus, our work is mainly related to three prior studies: Short Text Classification, Dataless Text Classification as well as Entity and Category Embedding.

Short Text Classification. In order to overcome the data sparsity problem of short text, recent works [20,21] proposed deep learning based approaches for short text classification. The results of these approaches have been compared with traditional supervised classification methods, such as SVM, multinomial logistic regression, etc., where the authors showed that in most of the cases their approach achieved superior results. While performing well in practice, the aforementioned approaches are slow both in the training and in the test phase. In addition, their performance highly depends on the size of training data, its distribution, and the chosen hyper parameters. In difference, our approach does not require any training data nor any parameter tuning.

Dataless Text Classification. In order to address the problem of missing labeled data, [2] introduced a dataless text classification method by representing documents and category labels in a common semantic space. As source, the online encyclopedia, Wikipedia was utilized supported with Explicit Semantic Analysis (ESA) [3] to quantify semantic relatedness between the labels to be assigned and the documents. As a result, it was shown that ESA is able to achieve better classification results than the traditional BOW representations. Further, [14] proposed a dataless hierarchical text classification by dividing the dataless classification task into two steps. In the semantic similarity step, both labels and documents were represented in a common semantic space, which allows to calculate semantic relatedness between documents and labels. In the bootstrapping step, the approach made use of a machine learning based classification procedure with the aim of iteratively improving classification accuracy.

In contrast to these approaches, our proposed approach differs in two main aspects. First, all the mentioned studies were designed for the classification of documents of arbitrary length. However the main purpose of this work is to categorize short text documents without the necessity of labeled training data. Second, none of the mentioned approaches did make use of the entities present in a short text document. To represent a document, all the mentioned approaches consider the words contained in the document.

Entity and Category Embeddings. To generate entity and category embeddings, different embedding models can be employed. For instance, RDF2Vec [11] and DeepWalk [9] adopt a language modeling approach to learn the representation of vertices in a large network. Further, DeepWalk is designed for homogeneous networks, while RDF2Vec aims to deal with RDF graphs, however, it treats each type of vertices and edges equally. HCE [5], as the state-of-the-art entity and category embedding model, integrates the category hierarchy structure into the embedding space. This model has been applied to the dateless

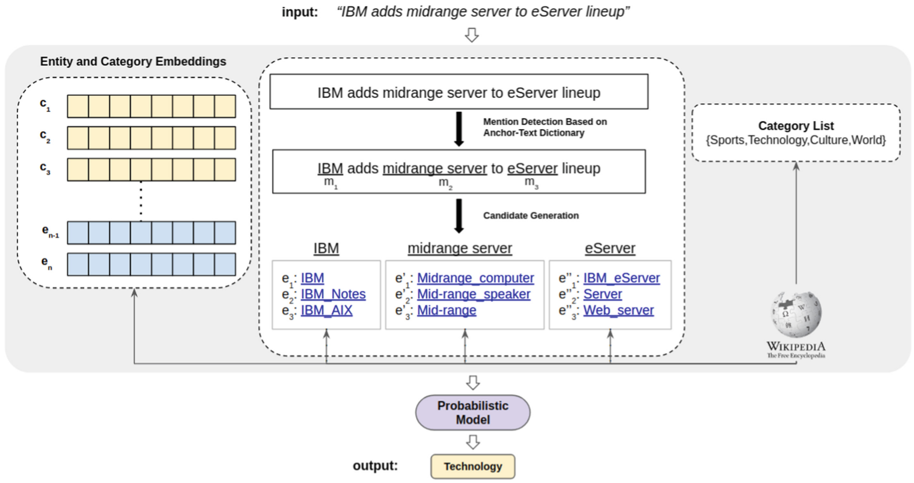

Fig. 1. The work flow of the proposed KBSTC approach (best viewed in color) (Color figure online)

classification task and it has outperformed the baselines. Recently, [15] proposed a *Predictive Text Embedding* (PTE) model, which uses labeled data and word co-occurrence information to build a heterogeneous text network, where multiple types of vertices exist, and then applies the proposed algorithm to learn the embedding of text. Inspired by PTE, our proposed entity and category embedding model firstly constructs a weighted network of entities and categories, and then jointly learns their embeddings from the network.

3 Knowledge-Based Short Text Categorization (KBSTC)

This section provides a formal definition of the Knowledge-Based Short Text Categorization (KBSTC) task, followed by the description of the proposed probabilistic approach for KBSTC.

Preliminaries. Given a knowledge base KB containing a set of entities $E = \{e_1, e_2, .., e_n\}$ and a set of hierarchically related categories $C = \{c_1, c_2, .., c_m\}$, we model KB as a graph $G_{KB} = (V, R)$ with $V = E \cup C$ as the set of vertices and $R = R_{EE} \cup R_{EC} \cup R_{CC}$ as the set of edges of the form (v_i, v_j) reflecting various relationships between the vertices v_i and v_j, where each edge in R_{EE} with $v_i, v_j \in E$ represents an entity-entity relation, each edge in R_{EC} with $v_i \in E$ and $v_j \in C$ represents an entity-category association, and each edge in R_{CC} with $v_i, v_j \in C$ reflects the category hierarchy.

In this work, we utilize Wikipedia as the knowledge base, where each article and each category page are considered as an entity in E and a category in C, respectively. In addition, each relationship (v_i, v_j) between the pair of vertices v_i and v_j are extracted from Wikipedia and the following rule applies:

– $(v_i, v_j) \in R_{EE}$ if and only if $v_i, v_j \in E$ and there is a link from the article v_i to the article v_j in Wikipedia,
– $(v_i, v_j) \in R_{EC}$ if and only if $v_i \in E, v_j \in C$ and the article v_i has the associated category v_j in Wikipedia, and
– $(v_i, v_j) \in R_{CC}$ if and only if $v_i, v_j \in C$ and v_i is subcategory of v_j in Wikipedia.

Definition (KBSTC task). Given an input short text t that contains a set of entities $E_t \subseteq E$ as well as a set of predefined categories $C' \subseteq C$ (from the underlying knowledge base KB), the output of the KBSTC task is the most relevant category $c_i \in C'$ for the given short text t, i.e., we compute the category function $f_{cat}(t) = c_i$, where $c_i \in C'$.

KBSTC Overview. The general workflow of KBSTC is shown in Fig. 1. In the first step, each entity mention present in a given short text t is detected. Next, for each mention, a set of candidate entities are generated based on a prefabricated Anchor-Text Dictionary, which contains all mentions and their corresponding Wikipedia entities. In order to detect entity mentions, first all n-grams from the input text are gathered and then the extracted n-grams matching surface forms of entities (based on the Anchor-Text dictionary) are selected as entity mentions. To construct the Anchor-Text Dictionary, all the anchor texts of hyperlinks in Wikipedia pointing to any Wikipedia articles are extracted, whereby the anchor texts serve as mentions and the links refer to the corresponding entities. Given the short text t as *"IBM adds midrange server to eServer lineup"*, the detected mentions are *"IBM"*, *"midrange server"* and *"eServer"*. Likewise the predefined categories, $C' = \{Sports, Technology, Culture, World\}$, are mapped to Wikipedia categories. Finally, applying the proposed probabilistic model (see Sect. 3) by utilizing the entity and category embeddings that have been precomputed from Wikipedia (see Sect. 4), the output of the KBSTC task is the semantically most relevant category for the entities present in t. Thereby, in the given example the category *Technology* should be determined.
Note that in this work we have utilized Wikipedia as a KB. However, KBSTC is applicable to any arbitrary domain as long as there exists a KB providing domain-specific entities and categories.

3.1 Probabilistic Approach

The KBSTC task is formalized as estimating the probability of $P(c|t)$ of each predefined category c and an input short text t. The result of this probability estimation can be considered as a score for each category. Therefore, the most relevant category c for a given text t should maximize the probability $P(c|t)$. Based on Bayes' theorem, the probability $P(c|t)$ can be rewritten as follows:

$$P(c|t) = \frac{P(c,t)}{P(t)} \propto P(c,t)\,, \tag{1}$$

where the denominator $P(t)$ can be ignored as it has no impact on the ranking of the categories.

To facilitate the following discussion, we first introduce the concepts of *mention* and *context*. For an input text t, a *mention* is a term in t that can refer to an entity e and the *context* of e is the set of all other mentions in t except the one for e. For each candidate entity e contained in t, the input text t can be decomposed into the mention and context of e, denoted by m_e and C_e, respectively. For example, given the entity e as IBM, the input text *"IBM adds midrange server to eServer lineup."* can be decomposed into a mention m_e as *"IBM"* and a context C_e as {*"midrange server", "eServer"*}, where *"midrange server"* and *"eServer"* can refer to the context entities Midrange_computer and IBM_eServer, respectively.

Based on the above introduced concepts, the joint probability $P(c,t)$ is given as follows:

$$\begin{aligned} P(c,t) &= \sum_{e \in E_t} P(e,c,t) = \sum_{e \in E_t} P(e,c,m_e,C_e) \\ &= \sum_{e \in E_t} P(e)P(c|e)P(m_e|e,c)P(C_e|e,c) \qquad (2) \\ &= \sum_{e \in E_t} P(e)P(c|e)P(m_e|e)P(C_e|e)\,, \qquad (3) \end{aligned}$$

where E_t represents the set of all possible entities contained in the input text t. We assume that in Eq. (2) m_e and C_e are conditionally independent given e, in Eq. (3) m_e and C_e are conditionally independent of c given e. The intuition behind these assumptions is that a mention m_e and a context C_e only rely on the entity e which refers to and co-occurs with, such that once the entity e is fixed, m_e and C_e can be considered as conditionally independent. The main problem is then to estimate each probability in Eq. (3), which will be discussed in the next section.

3.2 Parameter Estimation

Our probabilistic model has four main components, i.e., $P(e)$, $P(c|e)$, $P(m_e|e)$ and $P(C_e|e)$. This section provides the estimation of each component in detail.

Entity Popularity. The probability $P(e)$ captures the popularity of the entity e. Here, we simply apply a uniform distribution to calculate $P(e)$ as follows: $P(e) = \frac{1}{N}$, where N is the total number of entities in the KB.

Entity-Category Relatedness. The probability $P(c|e)$ models the relatedness between an entity e and a category c. With the pre-built entity and category embeddings (see Sect. 4), there are two cases to consider for estimating $P(c|e)$. Firstly, when the entity e is directly associated with the category, denoted by c_{a_e}, in KB, i.e., e appears in some Wikipedia articles that have associated category c_{a_e}, the probability $P(c_{a_e}|e)$ can be approximated based on similarity as

$$P(c_{a_e}|e) = \frac{sim(c_{a_e}, e)}{\sum\limits_{c'_{a_e} \in C_{a_e}} sim(c'_{a_e}, e)}, \tag{4}$$

where C_{a_e} is the set of categories that are directly associated with e, and $sim(c_{a_e}, e)$ denotes the cosine similarity between the vectors of the category c_{a_e} and the entity e in the embedding space. Secondly, in case where the entity e is not directly associated with the category c, the hierarchical structure of categories in KB is considered. More specifically, the categories in C_{a_e} are incorporated into the estimation of the probability $P(c|e)$ as follows:

$$P(c|e) = \sum_{c_{a_e} \in C_{a_e}} P(c_{a_e}, c|e) = \sum_{c_{a_e} \in C_{a_e}} P(c_{a_e}|e) P(c|c_{a_e}, e) = \sum_{c_{a_e} \in C_{a_e}} P(c_{a_e}|e) P(c|c_{a_e}), \tag{5}$$

where we consider that e is related to c only through its directly associated category c_{a_e}, such that once c_{a_e} is given, e and c are conditionally independent.

In Eq. (5), the probability $P(c_{a_e}|e)$ then can be simply calculated based on Eq. (4) and the probability $P(c|c_{a_e})$ that captures the hierarchical category structure, is estimated as follows:

$$P(c|c_{a_e}) = \begin{cases} \frac{1}{|A_{c_{a_e}}|} & \text{if } c \text{ is an ancestor of } c_{a_e}, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $A_{c_{a_e}}$ is the set of ancestor categories of c_{a_e}, which can be obtained by using the category hierarchy in KB.

Mention-Entity Association. The probability $P(m_e|e)$ of observing a mention m_e given the entity e is calculated based on the *Anchor-Text Dictionary* as follows:

$$P(m_e|e) = \frac{count(m_e, e)}{\sum\limits_{m'_e \in M_e} count(m'_e, e)}, \tag{7}$$

where $count(m_e, e)$ denotes the number of links using m_e as anchor text pointing to e as the destination, and M_e is the set of all mentions that can refer to e.

Entity-Context Relatedness. The probability $P(C_e|e)$ models the relatedness between the entity e and its context C_e that consists of all the other mentions in the input text t except m_e. Each mention in C_e refers to a context entity e_c from the given KB. The probability $P(C_e|e)$ can be calculated as follows:

$$\begin{aligned} P(C_e|e) &= \sum_{e_c \in E_{C_e}} P(e_c, C_e|e) = \sum_{e_c \in E_{C_e}} P(e_c|e)P(C_e|e_c, e) \\ &= \sum_{e_c \in E_{C_e}} P(e_c|e)P(C_e|e_c) \qquad (8) \\ &= \sum_{e_c \in E_{C_e}} P(e_c|e)P(m_{e_c}|e_c)\,, \qquad (9) \end{aligned}$$

where E_{C_e} denotes the set of entities that can be referred to by the mentions in C_e. In Eq. (8), the context C_e is conditionally independent of e given the context entity e_c, and in Eq. (9) e_c is assumed to be only related to its corresponding mention $m_{e_c} \in C_e$ such that the other mentions in C_e can be ignored.

Similar to $P(c_{a_e}|e)$ (cf Eq. (4)), the probability $P(e_c|e)$ in Eq. (9) can also be estimated based on the pre-built entity and category embeddings. Let $sim(e_c, e)$ be the cosine similarity between the entity vectors of e_c and e. Then the probability $P(e_c|e)$ can be calculated as follows:

$$P(e_c|e) = \frac{sim(e_c, e)}{\sum\limits_{e' \in E} sim(e', e)}\,, \qquad (10)$$

where E is the set of all entities in KB. In addition, the probability $P(m_{e_c}|e_c)$ in Eq. (9) can be calculated based on Eq. (7).

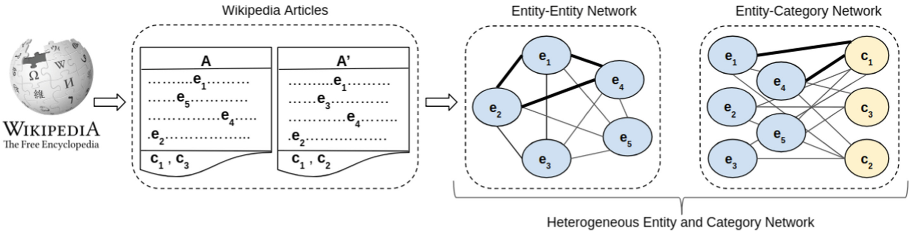

Fig. 2. Entity category network construction (best viewed in color) (Color figure online)

4 Entity and Category Embedding

This section provides a description of the proposed embedding model that embeds entities and categories into a common vector space by integrating knowledge from a knowledge base. We firstly present the entity-entity and entity-category network construction in Sect. 4.1, and subsequently, the joint entity and category embedding model is presented in Sect. 4.2.

4.1 Network Construction

To calculate the meaningful semantic relatedness between entities and categories, the proper semantic representation of them in a common vector space is essential for KBSTC. For this reason, two types of networks, i.e., entity-entity and entity-category, are firstly constructed, which are later utilized to generate the entity and category embeddings.

Figure 2 depicts the process of the entity-entity and entity-category network construction, where the *heterogeneous network* consists of both entity vertices and category vertices, and accordingly two types of edges, i.e., edges between two entity vertices and edges between an entity vertex and a category vertex. The weights of the edges between different vertices are crucial due to their significant impact on the embedding model (see Sect. 4.2). By leveraging the hyperlink structure in Wikipedia, we propose a method to calculate the edge weights for both entity-entity and entity-category networks.

Weights for Entity-Entity Edges. In order to explain the weight calculation, firstly the concept of *linked entity* has to be defined. The hyperlinks that are present in an arbitrary Wikipedia article and refer to another Wikipedia article are called linked entities. The weight of an edge between an entity-entity pair is the number of Wikipedia articles where both entities appear as a linked entity.

Weights for Entity-Category Edges. The weight of an edge between an entity-category pair is the number of Wikipedia articles where the entity appears as a linked entity and simultaneously the corresponding article containing the linked entity belongs to the category in Wikipedia.

As shown in Fig. 2, the linked entities and the associated categories for each Wikipedia article are used to generate the entity-entity and the entity-category edges. The edges of (e_1, e_2), (e_1, e_4), (e_2, e_4), (e_1, c_1) and (e_4, c_1) are thicker due to their higher co-occurrence frequency.

4.2 Embedding Model

As introduced before, the overall heterogeneous network consists of two homogeneous networks, i.e., the entity-entity and entity-category networks. Similar to PTE [15], to embed each of these networks, our proposed embedding model aims to capture the second-order proximity [16]. More specifically, the second-order proximity is calculated between two vertices in a network by considering their common (shared) vertices. Therefore, vertices that share many same neighbors should be placed closely in the vector space.

To model the second-order proximity of a homogeneous network, for each edge (v_i, v_j), the conditional probability $p(v_j|v_i)$ is defined as follows [16]:

$$p(v_j|v_i) = \frac{exp(-\boldsymbol{u}_j^T \cdot \boldsymbol{u}_i)}{\sum\limits_{v_k \in V} exp(-\boldsymbol{u}_k^T \cdot \boldsymbol{u}_i)}\,, \tag{11}$$

where V is the set of vertices connected with v_i in the network, $\boldsymbol{u}_i$, $\boldsymbol{u}_j$ and $\boldsymbol{u}_k$ are the vectors of vertices v_i, v_j and v_k, respectively. The empirical probability of $p(v_j|v_i)$ can be defined as $\hat{p}(v_j|v_i) = \frac{w_{ij}}{d_i}$, where d_i is the out-degree of v_i and w_{ij} is the weight of the edge (v_i, v_j).

In order to preserve the second-order proximity, the conditional distribution $p(v_j|v_i)$ is made close to $\hat{p}(v_j|v_i)$ based on the KL-divergence over the entire set of vertices in the network, such that the model minimizes the following objective function:

$$O_{homo} = - \sum_{(v_i,v_j)\in E} w_{ij} \log\left(p(v_j|v_i)\right), \tag{12}$$

The embedding of the individual entity-entity and entity-category networks can be learned by utilizing the second-order proximity between vertices. However, our goal is to simultaneously learn the embedding of the constructed heterogeneous network by minimizing the following objective function:

$$O_{heter} = O_{ee} + O_{ec}, \tag{13}$$

where O_{ee} and O_{ec} are the objective functions defined in Eq. (12) for the homogeneous entity-entity and entity-category networks, respectively. To optimize the objective function in Eq. (13), we adopt a similar approach as described in [15], where all the edges are firstly collected from these two homogeneous networks as two sets, one for entity-entity edges and the other for entity-category edges, and then in each training iteration, edges are sampled from both sets to update the model. Readers can refer to [15,16], for the detailed optimization process.

5 Experimental Results

This section provides a detailed description of the datasets and the baselines for evaluating the proposed approach, followed by the experimental results as well as a comparison to the existing state-of-the-art approaches in the related areas.

Table 1. Data distribution of the AG News dataset

Category	#Train	#Test
Business	30,000	1,900
Sports	30,000	1,900
World	30,000	1,900
Sci/Tech	30,000	1,900
Total	120,000	7,600

Table 2. Data distribution of the Google Snippets dataset

Category	#Train	#Test
Business	1200	300
Computers	1200	300
Cult-arts-entertainment	1880	330
Education-Science	2360	300
Engineering	220	150
Health	880	300
Politics-Society	1200	300
Sports	1120	300
Total	10,060	2,280

5.1 Datasets

AG News (AG)[1]**:** This dataset is adopted from [23], which contains both titles and short descriptions (usually one sentence) of news articles. The data distribution of the training and test datasets is shown in Table 1. In our experiments, the dataset has two versions, where one contains only titles and the other contains both titles and descriptions. The total number of entities and the average number of entities and words per text in the test datasets are shown in Table 3.

Google Snippets (Snippets)[2]**:** This is a well-known dataset for short text classification, which was introduced in [10] and contains short snippets from Google search results. The data distribution of the dataset is shown in Table 2. As shown in Table 3, the test dataset has in total 20,284 entities, an average of 8.9 entities and an average of 17.97 words in each snippet.

Table 3. Statistical analysis of the test datasets

Dataset	#Entities	Avg. #Ent	Avg. #Word
AG News (Title)	24,416	3.21	7.14
AG News (Title+Description)	89,933	11.83	38.65
Google Snippets	20,284	8.90	17.97

As the focus of this work is the KBSTC task, where the goal is to derive the most relevant category from the knowledge base for a given short text, we need to adapt these datasets by aligning the labels/categories with the categories in the used knowledge base. More specifically, each label/category in these datasets is manually mapped to its corresponding Wikipedia category, e.g., the category *Sports* from the AG dataset is mapped to the Wikipedia category *Sports*[3]. Furthermore, as KBSTC does not depend on any training/labeled data, the training

Table 4. The classification accuracy of KBSTC against baselines (%)

Model	AG (title)	AG (title+description)	Snippets
Dataless ESA [14]	53.5	64.1	48.5
Dataless Word2Vec [14]	49.5	52.7	52.4
NB+TF-IDF	86.6	90.2	64.4
SVM+TF-IDF	**87.6**	**91.9**	69.1
LR+TF-IDF	87.1	91.7	63.6
KBSTC+Our Embedding	67.9	80.5	**72.0**

[1] http://goo.gl/JyCnZq.

[2] http://jwebpro.sourceforge.net/data-web-snippets.tar.gz.

[3] https://en.wikipedia.org/wiki/Category:Sports.

datasets of AG and Snippets are only used for the training of the supervised baseline methods. Lastly, to measure the performance of KBSTC, the classification accuracy (the ratio of correctly classified data over all the test data) was used.

5.2 Baselines

Dataless ESA and Dataless Word2Vec: As described in Sect. 2, the dataless approaches do not require any labeled data or training phase, therefore, they can be considered as the most similar approaches to KBSTC. Two variants of the state-of-the-art dataless approach [14] are considered as baselines, which are based on ESA [3] and Word2Vec [7], respectively.

NB, SVM, LR: Additional baselines include the traditional supervised classifiers, i.e., Naive Bayes (NB), Support Vector Machine (SVM) and Logistic Regression (LR), with the features calculated based on the term frequency and the inverse document frequency (TF-IDF).

5.3 Evaluation of KBSTC

Table 4 shows that the accuracy of the proposed probabilistic KBSTC approach (see Sect. 3) based on our entity and category embedding model (see Sect. 4) in comparison to the baselines on the AG and Snippets datasets.

It is observed that the KBSTC approach considerably outperforms the dataless classification approaches. While Dataless ESA and Dataless Word2Vec have been assessed with longer news articles and achieved promising results in [14], they cannot perform well with short text due to the data sparsity problem.

Remarkably, KBSTC outperforms all the baselines on the Snippets dataset, however, all supervised approaches outperform KBSTC on the AG dataset. The reason here can be attributed to the different characteristics of the two datasets. AG is a larger dataset with more training samples (see Table 1) in comparison to Snippets (see Table 2). Moreover, the AG dataset provides only 4 different categories in comparison to 8 categories of the Snippets dataset. Those differences might be the reason of the significant decrease in accuracy for the supervised approaches on the Snippets dataset in comparison to the AG dataset. This could be an indicator that the size of the training data and the number of classes make a real impact on the classification accuracy for the supervised approaches. Since KBSTC does not require or use any labeled data, the number of the available training samples has no impact on its accuracy.

Regarding the results of KBSTC, the AG (title+description) dataset yields better accuracy than the Snippets dataset, which in turn, results in better accuracy than the AG (title) dataset. The reason might be found in the nature of the datasets. As shown in Table 3, the average number of entities per text in AG (title+description) is greater than Snippets, followed by AG (title). Often a richer context with more entities can make the categorization more accurate.

Overall, the results in Table 4 have demonstrated that for short text categorization, KBSTC achieves a high accuracy without requiring any labeled data, a time-consuming training phase, or a cumbersome parameter tuning step.

Table 5. The classification accuracy of KBSTC with different embedding models (%)

Model	AG (title)	AG (title+description)	Snippets
KBSTC+HCE	67.0	79.6	**72.3**
KBSTC+DeepWalk	57.1	74.2	64.3
KBSTC+RDF2Vec	62.7	77.5	68.2
KBSTC+Our Embedding	**67.9**	**80.5**	72.0

5.4 Evaluation of Entity and Category Embedding

To assess the quality of the proposed entity and category embedding model (see Sect. 4), we compared it with HCE [5], DeepWalk [9] and RDF2Vec [11] in the context of the KBSTC task.

While the Wikipedia entity and category embeddings generated by HCE can be directly used, DeepWalk has been applied on the network constructed using Wikipedia and RDF2Vec has been applied on the RDF graph of DBpedia to obtain the needed embeddings. Then, these embeddings are integrated into KBSTC to compute the entity-category and entity-context relatedness (see Eqs. (4) and (10)). The results of KBSTC with different embedding models are shown in Table 5. The proposed entity and category embedding model outperforms all other embedding models for the KBSTC task on the AG dataset, while HCE performs slightly better than our model on the Snippets dataset.

As HCE is a more specific embedding model that has been designed to learn the representation of entities and their associated categories from Wikipedia, it is not flexible to be adapted to other networks. In contrast, our model can deal with more general networks. For example, with words and word-category relations as an additional type of vertices and edges in the heterogeneous network described in Sect. 4.1, it is straightforward to adapt our embedding model by involving a new object function O_{wc} into Eq. (13), which is considered as our future work.

Although DeepWalk and RDF2Vec aim to learn the representation of vertices in general networks and RDF graphs, respectively, they have been either designed for homogeneous networks or treated each type of vertices and edges in a RDF graph equally. The results also indicate that our embedding model enables to capture better semantic representation of vertices by taking into account different types of networks, i.e., the entity-entity and entity-category networks.

5.5 Evaluation of Entity Linking

As discussed in Sect. 3, the first step of KBSTC is to detect entity mentions in a given short text and then for each mention to generate a candidate list of entities based on the anchor text dictionary, which are employed to determine the most relevant category for the input text based on the proposed probabilistic approach. An alternative way could be to firstly use an existing entity linking (EL) system to obtain the referent entity for each mention and then based on

Table 6. Statistics of the entity linking datasets

Dataset	#Doc	Avg. #Ent	Avg. #Word
Spotlight	58	5.69	32
RSS-500	500	1.18	34

Table 7. Micro F1 results for the entity linking task

Methods	Spotlight	RSS-500
AIDA	0.25	0.45
AGDISTS	0.27	**0.66**
Babelfly	0.52	0.44
DBpedia Spotlight	**0.71**	0.20
Our EL Method	0.69	0.64

that derive the category of the input short text. The reason we did not adopt the latter solution is that most of the existing EL systems rely on the rich context of the input text for the collective inference to boost the overall EL performance. However, due to the lack of such context in short text, existing EL systems might not perform well in our case, i.e., the correct entities in the input short text cannot be found, which play a vital role in our KBSTC approach.

Instead of directly using an existing EL system, our probabilistic approach actually involves an internal step of EL for the input short text t, where the main difference is that we consider a list of candidate entities for each mention. The output is a set of possible entities E_t present in t with the confidence score of each entity $e \in E_t$ as $P(e)P(m_e|e)P(C_e|e)$ (see Eq. (3)), where $P(e)$ captures the popularity of e, $P(m_e|e)$ and $P(C_e|e)$ reflect the likelihood of observing the mention m_e and the context C_e given e. By incorporating the confidence score of each $e \in E_t$ and its relatedness to each predefined category c, represented by $P(c|e)$, we can compute the final joint probability $P(c, t)$ to determine the most relevant category for t (see Eq. (3)).

To evaluate the effectiveness of the EL step in our approach, the experiments have been conducted on two datasets from the general entity linking benchmark GERBIL [19], i.e., DBpedia Spotlight released in [6] and N^3 RSS-500 as one of the N^3 datasets [12]. We have chosen these two datasets for the EL evaluation, because they contain only short text, similar to our test datasets (see Table 6). To make our EL method be comparable with existing EL systems, in the experiments we also generate one single entity for each mention, which maximizes the confidence score, computed by $P(e)P(m_e|e)P(C_e|e)$. The results of Micro F1 for various EL systems and our method are shown in Table 7. It is observed that our EL method achieves promising results for both datasets, which are very close to the best results yielded by the state-of-the-art EL systems. More importantly, because of insufficient context of short text required by the collective inference for EL, it is difficult to provide the correct referent entity for each mention in many cases, such that our EL method used in KBSTC takes into account a list of candidate entities with their confidence scores for each mention.

5.6 Using Wikipedia as a Training Set

To further demonstrate the effectiveness of the proposed KBSTC approach, an additional experiment has been conducted. The results in Table 4 indicates that supervised methods can perform well in case of existence of sufficient amount of training data. However, the labeled data might not be available and this is the case most of the time. An alternative solution to the expensive manual process of compiling a labeled training dataset would be to automatically extract the training data from existing publicly available sources such as Wikipedia.

Table 8. The classification accuracy of KBSTC against a traditional classifier (%)

Method	AG (title+description)	Google Snippets
SVM+TF-IDF	59.9	53.9
KBSTC	**80.5**	**72.0**

To generate the training data, for each category from the two datasets (AG and Snippets), training samples have to be assembled. For this purpose, Wikipedia articles associated with the corresponding categories (or their subcategories) are firstly collected, where 10,000 Wikipedia articles are then randomly selected as training data per category, which constitute the training datasets for AG and Snippets. Since SVM achieved the best results among the supervised approaches (see Table 4), two SVM classifiers are trained with the generated training data for AG and Snippets, respectively. In the experiments, we used the original test datasets from AG and Snippets for evaluating the trained SVM classifiers.

The results are shown in Table 8, which indicate that the KBSTC approach achieved higher accuracy in comparison to the SVM classifiers. More interesting, the same approach (SVM+TF-IDF) trained with the AG and Snippets datasets achieved the accuracy scores of 91.9% and 69.1% (see Table 4), while it only achieved the accuracy scores of 59.9% and 53.9% when trained with the collected Wikipedia articles. This provides us some insights that it might not be suitable to directly use Wikipedia as the training datasets for supervised approaches and also serves as the motivation of the KBTSC approach proposed in this work.

6 Conclusion and Future Work

We have proposed KBSTC, a new paradigm for short text categorization based on KB. KBSTC does not require any labeled training data, instead it considers entities present in the input text and their semantic relatedness to the predefined categories to categorize short text. The experimental results have proven that it is possible to categorize short text in an unsupervised way with a high accuracy. As for future work, we aim to include words along with entities for the KBSTC task, which requires also the extension of the proposed embedding model towards the additional inclusion of word embeddings into the common entity and category vector space. Further, the performance of KBSTC will also be evaluated on social media text such as tweets.

References

1. Burel, G., Saif, H., Alani, H.: Semantic wide and deep learning for detecting crisis-information categories on social media. In: d'Amato, C., et al. (eds.) ISWC 2017. LNCS, vol. 10587, pp. 138–155. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68288-4_9
2. Chang, M.W., Ratinov, L.A., Roth, D., Srikumar, V.: Importance of semantic representation: dataless classification. In: AAAI (2008)
3. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: IJCAI (2007)
4. Li, C., Xing, J., Sun, A., Ma, Z.: Effective document labeling with very few seed words: a topic model approach. In: CIKM (2016)
5. Li, Y., Zheng, R., Tian, T., Hu, Z., Iyer, R., Sycara, K.P.: Joint embedding of hierarchical categories and entities for concept categorization and dataless classification. In: COLING (2016)
6. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: Dbpedia spotlight: shedding light on the web of documents. In: I-SEMANTICS (2011)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS (2013)
8. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.M.: Text classification from labeled and unlabeled documents using EM. Mach. Learn. **39**, 103–134 (2000)
9. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: KDD (2014)
10. Phan, X.H., Nguyen, L.M., Horiguchi, S.: Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: WWW (2008)
11. Ristoski, P., Paulheim, H.: RDF2Vec: RDF graph embeddings for data mining. In: Groth, P., et al. (eds.) ISWC 2016. LNCS, vol. 9981, pp. 498–514. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46523-4_30
12. Röder, M., Usbeck, R., Hellmann, S., Gerber, D., Both, A.: N^3 - a collection of datasets for named entity recognition and disambiguation in the NLP interchange format. In: LREC (2014)
13. Song, G., Ye, Y., Du, X., Huang, X., Bie, S.: Short text classification: a survey. J. Multimedia **9**(5), 635–644 (2014)
14. Song, Y., Roth, D.: On dataless hierarchical text classification. In: AAAI (2014)
15. Tang, J., Qu, M., Mei, Q.: PTE: predictive text embedding through large-scale heterogeneous text networks. In: KDD (2015)
16. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: large-scale information network embedding. In: WWW (2015)
17. Türker, R., Zhang, L., Koutraki, M., Sack, H.: TECNE: knowledge based text classification using network embeddings. In: EKAW (2018)
18. Türker, R., Zhang, L., Koutraki, M., Sack, H.: "The less is more" for text classification. In: SEMANTiCS (2018)
19. Usbeck, R., et al.: GERBIL: general entity annotator benchmarking framework. In: WWW (2015)

20. Wang, J., Wang, Z., Zhang, D., Yan, J.: Combining knowledge with deep convolutional neural networks for short text classification. In: IJCAI (2017)
21. Wang, P., Xu, B., Xu, J., Tian, G., Liu, C.L., Hao, H.: Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. Neurocomputing **174**, 806–814 (2016)
22. Xuan, J., Jiang, H., Ren, Z., Yan, J., Luo, Z.: Automatic bug triage using semi-supervised text classification. In: SEKE (2010)
23. Zhang, X., LeCun, Y.: Text understanding from scratch. CoRR (2015)
24. Zhang, X., Wu, B.: Short text classification based on feature extension using the n-gram model. In: FSKD. IEEE (2015)