Short Text Categorization Using Joint Entity And Category Embeddings

Rima Türker
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany
Karlsruhe Institute of Technology, Institute AIFB, Germany
rima.tuerker@fiz-karlsruhe.de

Lei Zhang
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany
lei.zhang@fiz-karlsruhe.de

Maria Koutraki
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany
Karlsruhe Institute of Technology, Institute AIFB, Germany
maria.koutraki@fiz-karlsruhe.de

Harald Sack
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure, Germany
Karlsruhe Institute of Technology, Institute AIFB, Germany
harald.sack@fiz-karlsruhe.de

ABSTRACT

Short text categorization is an important and challenging task due to its application in various domains such as document organization and news filtering. Most of the traditional methods suffer from sparsity and shortness of the text. Moreover, supervised learning methods require a significant amount of training data. However, 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 using state-of-the-art network embedding techniques. Given a short text, its category can then be derived based on the entities mentioned in the text by exploiting semantic similarity between entities and categories according to their vector representations. To validate the effectiveness of the proposed method, we conducted experiments on two real-world datasets, i.e., AG News and Google Snippets, which show that our approach achieves comparable results to existing methods that require manually labeled data.

KEYWORDS

Short Text Classification, Dataless Text Classification, Network Embeddings

ACM Reference Format:
Rima Türker, Lei Zhang, Maria Koutraki, and Harald Sack. 1997. Short Text Categorization Using Joint Entity And Category Embeddings. In *Proceedings of ACM Woodstock conference (CIKM'18).* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Text categorization [9, 23, 26, 27] is gaining more and more attention due to the availability of a huge number of text data, which includes search snippets, news data as well as text data generated in social networks.

Recently, several deep learning approaches have been proposed for text classification [5, 12, 14, 23, 26, 27]. However, they all require a significant amount of labeled training data. Manual labeling of such data can be a very 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 [18, 25] have been proposed to reduce the labeling effort. Yet, due to the diversity of the documents in many applications, generating small training set for the semi-supervised approaches still remains an expensive process [15].

To overcome the problem of labeled data, a number of *dataless text classification* methods have been proposed [2, 20]. These methods do not require any labeled data as a prerequisite. Instead, they rely on the semantic similarity between a set of predefined categories and a given document to determine which category the given document belongs to. More specifically, categories as well as documents are represented in a common vector space, which allows to calculate a meaningful semantic relatedness between documents and category labels. The classification process depends on this semantic relatedness. However, the most prominent and successful dataless classification approaches are designed for long (natural language text) documents.

By considering *short text categorization*, most of the standard text classification approaches suffer from issues such as data sparsity and insufficient text length [3]. Simple text classification approaches based on bag of words (BOW) cannot properly represent short text as the semantic relatedness between words is not taken into account [23]. Approaches that operationalize word embeddings for classification perform better when dealing with longer text, in which case, even if a word is ambiguous, such ambiguity will

CIKM'18, October 2018, Lingotto,Turin,Italy
© 2018 Copyright held by the owner/author(s).
ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

be handled based on the given context. 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.

In this paper, we propose a novel probabilistic model for *dataless* Knowledge-Based Short Text Categorization (KBSTC), which does not require any labeled training data. Our approach differs from the traditional text categorization approaches since it is designed for the task of *short* text categorization. It is able to capture the semantic relation between the entities represented in a short text and the predefined categories by embedding them into a common vector space using network embedding techniques. Finally, the category of the given text sentence can be derived based on the semantic similarity between entities (present in the given text) and a 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 dataless short text categorization approach;
- a joint entity and category embedding method;
- an experimental evaluation using standard datasets for short text categorization.

The rest of this paper is structured as follows: Section 2 introduces related work and tries to differentiate the work presented in this paper from related approaches. In Section 3, the approach followed in this paper is explained. Section 4 presents the joint entity and category embeddings used in this approach, while Section 5 describes the experimental setup for the evaluation as well as the applied baselines. It further illustrates and discusses the achieved results. Last, Section 6, concludes the paper with a discussion of open issues as well as of ongoing and future work.

2 RELATED WORK

There exists a considerable amount of studies in the area of Text Classification [7, 11]. Most standard approaches calculate a simple feature set based on the words present in the documents such as Term Frequency and Inverse Document Frequency (TF-IDF). Subsequently, based on the feature sets, supervised learning approaches are applied such as Support Vector Machines (SVM), Naive Bayes or Logistic Regression, which rely on a sufficiently large labeled training set. However, the aim of this work is to classify a given short text without requiring any labeled data for training. Thus, *Dataless Text Classification* tasks can be considered as closely related to our approach, which is addressed in the first section of related work. However, the reviewed *Dataless Text Classification* studies have all been designed for the classification of documents of arbitrary length. For that reason, in the subsequent section also short text classification methods are discussed.

Dataless Text Classification. Chang et al. [2] introduced a dataless text classification method by representing documents as well as labels in a common semantic space. As source the online encyclopaedia Wikipedia was utilized supported with Explicit Semantic Analysis (ESA)[8] to quantify semantic relatedness between the labels to be assigned and the documents. As a result it was shown that ESA was able to achieve better classification results than the traditional bag of words (BOW) representations. Further, Song and Roth [20] proposed 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 to iteratively improve classification accuracy. Also Latent Dirichlet Allocation (LDA) was utilized for dataless text classification [10, 15]. Chen et al. [4] proposed an LDA based dataless classification method referred to as Descriptive LDA (DescLDA). DescLDA is an extension of the standard LDA model, in which a *describing device (DD)* is applied. The DD is a model which helps to infer Dirichlet priors from categories/labels of the given dataset.

In contrast to the mentioned Dataless Text Classification approaches, our proposed approach differs in two main aspects. First, all mentioned studies were designed for the classification of documents of arbitrary length. However the main purpose of this study 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.

Short Text Classification. Zhang et al. [27] proposed a character-level convolutional neural network based text classification approach for short text classification. The result of this approach has been compared with multinomial logistic regression, in which BOW, ngrams and ngram-TF-IDF are considered as feature sets. Another neural network based short text classification approach was studied by Wang et al. [23]. They compare their approach with an SVM classifier trained by TF-IDF and paragraph vectors as feature set and showed that their presented approach achieved superior results.

To overcome the sparsity problem of short text documents Chen et al. [3] proposed a method that models the short text more precisely by leveraging topics at multiple granularities. In order to expand the semantic representation of short text documents, Phan et al. [19] proposed an approach to discover hidden topics based on an external Wikipedia corpus using LDA.
There are also more sophisticated deep learning approaches, as e.g. [12, 14], which have been proposed for text classification. [26] proposed a deep convolution neural network to learn representations of words from their characters and proved that for text classification, a deep CNN performs well on character level. Another study utilized both convolutional and recurrent neural networks for character-level document classification without any explicit segmentation [24]. Furthermore, [5] for the first time applied a Very Deep CNN (VDCNN) to improve the learning capacity for syntactic embeddings applied for text classification.

While the presented related approaches of this section achieve very good performance in practice, they are usually slow both in the training as well as the testing phase. Moreover, their performance highly depends on the training data size, its distribution, and the chosen hyper parameters. However, our approach does not require any training data or any parameter tuning.

3 APPROACH

This section contains a formal definition of Knowledge-based Short Text Categorization (KBSTC), an example to illustrate the given definition, followed by the description of the proposed probabilistic approach for the KBSTC task.

Knowledge-based Short Text Categorization (KBSTC). Given a Knowledge Base KB, containing a set of entities $E = \{e_1, e_2, .., e_n\}$ and a set of hierarchically related categories $C = \{c_1, c_2, .., c_m\}$, where each entity $e_i \in E$ is associated with a set of categories $C' \subseteq C$ via a relation $cat \subseteq E \times C$, such that $cat(e_i) = C'$.

Let $G_{KB} = (V, ED)$ be a graph, with $V = E \cup C$ as the set of vertices and $ED = ED_{EE} \cup ED_{EC} \cup ED_{CC}$ as the set of edges consisting of entity-entity pairs $ED_{EE} = \{(e_i, e_j)|e_i, e_j \in E\}$ that represent semantic relations among entities in KB, entity-category pairs $ED_{EC} = \{(e_i, c_j)|e_i \in E, c_j \in cat(e_i) \subseteq C\}$, and category-category pairs $ED_{CC} = \{(c_i, c_j)|c_i, c_j \in C\}$ that represent category hierarchies within KB.

The input of the KBSTC task is defined as a short text t which contains a set of mentions $M_t = \{m_1, \ldots, m_k\}$ that uniquely refer to a set of entities $E_t \subseteq E$ such that $EL(m_i) = e_j, e_j \in E_t, m_i \in M_t$ as well 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 $f_{cat} : E \times C' \to C'$ with $f_{cat}(t, C') = c_i$ where $c_i \in C'$.

The following example illustrates the KBSTC task. Given the short text t and the predefined categories C', with
t : *"IBM adds midrange server to eServer lineup"*, and
$C' = \{Sports, Technology, Culture, World\}$,
the main goal of the KBSTC task is to assign the most relevant category $c_i \in C'$ to t, i.e. $f_{cat}(t, C') = Technology$.

KBSTC Overview. The general work flow of KBSTC is shown in Figure 1. The first step is *"Mention Detection Based on Anchor-Text Dictionary"*, where each entity mention present in t is detected based on a prefabricated *"Anchor-Text Dictionary"* from Wikipedia. The Anchor-Text Dictionary contains all mentions and their corresponding Wikipedia entities. In order to construct an Anchor-Text Dictionary all the anchor texts of hyperlinks in Wikipedia articles referring to another Wikipedia article are extracted, whereby the anchor texts serve as mentions and the Wikipedia article links refer to the corresponding entities. In the second step, for each detected mention in the given input text, candidate entities are generated based on the Anchor-Text Dictionary. In our example these are "IBM", "Midrange_computer" and "IBM_eServer". Likewise the predefined categories are mapped to Wikipedia categories. Finally, based on the entity and category embeddings that have been precomputed from Wikipedia, the output of the KBSTC task is the semantically most related category for the given entities. Thereby, in the given example the category *Technology* should be determined. The KBSTC approach will be explained in details in the following sections.

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. Therefor, 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 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:

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

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 and a context only rely on the entity it refers to and co-occurs with. 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., $P(e)$, $P(c|e)$, $P(S_e|e)$ and $P(C_e|e)$. This section provides the estimation of each component in details.

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:

$$P(e) = \frac{1}{N} \tag{4}$$

where N is the total number of entities in the knowledge base.

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 Section 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 the knowledge base, i.e., e appears in some Wikipedia articles that have

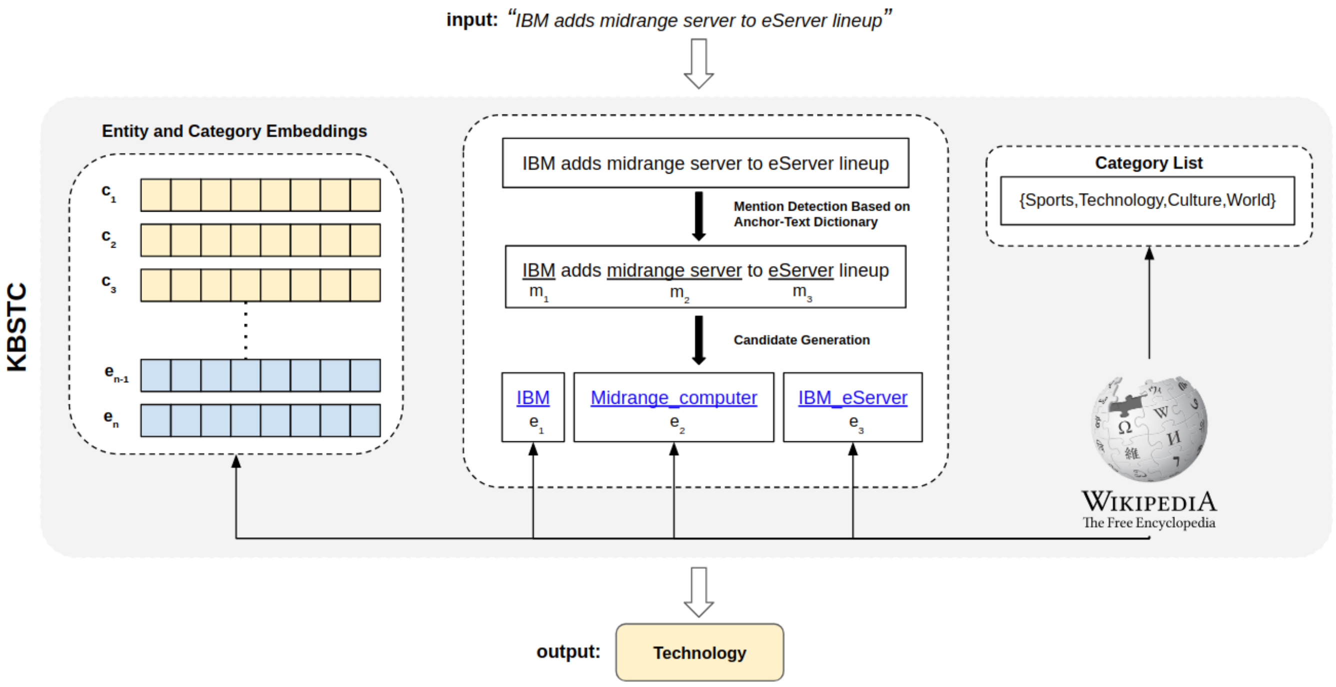
input: "IBM adds midrange server to eServer lineup"
KBSTC
Entity and Category Embeddings
c1
c2
c3
en-1
en
IBM adds midrange server to eServer lineup
Mention Detection Based on Anchor-Text Dictionary
IBM adds midrange server to eServer lineup
m1
m2
m3
Candidate Generation
IBM
e1
Midrange_computer
e2
IBM_eServer
e3
Category List
{Sports,Technology,Culture,World}
WIKIPEDIA
The Free Encyclopedia
output:
Technology

Figure 1: The work flow of the proposed KBSTC approach (best viewed in color)

the links to c_{a_e}, the probability $P(c_{a_e}|e)$ can be estimated as

$$P(c_{a_e}|e) = \frac{sim(c_{a_e}, e)}{\sum_{c'_{a_e} \in C_{a_e}} sim(c_{a_e}, e)} \qquad (5)$$

where C_{a_e} is the set of categories that 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 the knowledge base is considered. More specifically, the categories in C_{a_e} are incorporated into the estimation of the probability $P(c|e)$ as follows:

$$\begin{aligned} P(c|e) &= \sum_{c_{a_e} \in C_{a_e}} P(c_{a_e}, c|e) \\ &= \sum_{c_{a_e} \in C_{a_e}} P(c_{a_e}|e) P(c|c_{a_e}, e) \\ &= \sum_{c_{a_e} \in C_{a_e}} P(c_{a_e}|e) P(c|c_{a_e}) \qquad (6) \end{aligned}$$

In Eq. (6), we assume that the category c and e are conditionally independent given the directly associated category c_{a_e} of e.

Then the probability $P(c_{a_e}|e)$ in Eq. (6) can be simply calculated based on Eq. (5). In addition, the probability $P(c|c_{a_e})$ that captures the hierarchical category structure, is estimated as follows:

$$P(c|c_{a_e}) = \begin{cases} \frac{1}{|A_{c_{a_e}}|} & \text{if } c \text{ is an ancestor of } c_{a_e}, \\ 0 & \text{otherwise} \end{cases} \qquad (7)$$

where $A_{c_{a_e}}$ is the set of ancestor categories of c_{a_e}.

Mention-Entity Association. The probability of observing a mention m_e given an entity e is calculated based on an *anchor text dictionary*, which is constructed by collecting all the anchor texts (i.e., mentions) and their associated entities from Wikipedia. Based on that, the probability $P(m_e|e)$ is calculated as follows:

$$P(m_e|e) = \frac{count(m_e, e)}{\sum_{m'_e \in M_e} count(m'_e, e)} \qquad (8)$$

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. The context C_e consists of all the other mentions in the input text except m_e. Each mention in C_e refers to a context entity e_c from the given knowledge base. The probability $P(C_e|e)$ can be calculated as

$$\begin{aligned} P(C_e|e) &= \sum_{e_c \in E_{C_e}} P(e_c, C_e|e) \\ &= \sum_{e_c \in E_{C_e}} P(e_c|e) P(C_e|e_c, e) \\ &= \sum_{e_c \in E_{C_e}} P(e_c|e) P(C_e|e_c) \qquad (9) \\ &= \sum_{e_c \in E_{C_e}} P(e_c|e) P(m_{e_c}|e_c) \qquad (10) \end{aligned}$$

where E_{C_e} denotes the set of entities that can be referred to by the mentions in C_e. We assume that in Eq. (9) the context C_e is conditionally independent of e given the context entity e_c, and in

Eq. (10) e_c is 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 $P(c_s|e)$, the probability $P(e_c|e)$ in Eq. (10) can 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:

$$P(e_c|e) = \frac{sim(e_c, e)}{\sum_{e' \in E} sim(e', e)} \tag{11}$$

where E denotes the set of all entities in the knowledge base. In addition, the probability $P(m_{e_c}|e_c)$ can be calculated based on Eq. (8).

4 JOINT ENTITY AND CATEGORY EMBEDDING

This section contains a description of the proposed embedding method that embeds entities and categories into a common vector space by integrating knowledge from a large knowledge base. The generated embedding model is based on LINE [21], a state-of-the-art network embedding technology. We firstly present the entity-category network construction in Section 4.1, and then based on that introduce the joint embedding model for both entities and categories in Section 4.2.

4.1 Network Construction

The semantic representation of a text is crucial for text classification algorithms. Traditional text classification approaches use words to generate a feature set, and adopt machine learning algorithms. However, it is assumed that in short text words tend to be ambiguous, however entities carry much more information [22]. Therefore, in this study entities and their semantic relation with categories are used as a main feature to categorize the given short text.

To calculate the meaningful semantic relatedness, the proper semantic representation of entities and categories in a common vector space is essential for the proposed approach. For this reason, the entity-category network is firstly constructed, which will be later utilized to generate the entity and category embeddings.

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

Weights for Entity-Entity Edges. In order to explain the weight calculation, first the concept of a *linked entity* has to be defined. The hyperlinks which are present inside 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 article belongs to the category in Wikipedia.

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

4.2 Embedding Model

For joint embedding of entities and categories, we adopt a generic network embedding model, i.e., LINE [21], which is able to scale to very large, directed or undirected, weighted or unweighted networks. The model optimizes the objective functions, which preserve the *first-order proximity* and the *second-order proximity* between the nodes in a network.

First-order proximity. The first-order proximity is calculated for each pair of nodes that are directly connected by an edge. The weight of the edge indicates the first-order proximity between these two nodes. Therefore, it is expected that nodes that are connected through a strong edge (i.e., with high weight) should be placed closely in the vector space.

To model the first-order proximity, for each edge (i, j), the joint probability between nodes v_i and v_j is defined as follows:

$$p_1(v_i, v_j) = \frac{1}{1 + exp(-\vec{u}_i^T \cdot \vec{u}_j)} \tag{12}$$

where $\vec{u}_i$ ($\vec{u}_j$) is the vector representation of node v_i (v_j). In addition, its empirical probability can be defined as $\hat{p_1}(v_i, v_j) = \frac{w_{ij}}{W}$, where $W = \sum_{(i,j) \in E} w_{ij}$, E is the set of edges between nodes in the network, and w_{ij} is the weight of the edge (i, j). In order to preserve the first-order proximity, the model aims to minimize the KL-divergence between the two distributions $p_1(v_i, v_j)$ and $\hat{p_1}(v_i, v_j)$. By omitting some constants, the final goal is to minimize the following objective function:

$$O_1 = -\sum_{(i,j) \in E} w_{ij} \log p_1(v_i, v_j) \tag{13}$$

Second-order proximity. The second-order proximity is calculated between two nodes in a network by considering their common (shared) nodes. Therefore, nodes that share many same neighbors should be placed closely in the vector space.

To model the second-order proximity, for each edge (i, j), the conditional probability is defined as follows:

$$p_2(v_j|v_i) = \frac{exp(-\vec{u}_j^T \cdot \vec{u}_i)}{\sum_{k=1}^{|V|} exp(-\vec{u}_k^T \cdot \vec{u}_i)} \tag{14}$$

where V is the set of nodes connected with v_i in the network. The empirical probability of $p_2(v_j|v_i)$ can be defined as $\hat{p_2}(v_j|v_i) = \frac{w_{ij}}{d_i}$, where d_i is the out-degree of v_i. In order to preserve the second-order proximity, the conditional distribution $p_2(v_j|v_i)$ is made close to $\hat{p_2}(v_j|v_i)$ based on the KL-divergence over the entire set of nodes in the network, such that the model minimizes the following objective function:

$$O_2 = -\sum_{(i,j) \in E} w_{ij} \log p_2(v_j|v_i) \tag{15}$$

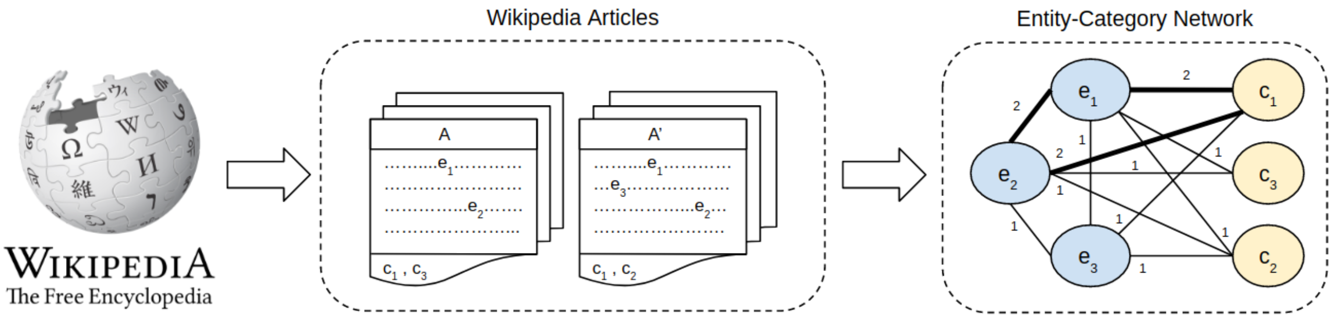

Figure 2: Entity-Category Network Construction

In order to keep both first-order and second-order proximities for each node in our constructed entity-category network, two LINE models are trained. Firstly, a LINE model is trained by preserving the first-order proximity, and then another LINE model is trained by preserving the second-order proximity. Finally, concatenating the embeddings of both models yields an embedding for each node. In this study, all the experiments are conducted with the embeddings where both first-order and second-order proximities are preserved.

5 EXPERIMENTAL RESULTS

This section contains a detailed explanation of the datasets and the experiments that are conducted to evaluate the proposed approach on short text categorization, as well as a comparison to existing state-of-the-art approaches.

5.1 Datasets

To validate the effectiveness of the proposed method experiments are conducted on two different benchmarks: AG News[1] and Google Snippets[2].

Google Snippets: The Google Snippets is a very well known short text classification dataset. The dataset consists of 10,060 training and 2,280 test snippets of 8 categories as shown in Table 1. The dataset was introduced in [19] as a short text classification benchmark. Since then, several short text classification approaches [1, 3, 6, 23] have used it as a validation dataset. The test set has an average of 6.3 mentions in each snippet as shown in Table 3.

AG News: This dataset is adopted from [26]. The dataset contains both titles and short descriptions (usually one sentence) from the AG corpus of news. The total number of categories in the dataset is 4, as shown in Table 2. The test dataset has an average of 10.09 mentions in each title and its corresponding description as depicted in Table 3.

Table 1: 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

Table 2: 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 3: Statistical analysis of the test datasets

Dataset	Total Entities	Avg-Number of Entities Per Text
AG News	76,730	10.1
Google Snippets	14,292	6.3

As it has already been mentioned, the semantic similarity between the predefined set of categories and the entities present in the given short text plays a vital role for the KBSTC method. To be able to calculate the similarity, each label/category in the datasets is mapped to its corresponding Wikipedia category, e.g. the category *Sports* from the AG news dataset is (manually) mapped to the Wikipedia category *Sports*[3]. Furthermore, as KBSTC does not depend on any training/labeled data, the training sets of AG News and Google Snippets are only used for the training of the supervised baseline methods.

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

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

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

5.2 Baselines

To demonstrate the performance of the KBSTC approach, the following supervised methods (both machine learning approaches and deep learning approaches) have been selected as baselines:

- **Bag of Words (BOW) + MLR [27]:** The bag-of-words is one of the most well-known methods to represent documents as vectors. This model is constructed by considering the 50,000 most frequent words from the training corpus and a Multinomial Logistic Regression (MLR) is adopted as a classifier.
- **n-grams + MLR [27]:** The bag-of-n-grams model is constructed by selecting the 500,000 most frequent n-grams (up to 5-grams) from the training set. This model enables the consideration of phrases such as *"Apple company"* as one unit. Treating such a phrase as two different words yields completely different vector representations. This model uses an MLR for classification as well.
- **n-grams-TF-IDF + MLR [27]:** Similar to *n-grams*, this model is also constructed by selecting the 500,000 most frequent n-grams (up to 5-grams) from the training set. The counts are considered as the term-frequency (TF) and the logarithm of the fraction of all samples divided by the number of samples with the corresponding n-gram in the training set is considered as the inverse document frequency (IDF). As a classifier again an MLR is adopted.
- **char-CNN [26]:** This approach learns the representation of words from their characters. The approach encodes word characters with a "one-hot" encoding technique. The length of each vector is 70 (26 English letters, 10 digits, 33 other characters). Subsequently, a Convolutional Neural Network (CNN) is applied for the classification process.
- **TF-IDF + SVM [23]:** In this model, a term frequency and an inverse document frequency (TF-IDF) are calculated as features for a subsequent SVM classifier.
- **Paragraph Vector + SVM [23]:** The algorithm introduced by [13] learns the representation of sentences, paragraphs, or documents as vectors. Subsequently an SVM classifier is applied.
- **LSTM [23]:** This method is based on the standard Long-Short Term Memory (LSTM) neural network. It contains a single LSTM layer followed by an average pooling and a logistic regression layer.
- **Clustering and CNN (CCNN) [23]:** In order to overcome the sparsity problem of short text, this model expands short texts based on word embedding clustering and uses a convolutional neural network (CNN) for the classification phase.

5.3 Comparison with the Supervised Methods

To assess the effectiveness of the proposed approach, the obtained experimental results are compared to the baselines presented in Section 5.2.

Table 4 and Table 5 show the accuracy of the KBSTC approach in comparison to the supervised methods on the Google Snippets dataset and the AG News dataset respectively. It should be noted that all results presented in those tables, regarding the baselines, correspond to the numbers reported in the original papers.

Based on the results of Table 4 the proposed KBSTC approach outperforms all supervised learning approaches except char-CNN. Remarkably, KBSTC achieves higher accuracy than the LSTM model which is a neural network based approach. This fact demonstrates the strength of the proposed approach. Considering the performance of the char-CNN approach, it follows a rather sophisticated process and together with the impact that the hyper parameters have, it outperforms KBSTC. It is worth noting here that suboptimal hyper parameters might led to an accuracy drop in char-CNN while KBSTC does not require any parameter tuning.

Table 4: The classification accuracy of KBSTC against supervised classifiers on Google Snippets (%)

Model	Google Snippets
CCNN	85.5
TF-IDF+SVM	62.6
Paragraph Vector+SVM	61.9
LSTM	63.0
KBSTC	68.1

Table 5 presents the results of KBSTC and the comparison against the competitors for the AG News dataset. In this case the supervised approaches all outperform KBSTC. The reason here can be attributed to the different characteristics of the two datasets. AG News is a larger dataset with more training samples (see Table 2) in comparison to Google Snippets (see Table 1). Moreover, the average length of each text in AG News is longer as shown in Table 3. Those differences might be the reason of the significant increase in accuracy (Table 5) in comparison to the Snippets dataset (Table 4) and an indicator that the size of the training set has a significant impact on the accuracy for all the supervised approaches.

However, the KBSTC approach achieves a higher accuracy score with the AG News dataset than with the Google Snippets dataset (Table 5). The reason for this difference might be found in the nature of the two datasets (Table 1 and Table 2). The AG news dataset provides only 4 different categories in comparison to the 8 categories of the Google Snippets dataset. Often a smaller number of different classes can make the classification task easier. Also, in the AG news dataset the categories are much more diverse in comparison to the Google Snippets categories. During the experiments it has been observed that semantically related or similar categories might confuse the KBSTC approach. For example, to distinguish between the category *"Engineering"* and the category *"Computer"* (from Google Snippets) was more difficult for KBSTC than to distinguish between the category *"Business"* and the category *"Sports"* (from AG news).

Overall, the achieved results have proven that for the short text classification task, the KBSTC achieves a high accuracy score without requiring any expensive labeled data, an additional time consuming training phase, or a difficult parameter tuning step. In addition, KBSTC outperforms most of the supervised methods for the Google Snippets dataset.

5.4 Using Wikipedia as a Training Set

To further demonstrate the effectiveness of the proposed KBSTC method, an additional experiment has been conducted. The results

Table 5: The classification accuracy of KBSTC against supervised classifiers on the AG News (%)

Model	AG News
BOW+MLR	88.8
n-grams+MLR	92.0
n-grams-TF-IDF+MLR	92.4
char-CNN	87.2
KBSTC	80.5

in Table 4 and Table 5 indicate that supervised methods can perform well, in case of existence of a 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 derive a training set from existing publicly available sources such as Wikipedia[4]. Wikipedia is a collaboratively edited knowledge source, which contains more than 5 million articles, where each article is associated with one or more categories. Moreover, Wikipedia contains a rich category hierarchy.

To generate a training set, for each category from two datasets (AG News and Google Snippets), training data samples have to be assembled. First, Wikipedia articles that are associated with the corresponding category (or its sub categories) are collected. In other words, if the dataset category is *Technology* then all the Wikipedia articles associated with the category *Technology* are collected. From the collected articles, 10,000 Wikipedia articles are randomly selected as a training set per category. These articles constitute training sets of AG News and Google Snippets separately. After generating the training set for each dataset, TF-IDF is calculated as a feature and two SVM classifiers are trained with the corresponding feature set. As test set we used the original AG News and Google Snippets test datasets.

Table 6: The classification accuracy of KBSTC against a traditional classifier. Trained on the Wikipedia dataset and tested on AG News and Google Snippets (%)

Method	AG News	Google Snippets
TF-IDF+SVM	59.9	53.9
KBSTC	**80.5**	**68.1**

The obtained results presented in Table 6 indicate that the KBSTC approach achieves higher accuracy in comparison to the SVM classifier. Even though the size of the training data is not too small, our proposed method outperforms the SVM classifier. In addition, exactly the same approach (TF-IDF+SVM) when both train and test phases are performed directly with the Google Snippets dataset, achieved an accuracy score of 62.6% (Table 4). While the same model when trained with Wikipedia articles and tested on Google Snippets achieved an accuracy of 53.9% (Table 6).

From these results we conclude that not only the size of the training set is important for the classification performance but also the nature and the characteristics of training and test datasets should be similar for the supervised classification approaches. This fact indicates that the classification accuracy highly depends on the nature of the chosen training dataset.

5.5 Entity-Category Embeddings

In order to demonstrate the quality of the proposed joint entity and category embedding model, it has been compared with recent work on joint embeddings of hierarchical categories and entities [16]. The study proposed two different embedding models, the Category Embedding (CE) model and the Hierarchical Category Embedding (HCE) model. The first model CE is based on the Skip-gram word embedding model [17] learning simultaneously representations of entities and their associated categories. However, to enhance the embedding quality, the authors have extended the CE model. They integrated the category hierarchy structure into the embedding space and proposed the HCE model. Both models were trained using Wikipedia. The authors applied both models to the dateless hierarchical classification task. The results suggest that HCE can capture semantic relatedness between entities and categories better than the CE model. Therefore, we consider the HCE model as a competitor in our experiments.

Table 7: The classification accuracy of the proposed embedding model against Joint Entity Category Embeddings (%)

Method	AG News	Google Snippets
KBSTC based on HCE	42.96	52.18
KBSTC based on our embeddings	**80.5**	**68.1**

The HCE model and the proposed joint entity and category model are applied to the short text classification task by using the two datasets introduced in Section 5.1. The experimental results of the accuracy scores are shown in the Table 7. The results suggest that the proposed entity-category embedding model outperforms the HCE model. The reason is that the proposed embedding model preserves both the local structure (first order proximity) as well as the global structure (second order proximity) of the network between the vertices. This indicates that preserving both first order proximity and second order proximity supports to capture better semantic representation between vertices.

5.6 Partitioning the Train Dataset

To additionally show the impact of the size of the labeled data on the text categorization task, a further experiment has been conducted. Here, four smaller training datasets from the AG News dataset have been randomly sampled. The new datasets are of size 100, 200, 1000 and 2000 texts. For each sampled datasets TF-IDF has been calculated as a feature and three different supervised classifiers (i.e. SVM, Logistic Regression (LR), and Naive Bayes (NB)) have been trained on these datasets. The complete AG news test dataset has been considered for testing. The experimental results of the accuracy scores are depicted in Figure 3. With 100 samples, SVM, NB and LR perform poorly. However, it is clear that by increasing the size of the training data the accuracy improves. The results suggest that the size of the training dataset has a huge impact on

[4] https://en.wikipedia.org/wiki/Wikipedia

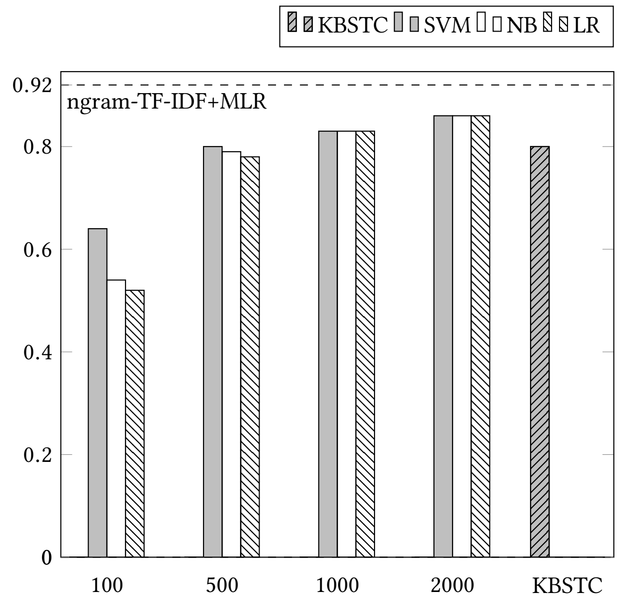

Figure 3: Performance of the supervised approaches for different training set sizes sampled from the AG News dataset. The x axis corresponds to the number of training samples, the y axis corresponds to the achieved accuracy score. The dashed line represents the best accuracy score 92.4%, achieved by n-gram-TF-IDF+MLR by using all the AG News training set.

the accuracy of the supervised methods. However, KBSTC performs solid and stable and achieves adequate accuracy.

6 CONCLUSION AND FUTURE WORK

In this paper we have proposed KBSTC, a probabilistic model for short text categorization. KBSTC is a dataless approach, meaning that it does not require any labeled training data. It considers entities present in the text and their semantic relatedness to predefined categories to classify short documents. Hence, entities and categories from a large knowledge base are embedded into a common vector space using state-of-the-art network embedding techniques. The experimental results have proven that it is possible to categorize short text in an unsupervised way with an accuracy of 80.5% for the AG News dataset.

As for ongoing work, we are about to compare our approach to the dataless text classification approach [10] designed for long text. Future works also include the extension of KBSTC towards long text classification as well as the additional inclusion of word embeddings into the common entity and category vector space. It is interesting to observe the performance of KBSTC when there are more entities in the given document, and when the disambiguation of words to entities is achieved through entity linking state-of-the-art approaches.

REFERENCES

[1] Ameni Bouaziz, Célia da Costa Pereira, Christel Dartigues-Pallez, and Frédéric Precioso. 2016. Introducing Semantics in Short Text Classification. In *Computational Linguistics and Intelligent Text Processing - 17th International Conference, CICLing 2016, Konya, Turkey, April 3-9, 2016, Revised Selected Papers, Part II.* 433–445. https://doi.org/10.1007/978-3-319-75487-1_34

[2] Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of Semantic Representation: Dataless Classification.. In *AAAI*. AAAI Press, 830–835. http://dblp.uni-trier.de/db/conf/aaai/aaai2008.html#ChangRRS08

[3] Mengen Chen, Xiaoming Jin, and Dou Shen. 2011. Short Text Classification Improved by Learning Multi-Granularity Topics. In *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011.* 1776–1781. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-298

[4] Xingyuan Chen, Yunqing Xia, Peng Jin, and John A. Carroll. 2015. Dataless Text Classification with Descriptive LDA.. In *AAAI*. AAAI Press, 2224–2231. http://dblp.uni-trier.de/db/conf/aaai/aaai2015.html#ChenXJC15

[5] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. 2016. Very Deep Convolutional Networks for Natural Language Processing. *CoRR* abs/1606.01781 (2016). arXiv:1606.01781 http://arxiv.org/abs/1606.01781

[6] Zichao Dai, Aixin Sun, and Xu-Ying Liu. 2013. Crest: Cluster-based Representation Enrichment for Short Text Classification.. In *PAKDD (2) (Lecture Notes in Computer Science)*, Vol. 7819. Springer, 256–267. http://dblp.uni-trier.de/db/conf/pakdd/pakdd2013-2.html#DaiSL13

[7] Evgeniy Gabrilovich and Shaul Markovitch. 2006. Overcoming the brittleness bottleneck using wikipedia: enhancing text categorization with encyclopedic knowledge. In *AAAI'06: proceedings of the 21st national conference on Artificial intelligence.* AAAI Press, 1301–1306.

[8] E. Gabrilovich and S. Markovitch. 2007. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. *Proceedings of the 20th International Joint Conference on Artificial Intelligence* (2007), 6–12.

[9] Edouard Grave, Tomas Mikolov, Armand Joulin, and Piotr Bojanowski. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers.* 427–431. https://aclanthology.info/papers/E17-2068/e17-2068

[10] Swapnil Hingmire, Sandeep Chougule, Girish K. Palshikar, and Sutanu Chakraborti. 2013. Document classification by topic labeling.. In *SIGIR*. ACM, 877–880. http://dblp.uni-trier.de/db/conf/sigir/sigir2013.html#HingmireCPC13

[11] Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98* (1998), 137–142.

[12] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL.* 1746–1751. http://aclweb.org/anthology/D/D14/D14-1181.pdf

[13] Quoc V Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents.. In *ICML*, Vol. 14. 1188–1196.

[14] Ji Young Lee and Franck Dernoncourt. 2016. Sequential Short-Text Classification with Recurrent and Convolutional Neural Networks. *CoRR* abs/1603.03827 (2016). http://dblp.uni-trier.de/db/journals/corr/corr1603.html#LeeD16

[15] Chenliang Li, Jian Xing, Aixin Sun, and Zongyang Ma. 2016. Effective Document Labeling with Very Few Seed Words: A Topic Model Approach.. In *CIKM*. ACM, 85–94. http://dblp.uni-trier.de/db/conf/cikm/cikm2016.html#LiXSM16

[16] Yuezhang Li, Ronghuo Zheng, Tian Tian, Zhiting Hu, Rahul Iyer, and Katia P. Sycara. 2016. Joint Embedding of Hierarchical Categories and Entities for Concept Categorization and Dataless Classification. *CoRR* abs/1607.07956 (2016). http://dblp.uni-trier.de/db/journals/corr/corr1607.html#LiZTHIS16a

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality.. In *NIPS*. 3111–3119. http://dblp.uni-trier.de/db/conf/nips/nips2013.html#MikolovSCCD13

[18] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom M. Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning* 39, 2/3 (2000), 103–134.

[19] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. 2008. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *WWW '08: Proceeding of the 17th international conference on World Wide Web.* ACM, New York, NY, USA, 91–100. https://doi.org/10.1145/1367497.1367510

[20] Yangqiu Song and Dan Roth. 2014. On Dataless Hierarchical Text Classification.. In *AAAI*. AAAI Press, 1579–1585. http://dblp.uni-trier.de/db/conf/aaai/aaai2014.html#SongR14

[21] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. *CoRR* abs/1503.03578 (2015). http://dblp.uni-trier.de/db/journals/corr/corr1503.html#TangQWZYM15

[22] Chenguang Wang, Yangqiu Song, Haoran Li, Ming Zhang, and Jiawei Han. 2016. Text Classification with Heterogeneous Information Network Kernels. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*. 2130–2136. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12392

[23] Peng Wang, Bo Xu, Jiaming Xu, Guanhua Tian, Cheng-Lin Liu, and Hongwei Hao. 2016. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. *Neurocomputing* 174 (2016), 806–814. http://dblp.uni-trier.de/db/journals/ijon/ijon174.html#WangXXTLH16

[24] Yijun Xiao and Kyunghyun Cho. 2016. Efficient Character-level Document Classification by Combining Convolution and Recurrent Layers. *CoRR* abs/1602.00367 (2016). http://dblp.uni-trier.de/db/journals/corr/corr1602.html#XiaoC16

[25] Jifeng Xuan, He Jiang, Zhilei Ren, Jun Yan, and Zhongxuan Luo. 2017. Automatic Bug Triage using Semi-Supervised Text Classification. *CoRR* abs/1704.04769 (2017). http://dblp.uni-trier.de/db/journals/corr/corr1704.html#XuanJRYL17

[26] Xiang Zhang and Yann LeCun. 2015. Text Understanding from Scratch. *CoRR* abs/1502.01710 (2015). arXiv:1502.01710 http://arxiv.org/abs/1502.01710

[27] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *NIPS*.