applied sciences

MDPI

Article

Cognitive Aspects-Based Short Text Representation with Named Entity, Concept and Knowledge

Wenfeng Hou *, **Qing Liu and Longbing Cao**

Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia; qing.liu-7@student.uts.edu.au (Q.L.); longbing.cao@uts.edu.au (L.C.)

* Correspondence: wenfeng.hou@student.uts.edu.au

Received: 19 June 2020; Accepted: 13 July 2020; Published: 16 July 2020

Abstract: Short text is widely seen in applications including Internet of Things (IoT). The appropriate representation and classification of short text could be severely disrupted by the sparsity and shortness of short text. One important solution is to enrich short text representation by involving cognitive aspects of text, including semantic concept, knowledge, and category. In this paper, we propose a named Entity-based Concept Knowledge-Aware (ECKA) representation model which incorporates semantic information into short text representation. ECKA is a multi-level short text semantic representation model, which extracts the semantic features from the word, entity, concept and knowledge levels by CNN, respectively. Since word, entity, concept and knowledge entity in the same short text have different cognitive informativeness for short text classification, attention networks are formed to capture these category-related attentive representations from the multi-level textual features, respectively. The final multi-level semantic representations are formed by concatenating all of these individual-level representations, which are used for text classification. Experiments on three tasks demonstrate our method significantly outperforms the state-of-the-art methods.

Keywords: short text representation; semantic representation; short text classification; knowledge graph; convolutional neural network; attention network

1. Introduction

With the development of Internet of Things (IoT) [1], various information can be found online and IoT networks in the form of short text, such as short descriptions, social media, news description, product review, and instant messages, and so forth. Unlike long-textual documents, one piece of short text only contains few sentences or even just a few words. For example, Twitter limits its tweet length to 280 characters. Sparsity and shortness are the two intrinsic characteristics of such short text. Lacking enough word co-occurrences and shared context, it is difficult to extract representative and informative features from short text. Therefore, document representation and word embedding methods, which heavily rely on the word frequency or shared context, may not capture sufficient information from short text in IoT networks and perform well in downstream tasks such as short text classification.

The semantic enhancement of short text representation is a common way to address the problems aforementioned. To implement the semantic enhancement, external knowledge bases like DBpedia and Microsoft Concept Graph are usually adopted as a complement for short text semantic enhancement. There are several reasons why external knowledge bases are chosen. First, mining the entity relationships from the knowledge base can enhance the short text semantic representation. As demonstrated in Figure 1, in the knowledge graph, *Cristiano Ronaldo* and *Lionel Messi* have a lot in common—both of them won the *Ballon d'Or*, *UEFA Champions League* and *La Liga*; they share the same career as a *football player*, and so forth. These common entities in the knowledge graph are highly correlated with the same category *Sport*. With the extra entity relationships from the knowledge graph, short text representation can be enhanced. Second, the entity level representation can help to disambiguate terms which have the same spelling. For example, both sentences "*WHO has named the disease COVID-19, short for Corona Virus Disease 19.*" and "*Corona is the best beer I have ever drunk.*" have the same term *Corona*. According to our common sense, the first one refers to *Coronavirus*, and the named entity is *Corona_Virus*; and the second one stands for the famous beer brand *Corona*, and its entity is *Corona_beer*. Hence, at the entity level, we can obtain more precise representation instead of the same word embedding at the word level. Third, the concept level representation is more abstract compared with both word and entity levels of representations. Hence, the concept representation can enhance short text semantic representation. A concept can be regarded as a set or class of entities or "things" within a domain [2]. It is a higher perspective of description of a "thing". Those higher perspective descriptions can strengthen the semantic representation. For instance, giving a piece of news "*Dunga will attend the award ceremony*", according to the keywords *Dunga* and *ceremony*, it would be difficult to identify which category this piece of news belongs to, as the meaning of the keyword *Dunga* is not clear here. If the news title changes to "*Brazilian football star will attend the award ceremony*", it is easy to point out that this is a sport news. *Dunga* was the captain of Brazilian football team which won the 2002 FIFA world cup, and the "*Brazilian football star*" is the concept of term *Dunga*. This example show that it would be easier to determine the category of short text by involve word-related concepts. Accordingly, we believe that the concept level representation is a significant supplement for short text representation based on keyword and entity.

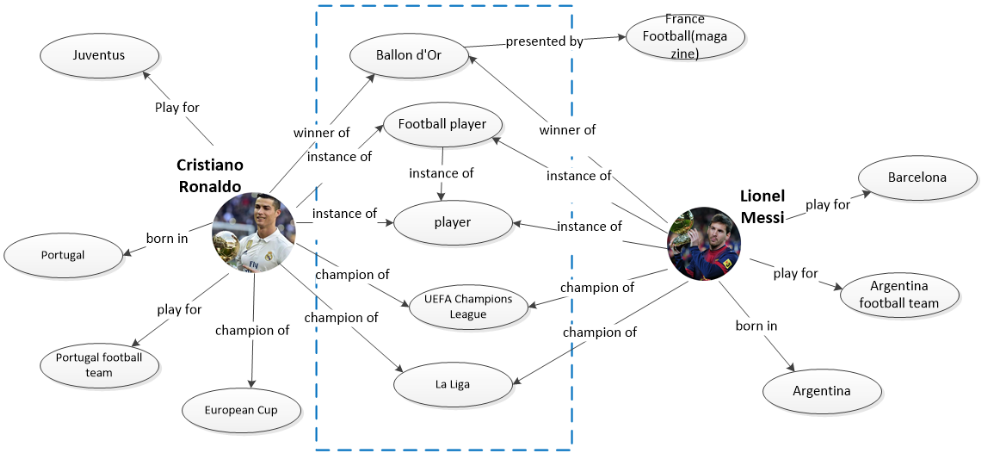


Figure 1. Illustration of C.Ronaldo and Messi connected through knowledge entities.

Owing to the convenience of integrating extra knowledge into neural networks, deep learning-based short text representation forms the common method for short text classification. Among a majority of neural network types, Kim [3] first introduced CNN to the text classification. CNN is good at extracting local features through the convolution layer. To capture the informative information from the text, Vaswani et al. [4] proposed an attention network in NLP. The improvement of combining knowledge bases for downstream deep short text classification tasks has been verified in recent research [5–7]. Although such methods gain more accurate short text representations, limitations exist such as on the way of combining extra knowledge bases, that is, they still suffer from making full use of external knowledge bases. They consider only one aspect (only the entity or concept information) from knowledge bases to enrich the short text representation.

In this paper, we involve multiple cognitive aspects [8–10] of short text including concept, knowledge and category into short text representation, and propose a multi-level Entity-based Concept Knowledge-Aware (ECKA) representation model for enhancing short text semantic representations. We first extract the named entities from short text, and then retrieve the corresponding concepts and knowledge graph entities through Microsoft Concept Graph and DBpedia, respectively. Short text representation learned from ECKA is very informative since it is the combination of four-level representations, that is, from word, entity, concept to knowledge levels. Specifically, the word-level representation refers to the pretrained word embedding. The entity-level representation represents the identified named-entity embedding. The knowledge-level representation, which is learned and transformed from a knowledge graph, stands for the external knowledge correlation. The concept-level representation refers to a higher perspective of descriptive embedding. Secondly, we apply CNN to extract the local features on different levels, respectively. Lastly, since different items (i.e., words, entities, concepts and knowledge) in one short text contribute differently to the downstream short text classification, the category of short text may be determined by the category-related words. For example, in the aforementioned sentence "*Brazilian football star will attend the award ceremony*", *football* is the category-related word for 'Sport'. Similarly, the category of short text may be determined by the category-related features. Therefore, we further apply the attention network to learn the category-sensitive weights of each item set in the four-level representation, respectively.

The main contributions of this paper are summarized as follows:

- We propose a novel multi-level model to learn the short text representation from different aspects respectively. To capture more semantic information, We use the named entity-based approach to obtain the external knowledge information—entity, concept, and knowledge graph. Such external knowledge information is utilized to enrich the short text semantic representation.
- To capture the category-related informative representation in terms of multi-level features, we build a joint model by using CNN-based Attention network to capture their respective attentive representations, and then the embeddings learned from different aspects are concatenated for the short text representation.
- We conduct extensive experiments on three datasets for short text classification. The results show that our model outperforms the state-of-the-art methods.

The rest of this paper is organized as follows—Section 2 summarizes a brief review of the related work; Section 3 presents the details of the proposed method; Section 4 presents the experiments and analysis; lastly, Section 5 concludes the paper and outlines the future work.

2. Related Work

Short text classification is an important task of NLP . Many traditional methods like BoW, SVM and KNN, and so forth, have been explored for this task. In recent years, deep neural networks have been increasingly employed in the short text analysis. For example, Kim [3] first introduced the Convolutional Neural Network (CNN for short) to the text classification. CNN is used to extract local and position-invariant features. Recurrent Neural Network (RNN for short) is another approach for the text processing. Unlike CNN, RNN is good at processing long range semantic dependency rather than local key-phrases. Yang [11] proposed an attention model to process the problem of different words in a document with informative difference.

The deep models aforementioned are flexible to some extent in the short text classification. However, due to the shortness and sparsity of short text, it is quite difficult for them to capture enough semantic information with limited words in the text content. From this perspective, how to enrich the short text semantic information with extra knowledge or common sense borrowed from other sources becomes a hot topic in this area. Concept is an aspect which is extensively used for text semantic enhancement. The Microsoft Concept Graph is a big graph of concepts, researchers have utilized it for the semantic enhancement. Wang et al. [2] proposed a 'Bag-of-Concept' (instead of word) approach for the short text representation and constructs a concept model for each category, they then conceptualize the short text to a set of relevant concepts. Wang et al. [7] proposed a deep convolutional neural network model, which utilizes the concept, word and character for short text classification. To measure the importance of each concept from the concept set, Chen et al. [6] proposed a knowledge powered multiple attention networks for text classification, it applies two attention mechanisms to measure the importance of each concept from two aspects: the concept towards short text attention and the concept towards concept set attention.

In addition, knowledge graph is another effective way to enhance the text semantic representation. A typical knowledge graph describes the structured and unstructured information with a Resource Description Framework (RDF). Information in the knowledge base is stored in the form of entity-relation-entity triples. There are many knowledge graph—DBpedia [12], Wikidata [13], Freebase [14] and YAGO [15]. They are widely employed in recent research on semantic enhancement for short text. Wang et al. [16] devised a multi-channel CNN by fusing the word and knowledge graph levels of representations for news text representation. Gao et al. [17] proposed a word and knowledge level-based self-attention mechanism for the text semantic enhancement.

For further semantic enhancement, entity is usually utilized together with the knowledge base. Flisar et.al [5] proposed an entity-based text classification, it utilizes entity and its related attributes for the short text enhancement. T̈urker [18] proposed a knowledge-based short text categorization, which utilizes the external knowledge base (Wikipedia) and entity.

3. The ECKA Method

The framework of our proposed ECKA representation is illustrated in Figures 2 and 3 further shows its semantic information retrieval module. We introduce the architecture of ECKA from bottom up. Our model consists of three modules—the semantic information retrieval module, the feature extraction module, and the attention module. The semantic information retrieval module as illustrated in Figure 3, retrieves the entity, concept, and knowledge graph from an external knowledge base. The feature extraction module and the attention module are illustrated in Figure 2. The feature extraction module implemented by CNN is used to extract the local and position-invariant features from multiple sources. The attention module is used to capture category-related informative representation from multi-level features respectively. Taking a short text as input, our model first extracts all the entities implicated in the short text by using the DBpedia Spotlight and then retrieves the relevant concepts and knowledge graph entities through the Microsoft Concept Graph and DBpedia, respectively. TransE is employed to get the knowledge graph embedding. We also utilize CNN with an attention network to capture category-related informative representation from multi-level features respectively. Finally, these multi-level semantic text representation is concatenated and fed into a fully-connected layer to get the category probability distribution. We describe the detail as follows.

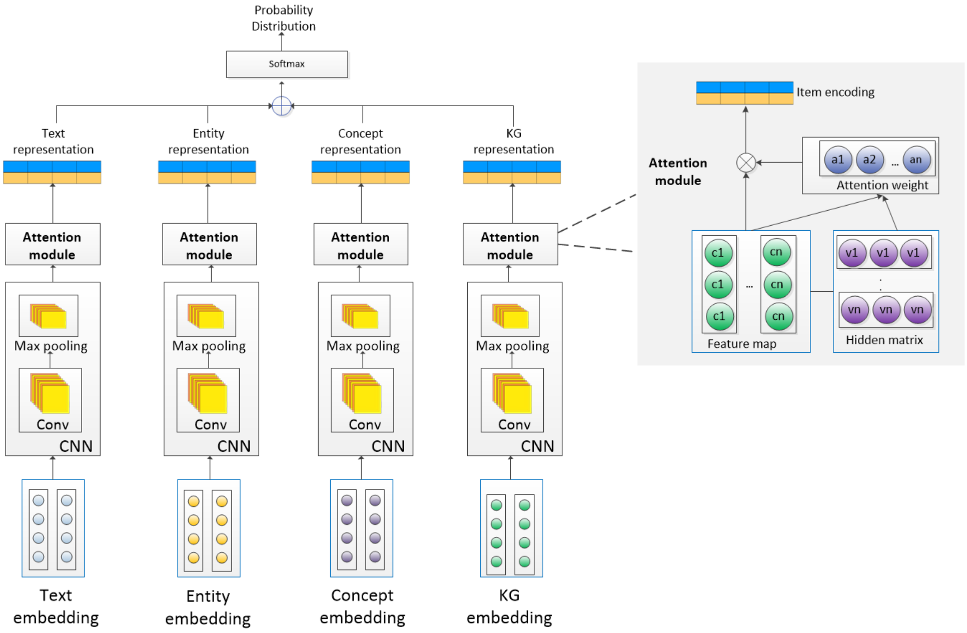


Figure 2. The framework of Entity-based Concept Knowledge-Aware (ECKA).

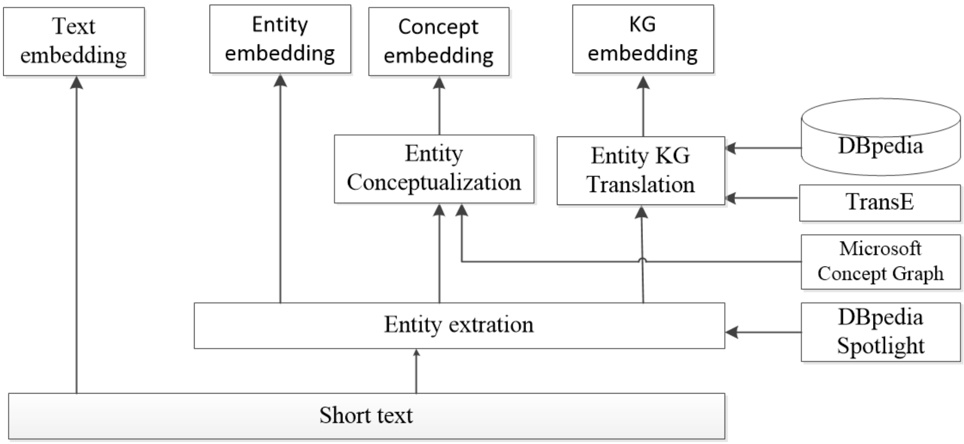


Figure 3. The semantic information retrieval module.

3.1. *Semantic Information Retrieval Module*

The goal of this module is to retrieve the relevant entities, concepts, knowledge graphs from the short text. Firstly, we extract the entities from short text. Entity annotation and linkage are the foundation for our model. Some recently proposed annotation and linking tools, such as the DBpedia Spotlight, TagMe, and wikify!, can satisfy our need here. In this work, we choose DBpedia as our knowledge base and DBpedia Spotlight as our annotation tool. With the DBpedia Spotlight, we can link the extracted named entities in the input short text to the DBpedia resources [19]. Secondly, we obtain relevant concept for the extracted entities. ConceptNet [20] and Microsoft Concept Graph [21–27] are the two widely used toolkits to obtain the concept of an object. We choose to use the Microsoft Concept Graph, which has 5.3 million concepts learned from billions of website pages and search logs for the conceptualization. Finally, the knowledge graph for the relevant entities can be obtained through DBpedia. A typical knowledge graph is a collection of relationship triples (h,r,t) in which h represents head, r represents relation and t represents tail. The structural knowledge graph information needs to be transformed to the embedding. There are many transform methods that can learn the low-dimensional vector spaces from the knowledge graph. The comparison of some widely-used methods, like TransE, TransD, TransH and TransR, can be found in Reference [28]. In our model, we choose to use TransE as the knowledge graph embedding method.

3.2. *Feature Extraction Module*

This module utilizes the word, entity, concept and knowledge graph to generate multi-level semantic short text representations. There are three components in this module: the input layer, the embedding layer, and the representation layer. The input layer demonstrates how to get the different sources from the external knowledge bases. The embedding layer shows how to get the embedding for the input layer and how to translate the different embeddings to the same vector space. The representation shows how to extract the higher level features from the embedding layer. The details of each layer are shown as follows.

3.2.1. The Input Layer

The input of each short text in our model consists of four-level sets which are obtained from different sources, where each set is defined as follows:

- The Word set: The word set contains all the words in each short text. W = $\{w_1, w_2, w_3, ...w_n\}$.
- The Entity Set: E = $\{e_1, e_2, e_3, ..e_n\}$, e_n represents the entities extracted from short text through the DBpedia Spotlight.
- The Concept Set: The concept of each entity is retrieved from the Microsoft Concept Graph, and the concept set can be represented as C = $\{c_1, c_2, c_3, ..c_n\}$.

- The Knowledge set: This set is denoted as KE = $\{ke_1, ke_2, ke_3 \ldots ke_n\}$, it is the same as the entity set, but its representation is learned from different aspects respectively.

3.2.2. The Embedding Layer

Each short text consists of a word level, an entity level, a concept level, and a knowledge level set. The semantic information retrieval process is demonstrated in Figure 3. We use the pretrained Google word2vec embedding to obtain the embeddings for the first three sets, which can be represented as W_e= $\{w_{1e}\ w_{2e}\ w_{3e} \ldots w_{ne}\}$, E_e=$\{e_{1e}\ e_{2e}\ e_{3e} \ldots e_{ne}\}$ and C_e=$\{c_{1e}\ c_{2e}\ c_{3e} \ldots c_{ne}\}$, n is the entity number in the short text. The knowledge entity embedding is learned by the following steps. First, the related entity of each knowledge entity is retrieved from the DBpedia, then the knowledge transforming method TransE is applied to learn the knowledge graph embedding. Finally, as the word, entity and concept embeddings with 300 dimensions are learned by word2vec and the knowledge graph embedding with 50 dimensions is learned from TransE, the two embeddings need to be transformed to the same vector space. The transformed knowledge entity embedding can be represented as:

$$t(ke_1 ... n) = [t(ke_1)\ t(ke_2)\ ... t(ke_n)]. \tag{1}$$

In our model, we use a nonlinear function to transform the knowledge entity embedding:

$$t(ke) = tanh(Mke + b), \tag{2}$$

where $M \in \mathbb{R}^{d \times k}$ represents the trainable transformed matrix, and $b \in \mathbb{R}^{d \times 1}$ stands for the trainable bias. By using this function, the knowledge entity embedding can be mapped to the word2vec embedding vector space.

3.2.3. The Representation Layer

CNN is a typical model to extract the local-level features from the embedding matrix. We apply CNN to generate the feature map. For the entity embedding matrix $E_e = [e_{1e}, e_{2e}, e_{3e}, \ldots e_{ne}]$, firstly, a convolution operation with the filter $w \in \mathbb{R}^{dh}$, where d is the dimension of the embedding and $h(h \leq n)$ represents the filter window size, is applied on the embedding matrix to generate a new future C_i:

$$C_i = f(W_c \cdot X_{i:i+h-1} + b_c), \tag{3}$$

where h stands for the filter window size, $i : i + h - 1$ represents the convolution starting from the i^{th} entity and ending at $(i + h - 1)^{th}$. $X_{i:i+h-1}$ represents the concatenation embedding and f is the nonlinear function, here we use $Relu$. b_c is the bias.

Filtering is applied in all possible windows, then a feature map is generated:

$$C_e = [c_{e1}, c_{e2}, c_{e3} \ldots c_{en-h+1}]. \tag{4}$$

Similarly, the feature map for the word, concept, knowledge entity sets can be represented as:

$$C_w = [c_{w1}, c_{w2}, c_{w3} \ldots c_{wn-h+1}] \tag{5}$$

$$C_c = [c_{c1}, c_{c2}, c_{c3} \ldots c_{cn-h+1}] \tag{6}$$

$$C_k = [c_{k1}, c_{k2}, c_{k3} \ldots c_{kn-h+1}], \tag{7}$$

where n represents the entity number and h stands for the window size.

3.3. The Attention Module

Not all items (words, entities, concepts, and knowledge) contribute equally to the representation of short text. The category of a short text may be determined by the category-related words. Similarly, the classification result may be determined by the category-related features. Hence, we apply the attention network on the feature map generated in the representation layer to obtain the attentive short text representation for each level. The feature C_i generated by the convolution layer is fed into a one-layer MLP to v_i, which can be treated as a hidden representation of C_i:

$$vi = tanh(W_c C_i + b_c) \tag{8}$$

where W_c is a weight matrix and b_c is the bias, then the weight β is calculated through a softmax function as follows:

$$\beta_i = softmax(W_\beta vi) \tag{9}$$

where w_β is a weight vector. Then, the entity representation can be calculated as follows:

$$C_\beta = \sum_{i=1}^{h} \beta_i C_i. \tag{10}$$

As there are multiple window sizes of the filter, there are multiple feature maps. A maxpooling function is applied over each feature map C to get the final pooling vector:

$$C_\beta = argmax(C_{\beta n}), \tag{11}$$

where n is the length of the convolution window. So far, the representations for the words, entities, concepts and knowledge can be represented as: $R_w = argmax(C_{\beta wn})$, $R_e = argmax(C_{\beta en})$, $R_c = argmax(C_{\beta cn})$ and $R_k = argmax(C_{\beta kn})$.

We concatenate all these different-level representations to get the final short text representation R as follows:

$$R = [R_w \oplus R_e \oplus R_c \oplus R_k]. \tag{12}$$

Finally, the short text representation R is fed into the fully-connected softmax layer to get the category probability distribution.

4. Experiments

Our experiment is implemented in Python Keras and on three widely used datasets. The computing infrastructure setting is listed as follows—(1) Operating system: Red Hat Enterprise Linux 7.7; (2) CPU: 8 core Intel(R) Xeon(R) CPU E5-2687W v2 @ 3.40 GHz; and (3) Memory: 32 GB. We demonstrate the evaluation from two aspects: the accuracy of short text classification result; and the variants of our model—how the semantic enhancement from different levels (word, entity, concept, and knowledge graph) affect the performance of our model. The performance is compared with various classical and the state-of-the-art text classification methods.

4.1. Datasets

The details of the three datasets are listed below.

Google Snippet—This dataset is adopted from Pan [29], snippet refers to the description portion of a Google search listing, the Google search snippet with eight classes contains 10,060 training and 2180 testing samples. The average length of this data set is 12, and the detail of each category is shown in Table 1.

Twitter—This dataset is a publicly available dataset collected from Github (https://github.com/vinaykola/twitter-topic-classifier). There are two categories—sport and politics in the data. It contains 4567 training samples and 1958 testing samples, and the detail of each category is demonstrated in Table 2.

AG news—This dataset contains four category news, each category contains 30,000 training samples and 1900 testing samples. Each document contains both title and short description.

In our experiment, we only use the title as it can better illustrate the ability of ECKA on short text classification. The detail is shown in Table 3.

Table 1. The Google Snippet data set.

Category	Training	Testing	Features	Value
Business	1200	300	Avg.Len per document	12
Computers	1200	300	Total entity	1,494,181
Cul-Arts-Ent.	1880	330	Total document	12,240
Edu-Sci	2360	300		
Engineering	220	150		
Health	880	300		
Politics-Society	1200	300		
Sports	1120	200		
Total	10,060	2180		

Table 2. The Twitter data set.

Category	Training	Testing	Features	Values
Politics	2241	959	Avg.Len per document	18
Sports	2326	999	Total entity	1,586,965
			Total document	6,525
Total	4567	1958		

Table 3. The AG news data set.

Category	Training	Testing	Features	Value
World	30,000	1900	Avg.Len per document	7
Sport	30,000	1900	Total entity	2,270,042
Business	30,000	1900	Total document	127,600
Sci/Tec	30,000	1900		
Total	120,000	7600		

4.2. Data Preprocessing

A typical data preprocessing pipeline is applied to get the word level representation—

- Tokenization—Tokenization means splitting text into minimal meaningful units. In our model, the short text will be split into single words.
- Stemming—We use the NLTK's PorterStemmer for the word's stemming.
- Stop words removal—Stop words are common but meaningless words. Stop words removal is done by the NLTK stopwords collection.

4.3. Baselines

To measure the improvement of our model, we compare it with multiple traditional and state-of-the-art methods below.

BoW+TFIDF—BoW is a traditional text representation method widely used in natural language processing, the terms in the text can be regarded as a bag of word, and the term frequency in the dataset is used as weight. In this method, we use TF-IDF instead of term frequency as weight.

CNN—CNN is a classical neural network model for classification task. Kim [3] first introduced CNN to text classification. Only a word embedding layer is used in this network and we use the same parameter settings as our proposed model.

LSTM—LSTM [30] is a variant of Recurrent Neural Network. LSTM can capture the long-term dependency among words in short texts. Only a word embedding layer is used in this network.

Bi-LSTM—It is a bidirectional LSTM [31], which learns bidirectional long-term dependencies between time steps of sequence data. Only a word embedding layer is used in this network.

GRU—Gate Recurrent Unit (GRU) [32] is similar with LSTM but has fewer parameters than LSTM. Only a word embedding layer is used in this network.

Attention—Attention [4] is a mechanism widely used in NLP. Here, we use self-attention in our experiment. Only a word embedding layer is used in this network.

KBSTC—This method is proposed by T̈urker et al [18], and it utilizes the entity and knowledge base (Wikipedia) for the short text classification.

WCCNN—This method is proposed by Wang et al [7]. It utilizes word embedding and concept embedding for the short text classification. We re-implement their code for evaluation on the Twitter and Google snippet data sets.

4.4. Parameter Setting

We use the Google pretrained 300-dimension word2vec as the word embedding. The knowledge graph embedding trained by TransE has a dimension of 50. For the Twitter dataset, the kernel window size of convolutional layer is [2–4]. For Google snippet and AG news, their kernel window size is changed to [2–6]. The mini-batch size is 64, and the epoch is 10. For the Google snippet and AG news datasets, we use their standard training and validation datasets. For the Twitter dataset, we split it manually with 70% for training and 30% for testing. The 10 folder validation is employed on it to obtain the result.

4.5. Result Analysis

Experimental results are shown in Table 4. We also test the variants of our model and the result is demonstrated in Table 5. It can be seen from the results that our model significantly outperforms the state-of-the-art methods.

Table 4. Text classification comparison of different models.

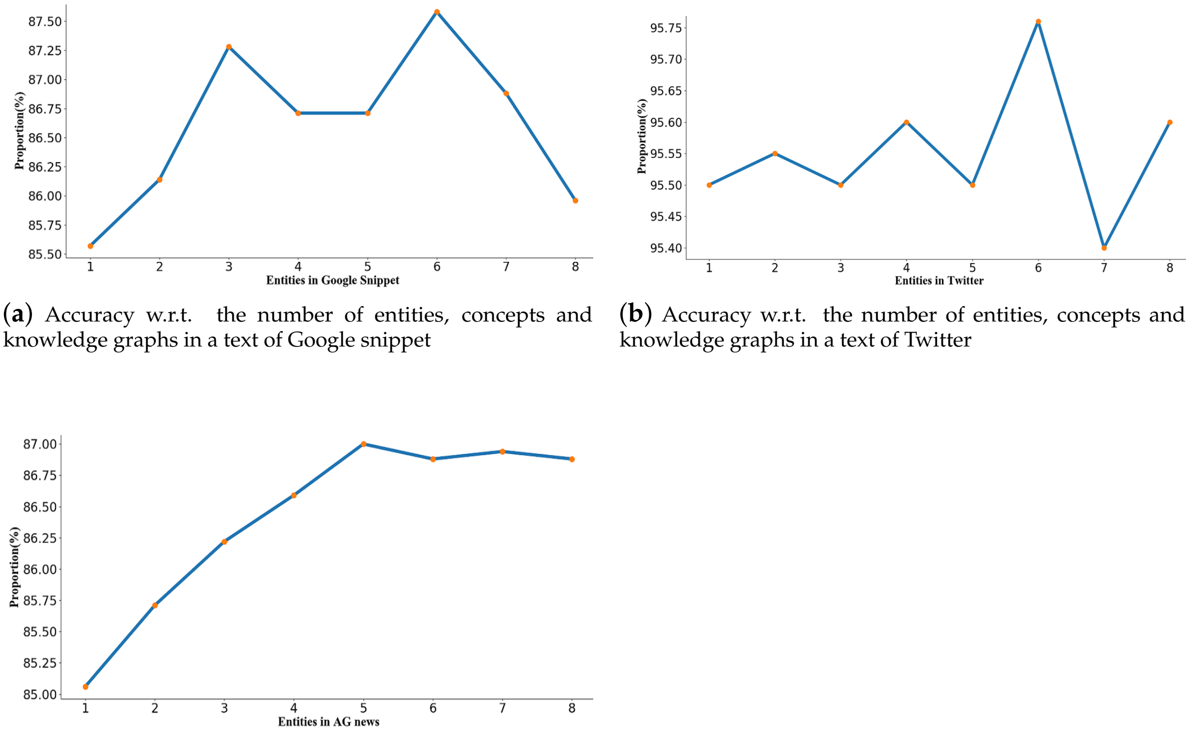
	Twitter	Google Snippet	AG News
BoW+TFIDF	94.25	61.84	72.7
CNN	95.14	85.21	83.97
LSTM	94.99	81.54	83.18
Bi-LSTM	95.10	84.86	83.5
GRU	94.99	80.92	82.98
Attention	94.43	84.73	83.34
KBSTC	-	72	67.9
WCCNN	95.09	85.83	85.57
ECKA(proposed)	95.76	87.59	86.93

Table 5. The text classification comparison of ECKA variants.

Variants	Twitter	Google Snippet	Ag News
ECKA with word only	95.19	83.70	84.13
ECKA with word and entity	95.60	86.4	86.75
ECKA with word and concept	95.50	86.28	85.15
ECKA with word and KG	95.65	86.62	84.22
ECKA with word and entity, concept and KG	95.76	87.59	86.93

4.5.1. Multiple Sources vs. Single Source

CNN uses the single source of words as its input, multiple kernels with different sizes are employed on it. From the result in Table 4, we can see CNN performs best among the baseline methods which only use a single source. This is because different window sizes of convolution operation is employed to extract the local features which can enhance the text representation. Our model performs better than all the baseline models. The reasons that our model achieves better result than the others are listed as follows—(i) The model handles the ambiguous term by using named entity technique. Base on the named entity, the model can get more precise representation on the entity, concept and knowledge levels. (ii) We enrich the short text representation from different sources. The model learns the superordinate representation through concept. And the latent semantic representation is obtained through knowledge entity and its linked entities within the knowledge graph. (iii) We use the CNN to extract the local features and the attention network to capture the attentive representation from multi-level features respectively, which better captures category-related informative features for short text classification.



(**a**) Accuracy w.r.t. the number of entities, concepts and knowledge graphs in a text of Google snippet

(**b**) Accuracy w.r.t. the number of entities, concepts and knowledge graphs in a text of Twitter

(**c**) Accuracy w.r.t. the number of entities, concepts and knowledge graphs in a text of AG news

Figure 4. The distributions of accuracy.

4.5.2. Comparison of ECKA Variants on Multiple Sources

In this section, we compare the variants of ECKA in terms of involving external knowledge to demonstrate the effectiveness of our model design. The results are listed in Table 5, which concludes that—(1) In comparison with the baseline which only uses words, the semantic enhancement by using entity, concept and knowledge graph respectively can boost the performance for short text classification. This result proves that involving external knowledge can enhance the semantic representation. (2) Compared to the two-source model, the model with four sources performs better, which proves that the use of multiple sources from different aspects is another effective way to improve the short text classification.

4.6. Parameter Sensitivity

In this section, we investigate how different numbers of entities affect the performance of our model. We use the different numbers of entities in the set [1–8] on the three datasets. The result is demonstrated in Figure 4. The results show that the best performance is associated with six entities in Google Snippet and Twitter but five entities in AG news. However, the performance does not increase when the entity number further increases. This may be because, when the majority of entities are involved, our model learns the informative representation from the entities. The learned informative entities benefit the classification result.

5. Conclusions

IoT networks involve increasing short text, which cannot be handled by document representation and classic NLP tools. This work involves multiple cognitive aspects of text from entity to concept and knowledge, and proposes a novel multi-level entity-based concept knowledge-aware model ECKA to enhance the short text semantic representation. ECKA learns the semantic information of short text from four different levels: the word level, the entity level, the concept level, and the knowledge level. CNN is used to extract the semantic features from different levels respectively. To capture the category-related attentive representations from these multi-level features, attention network is employed on different levels respectively. Experiments on short text classification demonstrate the effectiveness and merits of ECKA compared with traditional and state-of-the-art baseline methods.

The improvement made by ECKA is attributed to the entity identification and knowledge extraction. To further promote ECKA, we will focus on how to improve the accuracy of entity extraction and employ knowledge-enabled language representation model (e.g., K-BERT) for the short text representation. We'll explore ECKA to the data and tasks of IoT-specific systems.

Author Contributions: Conceptualization: W.H., Q.L. and L.C.; investigation: W.H.; methodology: W.H.; software: W.H.; supervision: L.C.; validation: W.H.;writing-original draft: W.H.; writing-review and editing: W.H., Q.L. and L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partly sponsored by the Australian Research Council Discovery grant DP190101079.

Acknowledgments: The authors would like to thank the editors and the reviewers for their comments on an earlier draft of this article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BiLSTM	Bidirectional Long Short Term Memory
BoW	Bag of Words
CNN	Convolutional Neural Networks
ECKA	Entity-based Concept Knowledge-Aware model
GRU	Gated Recurrent Unit
KG	Knowledge Graph
KNN	K Nearest Neighbor
LSTM	Long Short Term Memory
NLP	Natural Language Processing
RNN	Recurrent Neural Networks
SVM	Support Vector Machine

References

1. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [CrossRef]
2. Wang, F.; Wang, Z.; Li, Z.; Wen, J.R. Concept-based short text classification and ranking. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 1069–1078.
3. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
4. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.

5. Flisar, J.; Podgorelec, V. Document enrichment using DBPedia ontology for short text classification. In Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, Novi Sad, Serbia, 25–27 June 2018; pp. 1–9.
6. Chen, J.; Hu, Y.; Liu, J.; Xiao, Y.; Jiang, H. Deep short text classification with knowledge powered attention. *Proc. AAAI Conf. Artif. Intell.* **2019**, *33*, 6252–6259. [CrossRef]
7. Wang, J.; Wang, Z.; Zhang, D.; Yan, J. Combining Knowledge with Deep Convolutional Neural Networks for Short Text Classification. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 2915–2921.
8. Cao, L. *Metasynthetic Computing and Engineering of Complex Systems*; Advanced Information and Knowledge Processing; Springer: London, UK, 2015.
9. Hu, L.; Jian, S.; Cao, L.; Chen, Q. Interpretable Recommendation via Attraction Modeling: Learning Multilevel Attractiveness over Multimodal Movie Contents. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 3400–3406.
10. Cao, L. *Data Science Thinking: The Next Scientific, Technological and Economic Revolution*; Data Analytics; Springer International Publishing: Cham, Switzerland, 2018.
11. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. Hierarchical attention networks for document classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
12. Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P.N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; et al. DBpedia—A large-scale, multilingual knowledge base extracted from Wikipedia. *Semant. Web* **2015**, *6*, 167–195. [CrossRef]
13. Vrandečić, D.; Krötzsch, M. Wikidata: A free collaborative knowledgebase. *Commun. ACM* **2014**, *57*, 78–85. [CrossRef]
14. Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, Vancouver, BC, Canada, 10–12 June 2008; pp. 1247–1250.
15. Fabian, M.S.; Gjergji, K.; Gerhard, W.E.I.K.U.M. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In Proceedings of the 16th International World Wide Web Conference, Banff, AB, Canada, 8–12 May 2007; pp. 697–706.
16. Wang, H.; Zhang, F.; Xie, X.; Guo, M. DKN: Deep knowledge-aware network for news recommendation. In Proceedings of the 2018 World Wide Web Conference, Lyon, France, 23–27 April 2018; pp. 1835–1844.
17. Gao, J.; Xin, X.; Liu, J.; Wang, R.; Lu, J.; Li, B.; Fan, X.; Guo, P. Fine-Grained Deep Knowledge-Aware Network for News Recommendation with Self-Attention. In Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Santiago, Chile, 3–6 December 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 81–88.
18. Türker, R. Knowledge-Based Dataless Text Categorization. In Proceedings of the European Semantic Web Conference, Auckland, New Zealand, 26–30 October 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 231–241.
19. Daiber, J.; Jakob, M.; Hokamp, C.; Mendes, P.N. Improving efficiency and accuracy in multilingual entity extraction. In Proceedings of the 9th International Conference on Semantic Systems, Graz, Austria, 4–6 September 2013; pp. 121–124.
20. Liu, H.; Singh, P. ConceptNet—a practical commonsense reasoning tool-kit. *BT Technol. J.* **2004**, *22*, 211–226. [CrossRef]
21. Ji, L.; Wang, Y.; Shi, B.; Zhang, D.; Wang, Z.; Yan, J. Microsoft concept graph: Mining semantic concepts for short text understanding. *Data Intell.* **2019**, *1*, 238–270. [CrossRef]
22. Wang, Z.; Wang, H. Understanding Short Texts. 2016. Available online: https://www.microsoft.com/en-us/research/publication/understanding-short-texts/ (accessed on 15 October 2019).
23. Wang, Z.; Wang, H.; Wen, J.R.; Xiao, Y. An inference approach to basic level of categorization. In Proceedings of the 24th Acm International on Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2019; pp. 653–662.

24. Wang, Z.; Zhao, K.; Wang, H.; Meng, X.; Wen, J.R. Query understanding through knowledge-based conceptualization. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
25. Hua, W.; Wang, Z.; Wang, H.; Zheng, K.; Zhou, X. Short text understanding through lexical-semantic analysis. In Proceedings of the 2015 IEEE 31st International Conference on Data Engineering, Seoul, Korea, 13–17 April 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 495–506.
26. Wang, Z.; Wang, H.; Hu, Z. Head, modifier, and constraint detection in short texts. In Proceedings of the 2014 IEEE 30th International Conference on Data Engineering, Chicago, IL, USA, 31 March–4 April 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 280–291.
27. Song, Y.; Wang, H.; Wang, Z.; Li, H.; Chen, W. Short text conceptualization using a probabilistic knowledgebase. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011.
28. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]
29. 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 Proceedings of the 17th international conference on World Wide Web, Beijing, China, 21–25 April 2008; pp. 91–100.
30. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
31. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
32. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.