UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Semantic Enhancement for Text Representation

By
Wenfeng HOU

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree

Master of Analytics (Research)

University of Technology Sydney
Faculty of Engineering and Information Technology

June 2021

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Wenfeng HOU declare that this thesis, is submitted in fulfilment of the requirements for the award of Master degree, in the School of Computer Science, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 13.06.2021

ACKNOWLEDGMENTS

My great gratitude goes first and foremost to my supervisor Professor Longbing Cao for the persistent support of my research and study. Without his help and guidance, this thesis could not have been brought to the present appearance.

I also would like to thank Dr. Yanchang Zhao, who, as a co-supervisor, was always happy to help me for my research and study. His guidance helped me in all the time of research and writing of this thesis.

In addition, I would like to thank the fellow lab mates who study in Advanced Analytics Institute. Without their support, I could not finish my research here.

Last but not least, I would like to thank my family for their unconditional support throughout my whole master study.

CONTENTS

List of Figures

Figure **Page**

List of Tables

PUBLICATION

Accepted Paper:

W. HOU, Q. LIU, AND L. CAO, Cognitive aspects-based short textrepresentation with named entity, concept and knowledge, Applied Sciences, 10 (2020), p. 4893.

ABBREVIATION

BERT - Bidirectional Encoder Representations from Transformers

BiLSTM - Bidirectional Long Short Term Memory

BoW - Bag of Words

CBOW - Continuous Bag-Of-Words

CNN - Convolutional Neural Networks

ECKA - Entity-based Concept Knowledge-Aware model

ELMo - Embedding from Language Models

GPT - Generative Pre-Training

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

TF-IDF - Term Frequency - Inverse Document Frequency

TF-ICF - Term Frequency - Inverse Category Frequency

VSM - Vector Space Model

Abstract

Thanks to recent developments in web technology, various textual information can now be found online, including social media, news, product reviews and instant messages. How to automatically classify and organize such texts is currently a topic of great interest. In Natural Language Processing (NLP), text classification is a traditional task and text representation is its foundation. To represent text, we need to obtain a word's representation. The existing language representation models, including Word2vec, ELMo, GPT and BERT, were widely used for word representation. These word representation models were highly successful at processing natural languages. However, they mainly captured implicit representations. Other models that analyzed a text's context can potentially capture richer information which can help deep neural networks gained a better understanding of the text. It is crucial to incorporate semantic information into the text representation because the rich semantics associated with word representations can supplement text representation. New approaches are necessary to represent semantics in combination with existing text representations.

The models presented in this study improved text representation and term weighting by utilizing external knowledge to address the abovementioned research needs. In contrast to previous work, the models proposed here used multi-level knowledge to facilitate the semantic enhancement of text representation by involving external semantic information.

In Chapter 3, we proposed an Entity-based Concept Knowledge-Aware (ECKA) representation model to incorporate semantic information into short text representations. ECKA is a multi-level short text semantic enhancement model for short text representations which extracts semantic features from the word, entity, concept and knowledge levels by CNN. Since word, entity, concept and knowledge entity in the same short text

have different informativeness for short text classification, attention networks were formed to capture aspects-oriented attentive representations from a text's multi-level textual features. The final multi-level semantic representations were formed by concatenating all these individual-level representations, which were then used for text classification.

In Chapter 4, we proposed a hybrid term weighting method that works by utilizing frequency and semantic similarities for the term weighting calculation. When analyzing a term, we first used the Term Frequency-Inverse Document Frequency (TF-IDF) to calculate term weighting. Next, we used a named-entity-based concept-sense disambiguation process to obtain concepts. Following that, we calculated the term's semantic similarity to the document. The TF-IDF weights were then revised according to the term's semantic similarities to reflect both frequency and semantic similarities of the various terms in the text.

All of these models were applied to the text classification tasks. The proposed models' performance in semantic enhancement were compared with different methods to demonstrate their effectiveness.

CHAPTER 1

INTRODUCTION

Natural Language Processing (NLP) can help computers understand, interpret and manipulate human text-based language. Text classification is a traditional task in NLP. Its main components are text representation models and term weighting methods. In the following sections, we present the background of text representation, semantic-based short text representation and term weighting schemes. Afterwards, we discuss the challenges of text representation and term weighting schemes. Finally, we discuss our proposed methods in these areas.

1.1 Background

1.1.1 Text Representation

Thanks to recent developments in web technology, a variety of information in the form of text is available online. Text can be found on social media content, news descriptions, product reviews and instant messages, etc. In

text classification, a text needs to be transformed into vectors that the computer can understand. Therefore, the quality of text representation directly affects classification results. Word representation is used to represent the text while transforming words into feature vectors. One-hot encoding is the simplest way to transform a text into features. Specifically, the terms of a text corpus are used to form a dictionary. In terms of size, the dimensions of one-hot encoding are the same as the dictionary, and the position of the current word is labelled with "1", whereas all others are labelled with "0". One-hot encoding is suitable for category embedding transformation. Bag-of-words (BoW) is an extension of one-hot encoding. For both one-hot encoding and BoW, the dimensions' size is the same as the text corpus dictionary. Usually, dimension sizes are enormous, which can be disastrous for machine learning. Furthermore, in one-hot encoding and BoW, each term is seen as a single item, and the semantic information of the term cannot be involved. Terms that have similar semantic meanings will have an entirely different vector. For example, although the terms *"start"* and *"begin"* are similar, they have different vectors in BoW that do not reflect their semantic similarities. The drawback of this is that it has a bad influence on the machine learning algorithm. Researchers began to use continuous word representation to represent text on a dimensionally smaller set of vectors and address the issues created by synonymous words. Word2vec, which was presented by Mikolov et al. [40], is one of the most common models to learn word embeddings. The model uses a shallow neural network to learn word vectors from a large collection of text. Word vectors obtained from word2vec carry implicit semantic information within them. Since it was proposed, word2vec has

been widely used. However, it still has its limitations. The vector obtained from it is fixed, and it cannot address the problem that the same term can often have different meanings. To solve the problem of polysemous words, researchers began to utilize contextual word representations. One of the famous examples of this is Embedding from Language Models (ELMo) [41]. Based on the context, ELMo uses two-layer bidirectional language models to train word embedding. ELMo is dynamic, and its embeddings change according to the context, even when the term is the same. Models such as ELMo are implicit representation models that can capture richer contextual information through deep neural networks. However, they fail to capture the explicit representations for the text.

1.1.2 Short Text Representation and Semantic Enhancement

Unlike long textual documents, a short text contains a few sentences or even just a few words. For example, Twitter limits its tweet's length to 280 characters. Sparsity and shortness are the two intrinsic characteristics of such short texts. Since short texts lack enough word-based co-occurrences and shared contexts, extracting representative and informative features from them is hard. Therefore, document representation and word embedding methods that heavily rely on word frequency or shared contexts may not capture sufficient information from short texts nor perform well in downstream tasks such as short text classification.

Several reasons motivated us to make semantic enhancement. First, some terms have the same spelling while having different meanings in

different contexts. One solution to solve this problem is to use the named entity technique. The named entity level representation can help to disambiguate terms with same spelling. For example, both the sentences *"WHO has named the disease COVID-19, short for Corona Virus Disease 19"* and *"Corona is the best beer I have ever drunk"* contain the same term *"Corona"*. According to common sense, the first one refers to *"Coronavirus"*, and the named entity is *"Corona_Virus"*, whereas the second one stands for the famous beer brand *"Corona"*, and its entity is *"Corona_beer"*. Hence, one can obtain a more precise representation at the entity level instead of the same word embedding at the word level. The second motivation for this research is that some terms share the same concept. A concept is regarded as a higher perspective description of a thing, which is a benefit for the classifier. For instance, giving a piece of news *"Dunga will attend the award ceremony"*, according to the keywords *"Dunga"* and *"ceremony"*, it is 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 the Brazilian football team that won the 2002 FIFA world cup, and the *"Brazilian football star"* is the concept of the term *"Dunga"*. This example shows that it is easier to determine a short text's category by involving word-related concepts. Hence, we believed that the concept level representation is a significant supplement for short text representation. The last thing that motivated us to undertake this research is that some terms have knowledge level connections. For example, in the knowledge graph, the two terms, *"Cristiano Ronaldo"* and *"Lionel Messi"* have much in com-

mon - both of the two players 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 attributes in the knowledge graph are highly correlated with the same category, namely, *"Sport"*. Hence, mining the entity relationships from the knowledge base can enhance short text semantic representation. Based on these reasons, we decided to perform semantic enhancement for short texts.

To enhance the semantic text representation for short texts, researchers have resorted to external knowledge bases such as concept, which is widely used for this task. "Concept" is one of the knowledge bases that is widely used for semantic enhancement. Wang et al. [58] proposed a "Bag-of-Concept" approach for short text representation. In his approach, instead of a word, for each category, a concept model was constructed. Then, the short text was conceptualized to a collection of corresponding concepts. Wang et al. [60] also presented a deep convolutional neural network model which used concepts, words and characters for short text classification. To evaluate each concept's importance from the concept set, Chen et al. [10] proposed a knowledge-power multiple-attention network that can be used for text classification whereby two attention mechanisms were used to measure the significance of each concept relative to short text attention and concept set.

Knowledge graphs are another effective way to enhance semantic representation. Wang et al. [59] devised a multi-channel Convolutional Neural Networks (CNN) that employs word-level and knowledge-graph-level representations for news' representation. Turker [54] proposed a knowledge-based short text categorization that operates with Wikipedia

[56] as its external knowledge base.

1.1.3 Term Weighting Scheme for Text Analysis

In text classification, each term in the text is usually transformed into a vector space. The vectors are utilized to form the text representation that can be employed in a machine learning model. Usually, we use a number to represent a term's weighting - namely, its importance in the text. There are several reasons why we used term weighting in text analysis: First, from the perspective of computational cost: The large dimensional size of vector space has always been a significant problem for machine learning algorithms. The NLP task can choose the high weighting features to form text representation according to the term weighting, which can reduce dimension for the text. This step can help to improve computational cost. Second, from the perspective of text representation improvement: In the text, not all terms contribute equally to its representation. Hence, we need to select the most important ones using feature weighting. The top k terms can be employed to form the text representation. Third, from the perspective of semantic similarity: The higher similarity between the features and the text, the higher relevance they have. By weighting features in terms of the semantic similarity, the most semantic relevant features can be chosen to form the text representation.

Therefore, appropriately calculating term weighting is the foundation in text classification, and the quality of text representation will directly affect the classification results. The most common method to calculate term weighting is TF-IDF, where, TF represents the number of times a term appears within the document, and IDF stands for a term's ability

to distinguish a document. In addition, many other methods were used for term weighting, including Chi square, mutual information and information gain. Besides traditional methods, researchers have begun to use knowledge base to calculate term weighting. Luo et al. [33] presented a novel term weighting scheme that uses WordNet to calculate semantic similarity. Jatnika et al. [23] proposed a semantic similarity by utilizing word2vec.

1.2 Research Issues, Objectives and Contributions

1.2.1 Research Issues

Research Issue 1: Sparsity and shortness are intrinsic characteristics of short texts. As short texts lack enough word co-occurrences and shared contexts, extracting representative informative features from them is challenging. Therefore, document representation and word embedding methods that heavily rely on word frequency or shared contexts may not capture sufficient information from a short text nor perform well in downstream tasks such as short text classification. Semantic enhancement for short text representation is a common way to address such problems. However, traditional semantic enhancement does not make full use of external knowledge bases because it considers only one aspect (either entity or concept information) of the knowledge base to enrich short text representation.

Research Issue 2: High dimension is a traditional issue in NLP tasks.

We need to reduce the dimension and choose the most important terms to form the text representation. Traditional term weighting methods only use frequency or term relationships to calculate term weighting. Such methods ignore the semantic information within the text. According to common sense, terms with a higher semantic similarity within the document should be assigned a higher term weight. However, traditional term weighting methods do not consider explicit semantic similarities within a document.

1.2.2 Objectives

- **Objective 1: To improve semantic text representation**. In this objective, our primary focus is to improve short text semantic representation. To achieve this goal, we enriched semantic short text representation in multiple ways. First, we got the named entity from the short text. Next, we used the named-entity-based approach to obtain external knowledge information in the form of entity, concept and knowledge graph. Such external knowledge information was utilized to enrich the short text semantic representation. To capture category-related informative representation in terms of multi-level features, we built a joint model that uses a CNN-based attention network to capture respective attentive representations. Afterwards, the embeddings learned from the different aspects were concatenated for the short text representation.
- **Objective 2: To improve term weighting by utilizing semantic information**. To fulfil this objective, we utilized external knowledge to improve term weighting calculation. We calculated the se-

mantic term weighting for each term, and then we revised the frequency-based term weighting according to each term's semantic based weighting.

1.2.3 Research Contributions

This research makes the following contributions:

- We proposed using multiple aspects of short texts (including concepts, knowledge graphs and entities) in combination with a deep multi-level Entity-based Concept Knowledge-Aware (ECKA) representation model to enhance short text semantic representation. (Chapter 3)
- We proposed a hybrid term weighting method that utilizes frequency and concept-based semantic similarities for the term weighting calculation. To obtain the semantic term weighting, we proposed using a named-entity-based concept-sense disambiguation to acquire corresponding concepts for the terms. We implemented a concept-based approach to calculate the semantic term weight. The TF-IDF weights were then revised according to the term's semantic similarities to reflect both terms' frequency and semantic similarities in the text. (Chapter 4)

1.3 Thesis Structure

The structure of this thesis is illustrated in Figure1.1, the details of each chapter are as follows:

- Chapter 2 offers a literature review of text representation and semantic enhancement for short text representation. Term weighting methods are also reviewed.
- Chapter 3 proposes an ECKA representation model that incorporates semantic information into short text representation. ECKA is a multi-level short text semantic representation model that utilizes semantic features taken from the word, entity, concept and knowledge levels.
- Chapter 4 presents a hybrid term weighting method that utilizes frequency and concept-based semantic similarities for term weighting calculation.
- Chapter 5 presents conclusions and future work.

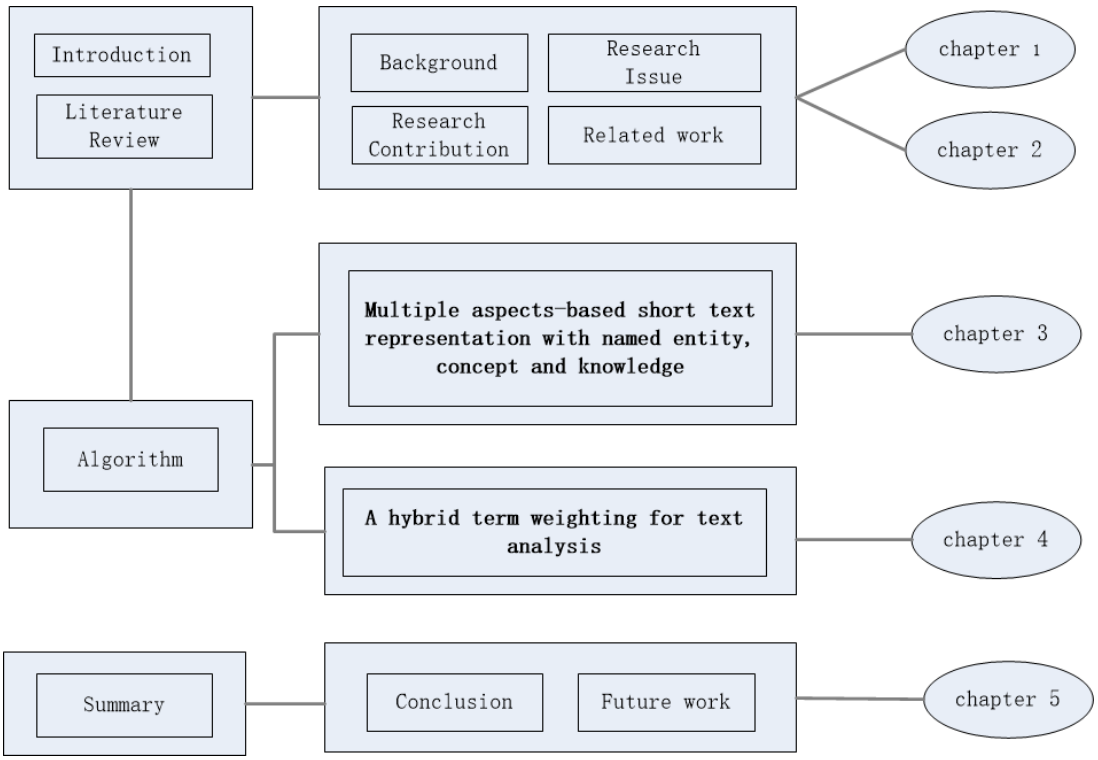


Figure 1.1: Thesis' structure

CHAPTER 2

LITERATURE REVIEW

In this chapter, we discuss background information and relevant techniques. First, we present text representation and semantic enhancement for short text representation. Then we illustrate term weighting scheme for text analysis.

2.1 Text Representation and Semantic Enhancement for Short Text Representation

2.1.1 Word Representation

2.1.1.1 Continuous Word Representation

Word2vec, proposed by Mikolov et al. [40], is one of the most common methods to learn word embeddings. The word2vec algorithm uses a shal-

low neural network to learn word associations from a large collection of text. Word2vec represents each distinct word with a vector that consists of a list of numbers and it can use two models to produce a representation of words: Skip-gram and Continuous Bag-Of-Words (CBOW). Figure 2.1 illustrates the framework of the two models.

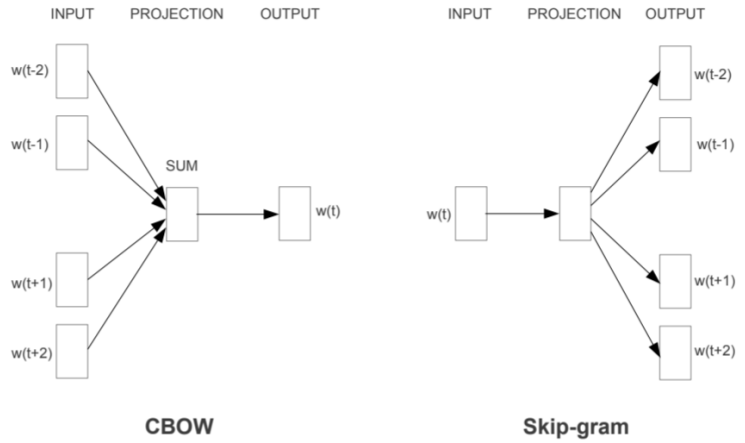


Figure 2.1: Architecture of CBOW and Skip-gram models [39]

CBOW

The CBOW model predicts a target word w_t by utilizing the words around it. The sum of the around word vectors is used to predict the target word, and a pre-defined window size is utilized to determine the number of around words.

Skip-gram

Unlike CBOW, Skip-gram predicts the surrounding words from the current word w_t. Generally speaking, given a word, it learns to predict another word in its context.

2.1.1.2 Contextual Word Representations

Word2vec embedding is a fixed embedding after pre-training. However, in different contexts, the same word can have different meanings. For example, the term *"bank"* can mean either the banking company or the land bordering a river. Hence, the meaning of *"bank"* in a given context has to be determined from the words around it. Therefore, word embedding often fails to represent polysemous words correctly. To solve the problems created by polysemous words, researchers have proposed several contextual word representations models that are discussed in the following section.

ELMo - Embedding from Language Models

ELMo [41] learns the word representation from the context. Unlike the traditional word embedding, in the ELMo model, the same term will have different representations depending on the term's context words. Figure 2.2 illustrates the working mechanism of ELMo. A word representation in ELMo is learned from a two-layer bidirectional language model (biLM). Two layers are stacked together in the biLM model. In each layer, there is a forward and a backward language model. In the above architecture, raw word vectors are formed by using a character-level CNN. Those raw word vectors are then fed into biLM's first layer. The forward language model contains the words before the target word w_i, whereas the backward direction contains the words after the target word w_i. The intermediate word vectors are formed by using the paired information from both directions to be then assigned to the biLM's second layer. The final embedding is the weighted sum of two intermediate word vectors and the raw word vectors.

Transformer-Based Pre-trained Models

Generative Pre-Training (GPT), which was presented by Radford et

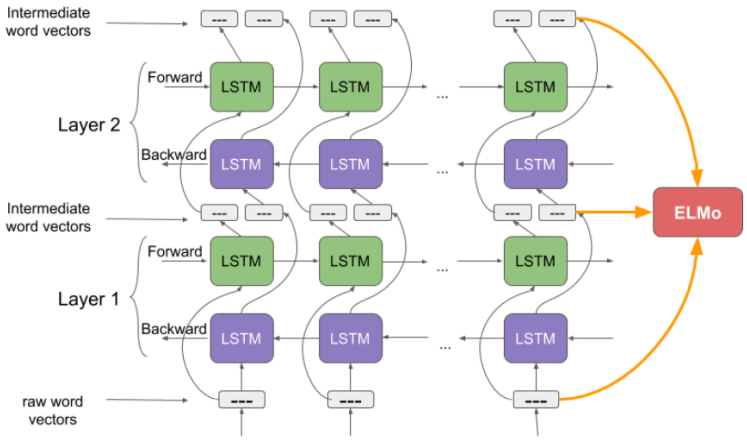


Figure 2.2: Elmo model architecture [1]

al. [44], is one of the transformer-based pre-trained language models. Figure 2.3 shows its framework. Unlike ELMo, GPT uses transformers to replace the Long Short-Term Memory (LSTM) network. Transformers are regarded as the new standard in NLP. There are 12 layers of transformers in GPT, each utilizes 12 independent attention mechanisms. Therefore, GPT has 144 distinct attention patterns. Unlike ELMo, GPT predicts the next word only based on its previous context.

BERT - Bidirectional Encoder Representations from Transformers

BERT [13], presented by Google, utilizes the bidirectional training of transformer models for language modeling. Unlike GPT, which predicts the next word only based on its previous words, BERT predicts the next word based on the words from both before context and after context.

[1]This image is taken from: https://analyticsvidhya.com/blog/2019/03/learn-to-use-elmo-to-extract-features-from-text/

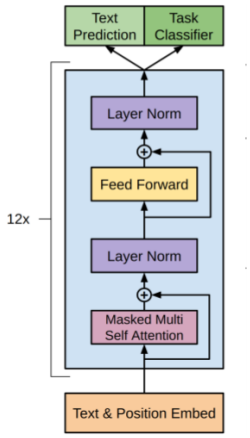


Figure 2.3: GPT model architecture [44]

2.1.2 Short Text Representation and Semantic Enhancement for Short Text Representation

In NLP, short text classification is an important task. Many traditional methods such as BoW, SVM and KNN have been used for this task. Recently, deep neural networks have been increasingly employed in short text analysis. For example, Kim [25] first introduced CNN for text classification. CNN was utilized to extract local and position-invariant features. Recurrent Neural Network (RNN) is another approach for text processing. Unlike CNN, RNN is good at processing long-range semantic dependency rather than local key phrases. Yang et al. [70] presented an attention model to process the problem of different words in a document with informative differences. Sinoara et al. [51] proposed a knowledge-enhanced model to represent document representation which based on embedded words and word sense. Alam et al. [2] proposed a model that utilizes entity-based short text classification with convolutional neural networks.

Concept-Based Semantics Enhancement

Deep models are flexible when they were applied to short text classification tasks. However, due to short texts' shortness and sparsity, it is tricky for them to obtain enough semantic information from those short texts. Therefore, finding ways to enrich short text semantic information with extra knowledge or common sense borrowed from other sources has become an essential issue for researchers. "Concept" is an aspect that is extensively used for text semantic enhancement. Microsoft Concept Graph is a large graph of concepts used by researchers for semantic enhancement. The graph, which has 5.3 million concepts, is an extensive concept-based system, whose concepts were learned from web search logs and billions of website pages [21, 52, 62–65]. Wang et al. [58] presented a "Bag-of-Concepts" approach to improve short text representation. Rather than using words, their model conceptualized the short text in a collection of corresponding concepts. Then, for each category, a concept model was constructed. Wang et al. [60] also presented a deep CNN model that utilizes concepts, words and characters for short text classification. To evaluate each concept's importance from the concept set, Chen et al. [10] proposed a knowledge-power multiple attention network that can be used for text classification. Specifically, two attention mechanisms were used to measure each concept's importance via two aspects: concept towards short text attention, and concept towards concept set. Xu et al. [67] proposed a model that incorporates context-relevant concept into text representation. In their model, two layer CNN and an attention layer were used to extract concepts and context, respectively. To get the context-relevant concepts. Cheng et al. [11] utilized both words and concepts embedding to represent the text. Huang et al. [22] proposed a model which employs short text

conceptualization algorithm for the short text representation. Li et al. [28] put forward a model that uses semantic contexts into short texts' representation to solve the data sparsity issue. In addition, concept also utilized in the other NLP tasks. Cambria et al. [6] proposed a termed concept-level sentiment analysis through concept information.

Those approaches that used concepts to enrich the short text representation achieved some success in text classification. The terms in the short text were conceptualized through an external knowledge base (Probase). Concepts allow for a higher perspective of description that improves text classification. Furthermore, explicit and implicit representations can be captured for the short text representation by utilizing terms and concepts. However, limitation still exists in these approaches. First, they cannot weigh the most informative term's representation. Not all concepts contribute equally to the short text representation. Short text classification may be determined by some special words, while these approaches failed to capture this characteristic. Second, ambiguous issues cannot be entirely resolved, concept conceptualization by single terms may still have ambiguities. Third, these approaches failed to capture semantic representation from other levels. For example, terms' knowledge level connection representation, while representing those latent connections, is a significant supplement to semantic representation.

Knowledge Graph and Knowledge Graph Embedding

Knowledge graph is another effective tool to enhance semantic representation, it comes from human knowledge, and it can be understood as a formal understanding of the world. Discovering ways to integrate human knowledge into machine learning is a popular research topic today. Knowl-

edge graph store human knowledge using an entity-relation-entity triple. The following sections review several knowledge graphs.

DBpedia

The DBpedia [26] community project extracts structured, multilingual knowledge from Wikipedia and has more than 4 million entities in different domains, including organizations, places and people. It uses the resource description framework (RDF) to describe the extracted information. The RDF in the DBpedia links to more than 30 external data sources so that their data can be used in combination with DBpedia data.

Freebase

Freebase [4], developed by Metaweb Technologies, is a collaborative knowledge base. The data in the Freebase was obtained from multiple sources. Freebase is available for non-commercial and commercial use. In 2014, it was closed, and the data was transferred to Wikidata by its new owner Google.

YAGO

YAGO [14], derived from Wordnet, Wikipedia and GeoNames, is a sizeable semantic knowledge base. Currently, it has more than 10 million knowledge entities in different domains, such as organizations, cities and people.

Among those knowledge graphs, we selected DBpedia as our knowledge base due to the following reasons: i) it can enhance the semantic representation from the knowledge-level connection; ii) it is the largest knowledge base that extracts knowledge from Wikipedia; and iii) many tools were developed to extract the information from its resources. One of them is DBpedia Spotlight - a tool to annotate entities from its resources

automatically. We also used it as an annotating tool for entity extraction. Based on the above reasons, we selected the DBpedia as the information source.

Knowledge Graph Embedding

In the knowledge graph, structural information is stored as an entity-relation-entity triple. Such structural information must be transformed into a low-dimensional representation vector in order to be used in a machine learning model. Knowledge graph embedding is employed to learn the knowledge representation of entities and relationships. Translation based methods are the most popular ones to learn the embedding of knowledge graphs, as introduced below.

TransE

TransE [5] is a typical translation method used to learn low-dimensional embedding for the knowledge graph entity-relation-entity triples. Figure 2.4 illustrates the simple working mechanism of TransE. In the figure, h represents the head representation vector, r represents the relation representation vector and t represents the tail representation vector. TransE aims to make the sum of h and r as close as possible to the t. The score function of TransE is denoted as:

$$f(h,r,t) = \|h + r - t\|_{l\,1/l2} \tag{2.1}$$

where l_1 is the L1-nom and l_2 is the L2-norm.

TransH

Unlike TransE, TransH introduces a new approach that interprets the relation as a translating operation on a hyperplane. Each relation has two

[1]This image is taken from: https://towardsdatascience.com/summary-of-translate-model-for-knowledge-graph-embedding-29042be64273

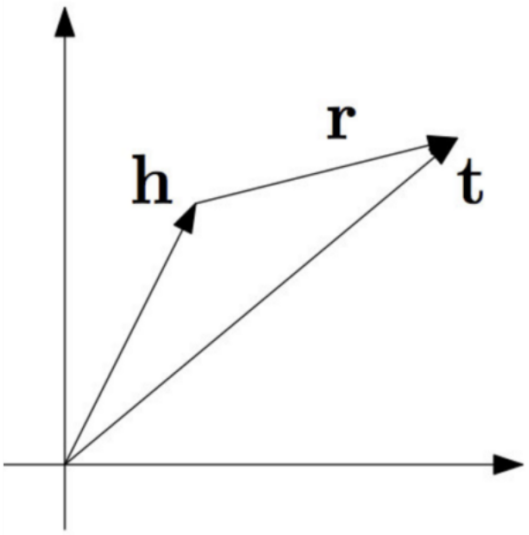


Figure 2.4: Illustration of TransE [1]

vector. Figure 2.5 shows the simple working mechanism of TransH. The score function of TransH is denoted as follow:

$$f(h,r,t) = -\|h_\perp + r - t_\perp\|_2^2 \tag{2.2}$$

where $h_\perp = h - w_t^r h w_r$, $t_\perp = t - w_t^r t w_r$. w_r is the hyperplane, $h_\perp$ is the projection of h to the hyperplane w_r and $t_\perp$ is the projections of t to the hyperplane w_r.

TransR

TransH and TransE assume that both the entity and relation are located in the same semantic space. However, the entity will have different semantic meanings under different relations. To solve this problem, TransR uses a different approach in which entity and relation are mapped in separate spaces. Figure 2.6 shows the simple working mechanism of TransR, whose

[1]This image is taken from: https://towardsdatascience.com/summary-of-translate-model-for-knowledge-graph-embedding-29042be64273

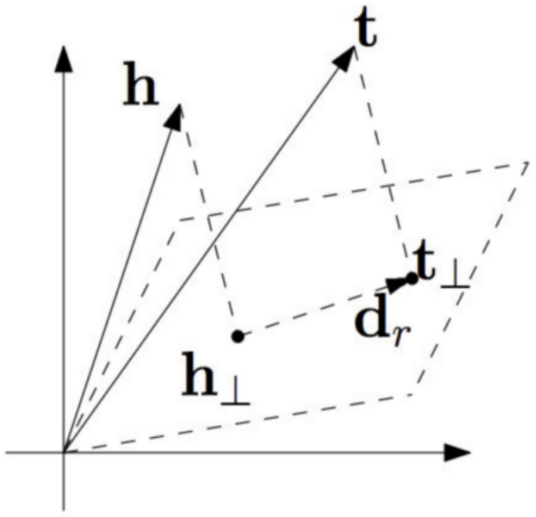


Figure 2.5: Illustration of TransH [1]

score function is denoted as follow:

$$f(h,r,t) = -\|h_r + r - t_r\|_2^2 \tag{2.3}$$

where $h_r = hM_r$, $t_r = tM_r$. M_r is the projection matrix.

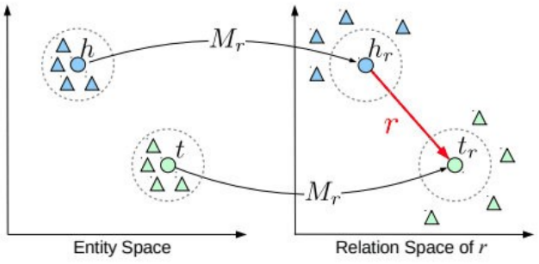


Figure 2.6: Illustration of TransR [1]

Knowledge-based Semantic Enhancement

Knowledge-based semantic enhancement has been widely employed in recent research for the semantic enhancement for short texts representa-

[1]This image is taken from: https://towardsdatascience.com/summary-of-translate-model-for-knowledge-graph-embedding-29042be64273

tions. Wang et al. [59] devised a multi-channel CNN by fusing the word and knowledge graph levels of representations for news text representation. Gao et al. [17] presented a word-level and knowledge-level-based self-attention mechanism for the semantic enhancement of text.

Those models which applied knowledge base to enrich the text representation is a novel approach to enrich text representation. The latent knowledge level connection can be captured through the knowledge graph. Those connections can be seen as an implicit representation that benefits text classification tasks. However, since those models were mainly utilized in the news recommendation tasks, there would still be some disadvantages if we applied them in the text classification task. First, if the entity of the term is missing in the knowledge base, we cannot obtain the implicit representation for this term. In addition, those models failed to capture the explicit representation, which may lead to semantic information missed in the text representation.

Entity-based Semantics Enhancement

For further semantic enhancement, entities are commonly employed together with the knowledge base. Flisar et.al [16] proposed an entity-based semantics enhancement for text classification that uses entities and their related attributes for short text enhancement. T̈urker [54], on the other hand, proposed a knowledge-based short text categorization that utilizes an external knowledge base (Wikipedia) and entities. Li et al. [27] also proposed a Wikipedia based model that uses Wikipedia to enrich text representation. Li et al. [29] proposed a model incorporating knowledge base - probase to enrich text representation, whereas Ren et al. [45] proposed a multi-stream neural network for text classification using background

knowledge consisting of keywords and co-occurred words extracted from external corpus.

Entity-based semantic enhancement opens another direction to text classification. Entities can be used for term disambiguation. In addition, the number of features will become smaller by utilizing entities, which benefits reducing the computational cost. However, limitations still exist. For example, entity-based representation is implicit, and the explicit representation cannot be captured. In addition, entities may not be found in the knowledge data base, which may fail to capture the implicit representation.

Table 2.1 shows current models used for text representation and semantic enhancement. Although such methods gained more accurate short text representations, limitations still exists, such as on the way of combining extra knowledge bases - namely, these models still suffer from making full use of external knowledge bases. Moreover, they considered only one aspect (only the entity or concept information) from knowledge bases to enrich the short text representation. To overcome these issue, we introduce our model for semantic enhancement in the Chapter 3.

2.2 Term Weighting Scheme for Text Analysis

In NLP, text classification is a significant task. In the vector space model (VSM), each document is presented as a vector of terms, and a corresponding weights vector. Usually, we use a number to represent term weighting - namely, a term's importance in a text. The most important terms are

Table 2.1: Comparison of current text representation and semantic enhancement models

Existing Work	Model	Pros	Cons
Legacy word representation	One hot encoding, BoW	It is easy to compute.	It cannot capture implicit and explicit semantic information from the text.
Continuous word representation	Word2vec, Glove	It can capture implicit semantic information.	It cannot capture explicit semantic information from the text.
Contextual word representation	Emlo, GPT, Bert	It solves the problem of polysemy.	It cannot capture explicit semantic information from the text.
Concept-based enhancement	-	It can capture the explicit semantic information from concept.	i) It cannot weigh the most informative terms' representation. ii) Ambiguous issues cannot be entirely resolved. iii)It lacks other level semantic information.
Knowledge-based enhancement	-	It can capture the latent semantic information from knowledge level connection.	i)It cannot capture the explicit representaion. ii)It lacks other level semantic information.
Entity-based enhancement	-	It can capture the implicit semantic information from named entities.	i) It may fail to capture the entity's representation when the entity is missing in entity knowledge base. ii) It cannot capture the explicit representaion. iii)It lacks other level semantic information.

selected to form the text representation. Therefore, appropriately calculating the term weighting is the foundation of text classification tasks, and the quality of text representation directly affects classification results. Several ways to calculate term weighting are described in the following section.

TF-IDF

TF-IDF is a classic method to calculate term weighting that can be represented as follows:

$$\text{TF-IDF} = \text{TF}(i, j) \cdot \text{IDF}(i) \tag{2.4}$$

$$\text{TF}(i\ ,\ j) = \frac{\text{Frequency of term } i \text{ in document } j}{\text{Number of terms in document } j} \tag{2.5}$$

$$\text{IDF}(i) = log(\frac{\text{Number of documents}}{\text{Number of documents with term } i}) \tag{2.6}$$

where i represents the term and j stands for document.

TF stands for the number of times a term appears within the document. DF represents the document frequency, namely the number of times that the term appears within the collection. Inverse document frequency measures how common or rare the word is within the whole document collection. The IDF value for the more common words is close to 0. Hence, if the word is very common and occurs in many documents, this value will be close to 0. TF-IDF does considers not only the co-occurrence frequency but also takes the term's discriminate ability into account. Based on the TF-IDF, Sabbah et al. [47] proposed a modified frequency-based term weighting scheme that takes the number of missing terms into account by calculating the term weighting. To handle documents' length and frequency distribution in TF-IDF, Roul et al. [46] presented a modified TF-IDF term weighting model that uses a length normalizing factor to improve TF-IDF. The TF-IDF method is vulnerable to biases as the most important terms are sometimes referred to as noise, hence, Chen et al. [9] proposed an improved TF-IDF method to overcome this issue.

Information Gain

Information gain [69] is widely used to calculate term weighting in term

selection by measuring the difference in category prediction entropy based on whether a term is absent or present in a document. Information gain is easy to filter out of the characteristic terms with a low IG score, but those terms sometimes have a strong text-type identification ability. To overcome this issue, Zhu et al. [72] recommended an improved information gain feature selection method with word embedding to overcome it.

Mutual Information

Mutual information measures how much information the absence and presence of a term contribute to making correct classification decisions.

Chi-Square

Chi-Square measures the correlation between categories and terms through hypothesis testing in statistics. Specifically, it assumes that the term is directly unrelated to the category. If the test value deviates from the threshold, the original hypothesis can be confidently negated, and the alternative hypothesis that the term and the category have a high degree of relevance can be accepted.

Semantic-based Term Weighting

Besides traditional methods, researchers have begun to use the knowledge bases to calculate term weighting. Luo et al. [33] presented a novel term weighting scheme that employs WordNet to calculate semantic similarity, and Jatnika et al. [23] proposed a semantic similarity through Word2vec. Semantic similarity between terms and document is calculated through Word2vec and can be seen as term weight. To evaluate the performance for term weighting, term weighting is used in the feature selection tasks. To capture the most informative semantic features from the text, many researchers have applied ontology on feature selection

[1, 3, 18, 32, 34, 48, 49, 53, 68]. Mendez et al. [37] also proposed a feature selection method that takes advantage of semantic ontology, which ontology is obtained from wordnet. Matsuo et al. [35] proposed a semantic term weighting for clinical texts that uses UMLS's (Unified Medical Language System) medical resources. In addition, to capture more semantic information for the term, concept is also utilized to enhance the semantic based feature selection. To take advantage of the concept of the term, Zhang et al. [71] presented a model which utilizes concept information in term weighting calculation. To consider concepts and relations to represent the term's importance, Qazi and Goudar [43] proposed a model which uses the ontology-based term weighting to select the feature. Those methods achieved certain success in term weighting calculation.

Traditional term weighting methods (e.g. TF-IDF, information gain, mutual information) were widely used at the early age. Those models were easy to implement and their efficiency is high. However, limitations still exist in those approaches, as they only consider the term frequency or term relations but cannot capture both the implicit and explicit term weightings. Concerning the word2vec-based semantic term weighting, those models can capture implicit term weighting, but fail to capture explicit term weighting, whereas the WordNet-based semantic term weighting is limited because its database is a bit out of date. Furthermore, it lacks the phrase relation.

Table 2.2 illustrates currently used term weighting models. Although such methods gained more accurate term weighting, limitations exist, such as that they cannot capture both implicit and explicit weighting for the text. To overcome those issue, we introduce our model for term

Table 2.2: Comparison of current term weighting models

Existing Work	Pros	Cons
TF-IDF	It is easy to compute.	It cannot capture semantic similarities between terms.
Information gain	Simple interpretation.	i) It does not work well for large number of distinct values. ii) It cannot capture semantic weighting.
Mutual information	It can consider third or higher order feature interaction.	It cannot capture semantic weighting.
Word2vec-based semantic similarity	It can capute the implicit semantic term weighting.	It cannot capture explicit semantic weighting.
Semantic-based term weighting (WordNet- based).	It can capture the semantic term weighting.	Wordnet's database is a bit out of date and lacks the phrase relation.

weighting calculation in Chapter 4.

CHAPTER 3

MULTIPLE ASPECTS-BASED SHORT TEXT REPRESENTATION WITH NAMED ENTITY, CONCEPT AND KNOWLEDGE

3.1 Introduction

Text representation is the foundation of text classification. To represent a text, we need to obtain the word representation. The existing language representation models, including Word2vec, ELMo, GPT, and BERT, were widely used for word representation. These word representation models were excellent at processing natural languages. However, they mainly captured implicit representations by using deep neural networks. Therefore, they may not capture sufficient information from short texts, and they may not perform well in text representation.

Semantic enhancement for short text representation is a common way to address these problems. To enhance the semantic text representation

for short texts, researchers have tried to exploit external knowledge bases. Concept is one of the knowledge bases that was widely used for semantic enhancement. Wang et al. [58] proposed a "Bag-of-Concept" approach for short text representation. In their model, instead of using a word, for each category, a concept model was constructed. Next, the short text was conceptualized as a collection of corresponding concepts. Wang et al. [60] also presented a deep CNN model that utilizes concepts, words, and characters for short text classification. To evaluate each concept's importance from the concept set, Chen et al. [10] proposed a knowledge-power multiple attention network that can be used for text classification, two attention mechanisms were used to measure each concept's importance via two aspects: concept towards short text attention, and concept towards concept set.

Knowledge graphs is another effective tool to enhance semantic representation. Wang et al. [59] devised a multi-channel CNN through knowledge graph-level and word-level representations for news text representation. Turker et al. [54] also presented a knowledge-based short text categorization method that resorts to the external knowledge base of Wikipedia.

Challenges

Although semantic enhancement methods yield more accurate short text representations, limitations still exist when it comes to combining extra knowledge bases. In other words, researchers are still struggling to make full use of external knowledge bases. They often used only one aspect (either entity or concept information) of a knowledge base to enrich short

text representation. However, semantic text representation can be improved through multiple external knowledge bases.

Objectives

Our objective is to improve semantic short text representation. To do so, we utilized entities, concepts and knowledge graphs to make semantic enhancement. We also captured category-related informative representations to improve the semantic short text representation.

Contributions

To address above issues, we used multiple cognitive aspects [7, 8, 20] of short texts, including concepts, knowledge graph and entities to improve short text representation. We also used a multi-level, Entity-based Concept Knowledge-Aware (ECKA) representation model to enhance short text semantic representations. The main contributions of our model are the following:

- We proposed a multi-level model which learns short text representation from different aspects, respectively. To capture more semantic information, we utilized a named-entity-based approach to obtain the external semantic information from entities, concepts and knowledge graphs. External knowledge information was then utilized to enrich short text semantic representation.
- To capture category-related informative representation in terms of multi-level features, we built a joint model by using CNN-based attention networks that capture features' attentive representations. Afterwards, the embeddings learned from the various levels were concatenated to form the short text representation.
- For this study, we conducted extensive short text classification ex-

periments using three datasets. The results demonstrates that our model performs better than current typical methods.

3.2 Multiple Aspects-based Short Text Representation with Named Entity, Concept and Knowledge

The framework of our proposed ECKA representation, which is illustrated in Figures 3.1 and 3.2 further shows its semantic information retrieval module. Our model comprises three modules: the semantic information retrieval module, the feature extraction module, and the attention module. The semantic information retrieval module as illustrated in Figure 3.2, retrieves entity, concept and knowledge graph from the external knowledge bases. The feature extraction module and the attention module are shown in Figure 3.1. The feature extraction module implemented by CNN was used to extract the local and position-invariant features from multiple sources. The attention module was then used to capture category-related informative representations from multi-level features, respectively. Taking a short text as input, our model first extracted all the entities implicated in the text by using DBpedia Spotlight [36]. The model then retrieved the relevant concepts and knowledge graph entities by using Microsoft Concept Graph and DBpedia, respectively. TransE was employed to learn the knowledge graph embedding. We also utilized CNN with an attention network to capture category-related informative representation from multi-level features respectively. Finally, the multi-level semantic text representation was concatenated and fed into a fully-connected layer

to obtain the category probability distribution. We describe the details as follows.

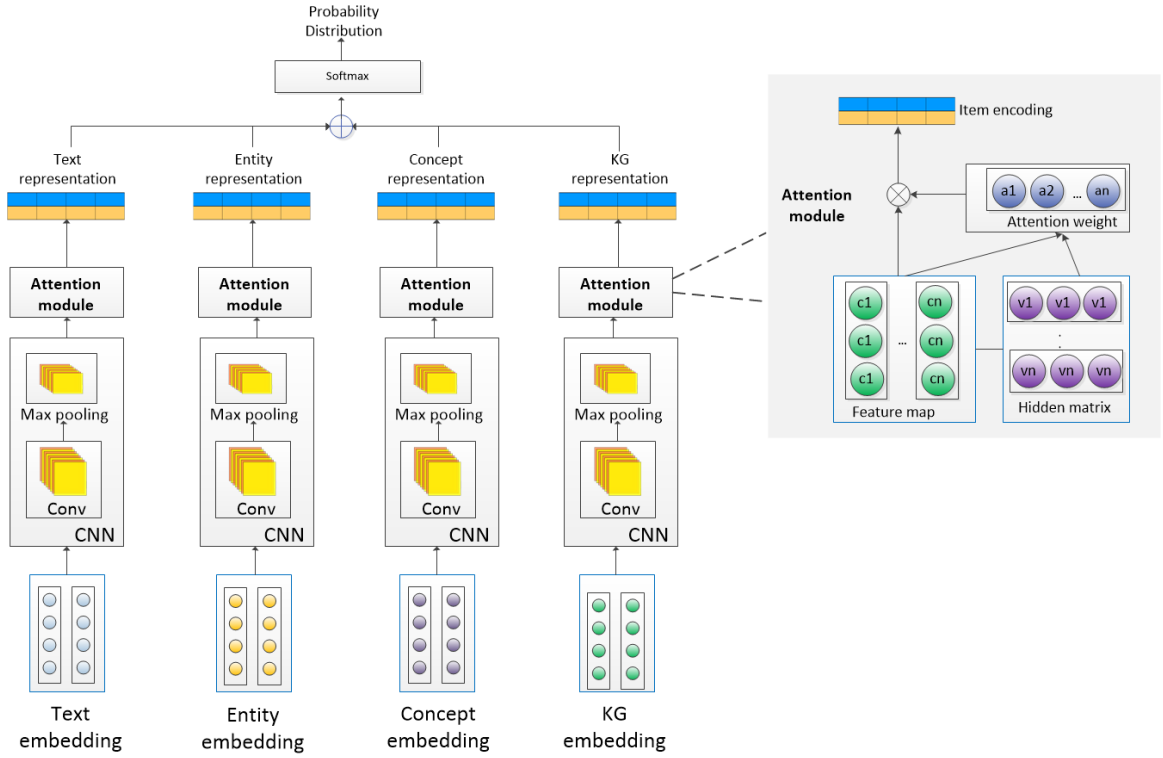


Figure 3.1: The framework of Entity-based Concept Knowledge-Aware (ECKA).

3.2.1 Semantic Information Retrieval Module

This module aims to retrieve the relevant entities, concepts and knowledge graphs from the short text. Firstly, we extracted entities from short text. Entity annotation and linkage is the foundation for our model. Some recently proposed annotation and linking tools, such as DBpedia Spotlight [36], TagMe [15], and Wikify! [38], can satisfy our needs here. For this particular study, we chose DBpedia as our knowledge base and DBpedia Spotlight as our annotation tool. With DBpedia Spotlight, we linked the extracted named entities of the input short text to the DBpedia resources.

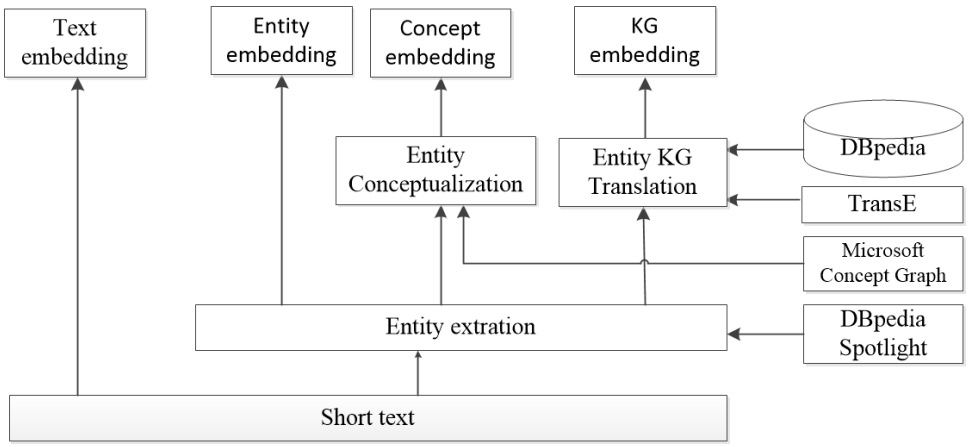


Figure 3.2: The semantic information retrieval module.

Next, we obtained the relevant concepts for the extracted entities. ConceptNet [30] and Microsoft Concept Graph [21, 24, 52, 62–65] are two widely used toolkits that can obtain the concept for an entity. The model described here used Microsoft Concept Graph, which has 5.3 million concepts learned from billions of website pages and search logs. We obtained the knowledge graph for the relevant entities through DBpedia. A typical knowledge graph is a set of relationship triples (h,r,t) in which h stands for head, t represents tail and r stands for relation. The structural knowledge graph information needs to be transformed to a low-dimensional vector which can be used in machine learning. Many transformation methods can learn the low-dimensional vector spaces for knowledge graphs. A comparison of some widely used methods, such as TransE, TransH and TransR, can be found in reference [61]. Since TransE is the most common and effective embedding method, we used TransE as the knowledge graph embedding method in our model.

3.2.2 Feature Extraction Module

This module used the word, entity, concept and knowledge graph to generate multi-level semantic short text representations. This module has three components: input layer, embedding layer and representation layer. The input layer retrieved different sources from external knowledge bases, whereas the embedding layer received the embedding from input layer and translated the different embeddings into the same vector space. The representation layer extracted the higher level features from the embedding layer. The details of each layer are shown as follows.

3.2.2.1 The Input Layer

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

- The Word Set: This set contains all the words in each short text. W = $\{w_1, w_2, w_3, ...w_n\}$.
- The Entity Set: The entity set is denoted as E = $\{e_1, e_2, e_3, ..e_n\}$. This set represents the entities extracted from the short text via DBpedia Spotlight.
- The Concept Set: Each entity's concept was 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 K = $\{k_1, k_2, k_3, ...k_n\}$. It is the same as the entity set, but its representation was learned from different aspects respectively.

3.2.2.2 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 3.2. We used Google pre-trained word2vec embedding to obtain the embeddings for the first three sets. These sets can be represented as W_e= $\{w_{1e}\ w_{2e}\ w_{3e}\ ...w_{ne}\}$, E_e=$\{e_{1e}\ e_{2e}\ e_{3e}\ ...e_{ne}\}$ and C_e=$\{c_{1e}\ c_{2e}\ c_{3e}...c_{ne}\}$, where n is the entity number in the short text. Knowledge graph embedding is learned through the following steps: first, each knowledge graph entity's related entity is retrieved from the DBpedia. Next, the knowledge transforming method TransE was applied to learn the knowledge graph embedding. Finally, as the word, entity and concept embeddings with 300 dimensions were learned by word2vec and the knowledge graph embedding with 50 dimensions was learned from TransE, the two embeddings need to be transformed to the same vector space. The transformed knowledge entity embedding can be represented as:

$$t(k_1...n) = [t(k_1)\ t(k_2)\ ...t(k_n)]. \tag{3.1}$$

In our model, a nonlinear function was used to transform the knowledge entity embedding:

$$t(k) = tanh(Mk + b), \tag{3.2}$$

Here, $M \in \mathbb{R}^{d \times k}$ represents the transformation matrix that is trainable, and $b \in \mathbb{R}^{d \times 1}$ stands for the trainable bias. The knowledge entity embedding was transformed into the same vector space as word2vec through this function.

3.2.2.3 The Representation Layer

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

$$C_i = f(W_c \cdot X_{i:i+h-1} + b_c), \tag{3.3}$$

where h stands for the filter window size, and $i : i+h-1$ 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 represents the nonlinear function. Here, we used $Relu$, and b_c is the bias.

Filtering was then applied to all possible windows, and a feature map was generated:

$$C_e = [c_{e1}, c_{e2}, c_{e3}...c_{en-h+1}]. \tag{3.4}$$

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

$$C_w = [c_{w1}, c_{w2}, c_{w3}...c_{wn-h+1}] \tag{3.5}$$

$$C_c = [c_{c1}, c_{c2}, c_{c3}...c_{cn-h+1}] \tag{3.6}$$

$$C_k = [c_{k1}, c_{k2}, c_{k3}...c_{kn-h+1}], \tag{3.7}$$

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

3.2.3 The Attention Module

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

$$v_i = tanh(W_c C_i + b_c) \tag{3.8}$$

Here, W_c represents a weight matrix and b_c represents the bias. Then, the weight β was calculated through the softmax function as follows:

$$\beta_i = softmax(W_\beta vi) \tag{3.9}$$

Here, w_β is a weight vector. At this point, the entity representation can be calculated as follows:

$$C_\beta = \sum_{i=1}^{h} \beta_i C_i. \tag{3.10}$$

As there are multiple window sizes of the filter, there are multiple feature maps. For each feature map C, a maxpooling function was then applied to get the final pooling vector:

$$C_\beta = argmax(C_{\beta n}), \tag{3.11}$$

Here, n represents the length of the convolution window. So far, the representations for words, entities, concepts and knowledge can be represented

as:

$$R_w = argmax(C_{\beta wn}) \tag{3.12}$$

$$R_e = argmax(C_{\beta en}) \tag{3.13}$$

$$R_c = argmax(C_{\beta cn}) \tag{3.14}$$

$$R_k = argmax(C_{\beta kn}) \tag{3.15}$$

All these different-level representations were concatenated to get the final short text representation R as follows:

$$R = [R_w \oplus R_e \oplus R_c \oplus R_k]. \tag{3.16}$$

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

3.3 Experiments

Our experiment was implemented in Python Keras with three datasets. We demonstrated the evaluation for two aspects: (1) the accuracy of short text classification results; and (2) the variants of our model. The goal here is to evaluate how semantic enhancement from the different levels (word, entity, concept and knowledge graph) affects our model's performance. The performance was then compared to various state-of-the-art text classification models.

3.3.1 Datasets

The three datasets can be described as follows.

Google Snippet - This dataset was adopted from Pan et al. [42]. This snippet refers to the descriptive portion of a google search listing. The google search snippet has eight classes and contains 10,060 training and 2,180 testing samples. The average length of this data set is 12, and the details of each category are shown in Table 3.1.

Twitter - This dataset is a publicly available dataset collected from Github[1], and it includes two categories - sport and politics, and it contains 4567 training samples and 1958 testing samples. The details of each category are shown in Table 3.2.

AG news - This dataset contains four categories of news, and each category has 30,000 training texts and 1,900 testing texts. We only used the title in our experiment, as it can better illustrate the ability of ECKA on short text classification. The details of each category are shown in Table 3.3.

Table 3.2: 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		

[1]https://github.com/vinaykola/twitter-topic-classifier

Table 3.1: 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 3.3: 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		

3.3.2 Data Pre-processing

A typical data pre-processing pipeline was applied to obtain the word level representation.

- Tokenization - Tokenization involves splitting the text into minimal meaningful units. In our model, the short text was split into single word.
- Stemming - We used NLTK's [31] PorterStemmer for the word stemming.

- Stop words removal - Stop words are common but meaningless words. Stop words removal was executed by using NLTK stop words collection.

3.3.3 Baselines

To measure the improvement of our model, we compared it with multiple traditional and state-of-the-art methods:

BoW+TFIDF - BoW is a traditional text representation method widely used in NLP. In BoW, term frequency is used as weight. In this experiment, we used TF-IDF instead of term frequency as weight.

CNN - Kim [25] first introduced CNN for text classification. Only a word embedding layer was used in this network and we used the same parameter settings as our proposed model.

LSTM - LSTM [19] is a variant of RNN. It can capture the long-term dependencies among words in short texts. Only a word embedding layer was used in this network.

Bi-LSTM - Bi-LSTM is a bidirectional LSTM [50], which learns the long-term bidirectional dependencies between the time steps of the sequence data. Only a word embedding layer was used in this network.

GRU - Gate Recurrent Unit (GRU) [12] is similar to LSTM but it has fewer parameters than LSTM. Only a word embedding layer was used in this network.

Attention - Attention [55] is a widely used mechanism in NLP. We used self-attention in our experiment, and only a word embedding layer was used in this network.

KBSTC - This method was presented by T̈urker et al. [54], and it utilized entity and knowledge base (Wikipedia) for short text classification.

WCCNN—This model was presented by Wang et al. [60]. It utilized word embedding and concept embedding for the short text classification. We re-implemented their code for evaluation on the twitter's and google snippet's data sets.

3.3.4 Parameter Setting

We used google's pre-trained, 300-dimension word2vec as the word embedding. The knowledge graph embedding trained by TransE had a dimension of 50. For twitter's dataset, the kernel window size of the convolutional layer is [2,3,4]. For google snippet and ag news, the kernel window size changed to [2,3,4,5,6]. The mini-batch size is 64, and the epoch is 10. For the google snippet and ag news datasets, we used the standard training and validation datasets. For the twitter dataset, we split it manually, allocating 70% for training and 30% for testing. 10 folder validation was then employed to obtain the result.

3.3.5 Result Analysis

The experimental results are shown in Table 3.4. We also tested the variants of our model, and the results are shown in Table 3.5. The results demonstrate that our model performs better than the current state-of-the-art methods.

Table 3.4: Text classification accuracy 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 3.5: The text classification accuracy 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

3.3.5.1 Multiple Sources vs Single Source

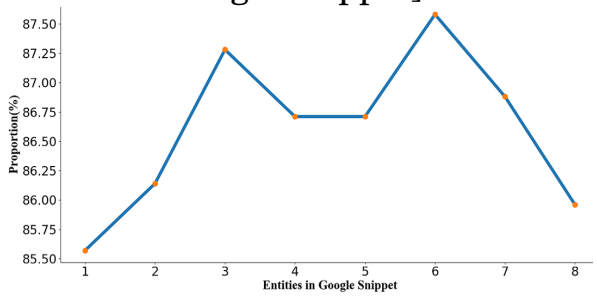
CNN uses the single source of words as its input, and multiple kernels with different sizes were employed on it. From the result in Table 3.4, we can see that CNN performs best among the baseline methods that only use a single source. This improved performance occurs because different

window sizes of convolution operations were employed to extract the local features, which can then enhance the text representation. Our model performs better than the others due to the following reasons: (i) it handled ambiguous terms through the named entity technique, thus providing a more precise representation of the entity, concept and knowledge graph levels; (ii) we enriched the short text representation from different sources. The model learned the superordinate representation via the concept level. The latent semantic representation was obtained through the knowledge entity and the linked entities within the knowledge graph; (iii) we used CNN to extract the local features and attention network to capture the attentive representation from multi-level features respectively, which better captured category-related informative features for short text classification.

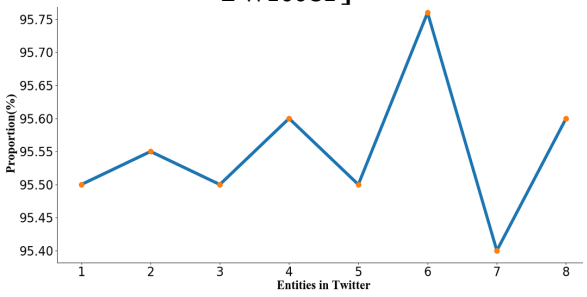
3.3.5.2 Comparison of ECKA Variants on Multiple Sources

In this section, we compare the variants of ECKA in terms of involving external knowledge to show our model design's effectiveness. The results are listed in Table 3.5 and show that: compared to the baseline (which only used words), the semantic enhancement created by using entity, concept and knowledge graph can boost short text classification's performance. This result proves that involving external knowledge can enhance semantic representation. Furthermore, compared to the two-source model, the model with four sources performs more effectively, demonstrating that the used of multiple sources from different aspects is an effective way of improving short text classification.

[Accuracy w.r.t. the number of entities, concepts and KG in a text of Google snippet]



[Accuracy w.r.t. the number of entities, concepts and KG in a text of Twitter]



[Accuracy w.r.t. the number of entities, concepts and KG in a text of AG news]

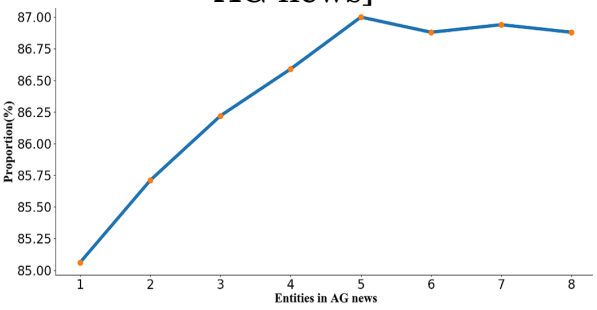


Figure 3.3: Accuracy distribution

3.3.6 Parameter Sensitivity

In this section, we investigated how different numbers of entities affect our model's performance. We applied different numbers of entities in the set [1,2,3,4,5,6,7,8] on the three datasets. The results are shown in Figure 3.3. The results show that the best performance is associated with six entities in google snippet and twitter, and five entities in ag news. Interestingly, performance does not increase when the entity number increases beyond these numbers - perhaps because when most entities

are involved, our model learns the informative representation from them. The learned informative entities then benefit the classification results.

3.3.7 Computational Cost

To improve the computational cost, We undertook the following steps. First, we used the named entity technique, which leads to less irrelevant terms invoke. In machine learning model, the computational cost increases with the number of features increasing. Irrelevant terms are the noise data for the text classification and a significant cause of the dimensionality problem. Second, we extracted the entity, concept and knowledge graph for the short text, which is a pre-processing job. Pre-processing job could be organized offline, which can improve efficiency. However, although we tried to reduce computational cost, some additional work needs to be done in the future. In our model, we used CNN to extract the text feature from the short text. In CNN, useful representations and features were learned automatically from the raw data. Due to its complexity, deep learning-based approaches have a higher computational cost than the traditional machine learning models. In this model, computational cost can be attributed to the number of filters, kernel size, and stride, and so forth. However, from the text classification experiment result in Table 3.4, we can see that deep leaning based approaches have higher accuracy than traditional models. We need to find a balance between computational cost and accuracy. As computational cost is a complex process in deep learning, in the future, we will explore how to reduce them to improve efficiency.

3.4 Conclusions

Short texts' representation cannot be effectively handled by either document representation or the classic NLP tools. Our model involved using multiple-level aspects source from the text, including entity, concept and knowledge graph. Our model is a novel multi-level, entity-based, concept and knowledge-aware model (ECKA) that enhances short text semantic representation. ECKA learned four different levels of semantic information from a short text: the word level, the entity level, the concept level, and the knowledge graph level. CNN extracted semantic features from these different levels. An attention network was employed on the different levels to capture the category-related attentive representations from these multi-level features. Experimenting with short text classification demonstrated the effectiveness and merits of the ECKA model described here, especially when it was compared with traditional and state-of-the-art baseline models. The improvements made by the ECKA model can be attributed to entity identification and knowledge extraction. To further promote ECKA, we will focus on improving the accuracy of entity extraction and employing knowledge-enabled language representation models (e.g., K-BERT) for short text representation.

This research opens new opportunities for semantic text representation. Our further research steps will investigate how to apply our model to the other NLP tasks, such as news recommendation. Since news recommendation needs to consider time, we need to further explore how to integrate our model into it. Moreover, the time complexity brought by the deep neural network also needs further exploration.

CHAPTER 4

A HYBRID TERM WEIGHTING FOR TEXT ANALYSIS

4.1 Introduction

Text documents usually need to be transformed into numeric vectors that can be understood by computer before they can be assigned into machine learning models. Usually, the dimension of vector space is huge, which is a major problem for text classification tasks. Hence, we need to select the most important terms to form the text representation to reduce dimension. Term weighting measures the term's importance in a document. The top k terms with a high term weighting score can be selected to form the text representation. Various term weighting schemes have been used to calculate term weighting, and finding the appropriate one is essential for text classification, as whatever method is used will directly affect classification results. TF-IDF is one of the typical methods to calculate term weighting. Motivated by TF-IDF, Wang and Zhang [57] proposed

TF-ICF (Term Frequency - Inverse Category Frequency) to calculate term weighting. Category frequency (CF) measures how many times a term appears in a given number of classes. Many other methods also can be used to calculate term weighting, such as Chi square, mutual information and information gain. Traditional term weighting utilizes statistical information. However, an alternative method exists that involves exploiting semantic information to calculate term weighting. WordNet is a lexical database used by many researchers. Luo et al. [33] used it to calculate semantic term weighting. Word2vec is also used to calculate the semantic similarity between terms and document. The similarity score is used as the term weighting score. These methods have proved to calculate term weighting effectively.

Challenges:

Traditional term weighting methods only considered term distributions or implicit semantic relevancy while ignoring the explicit semantic similarity of terms. Terms within a document that have higher semantic similarity should be assigned a higher term weight. Therefore, finding an essential way to measure a term's semantic similarities is important for term weighting calculation.

Objectives:

To improve term weighting, we focused on utilizing the external knowledge base. We made use of a concept-based approach to calculate semantic term weighting. Furthermore, we presented a hybrid term weighting method that utilizes frequency and semantic similarity weighting.

Contributions:

The main contributions of our model are the following:

- We presented a named-entity-based, concept-sense disambiguation to obtain the corresponding concepts of a term, and utilized a concept-based approach to calculate semantic weighting for the term.
- We proposed a hybrid term weighting method that involves frequency and concept-based term semantic similarities.
- We applied the hybrid term weighting method to do the text classification task.

4.2 A Hybrid Term Weighting Method for Text Analysis

In this section, we present our term weighting model. Besides term frequency, we took advantage of semantic information to calculate term weighting via the semantic similarities of terms in relation to the document. In our model, we first used TF-IDF to calculate frequency-based term weighting. Next, we calculated a term's semantic weighting by following steps: (1) we retrieved the term's named entity through DBpedia Spotlight; (2) we obtained the candidate concepts for the term by using Microsoft Concept Graph, and, to get the term's corresponding concept, the similarity between the candidate concepts and the named entity was calculated, and the concept with the highest similarity score was selected as the corresponding concept for the term; (3) we used concept vector combinations as the document representation, and we calculated the similarity between the concept and the document to get the term's semantic similarity. Finally, the TF-IDF weights were revised according to the term's semantic similarities. The details are shown as follows.

4.2.1 Frequency-based Term Weighting

In this part, we utilized TF-IDF to calculate the frequency-based term weighting. TF-IDF is a classic method to calculate term weighting. The details can be found in Section 2.2.

4.2.2 Semantic-based Term Weighting

In this section, we proposed a concept-based semantic term weighting method that uses a concept-based approach to calculate the semantic term weighting. To get the corresponding concept of a term, we used the named entity technique to obtain the concept. First, we obtained the relevant DBpedia resource, which can be regarded as the named entity, from the text by using DBpedia Spotlight. The entities in the text can be represent as: E = $\{e_1, e_2, e_3, ..e_n\}$, where e_n represents the entities extracted from the document via DBpedia Spotlight.

Conceptualization

After obtaining the named entity from the text, we must obtain the corresponding concept for the term. We obtained the concept via Microsoft Concept Graph, which is based on Probase. Probase [66] is a knowledge base developed by Microsoft. Probase aims to help machines better understand human language. As Probase has a knowledge base as large as the concept space of a human mind, it has unique advantages. We obtained the top 10 candidate concepts from Microsoft Concept Graph, which utilizes Basic-level Categorization(BLC) [64] to calculate the similarity score between candidate concepts and the term. The BLC calculation can be

define as:

$$BLC(i) = argmax Rep(i,c) \tag{4.1}$$

where $Rep(i,c)$ was calculated as follows:

$$Rep(i,c) = P(c|i) \cdot P(i|c) \tag{4.2}$$

Here, $P(c|i)$ and $P(i|c)$ were calculated as follows:

$$P(c|i) = \frac{n(c|i)}{\sum_{i \in c_j} n(c_j, i)} \tag{4.3}$$

$$P(i|c) = \frac{n(c|i)}{\sum_{i_j \in c} n(c, i_j)} \tag{4.4}$$

where, $P(i|c)$ represents the typicality of an instance i in concept c, and $P(c|i)$ represents the typicality of concept c for instance i. $n(c|i)$ stands for the co-occurrence counts of instance i and concept c, whereas $\sum_{i \in c_j} n(c_j, i)$ indicates the sum of the co-occurrence counts of instance i and instance i belong to all concepts. $\sum_{i_j \in c} n(c, i_j)$ indicates the sum of co-occurrences of concept c and all the instances within concept c, i represents the instance and c represents the candidate concept.

Given the entity list E = $\{e_1, e_2, e_3, ..e_n\}$. we got a candidate list of concepts from the Microsoft Concept Graph for each entity. At this point, it is necessary to perform disambiguation to get the corresponding concept. For instance, the term *"apple"* has two meanings: a kind of fruit, and a famous computer company. For the given text *"I buy an iPhone from the apple store"*, the named entity for apple is *"apple corporation"*. Hence, we utilized the named entity technique to do the disambiguation. Then,

the similarity between the named entity and the concept in the concept candidate list C=$\{c_1, c_2, c_3..c_n\}$ was calculated by using the cosine function. The calculation can be represented with the following equation:

$$Sim(entity, concept) = \frac{\sum_{i=1}^{n} V_e V_c}{\sqrt{\sum_{i=1}^{n} V_e^2}\sqrt{\sum_{i=1}^{n} V_c^2}} \tag{4.5}$$

In the above equation, V_e represents the vector of named entity and V_c represents the candidate concept vector. The candidate concept with the highest similarity score was chosen as the corresponding concept for the term.

To calculate the semantic similarity between the term and the document, we used a concept-based approach to calculate semantic similarity. A text can be represented as d=$\{c_1, c_2, c_3, ..c_n\}$, where c_n represents the term's concept in the text. The term's semantic similarities can be calculated as:

$$Semantic\_Sim(t_c, d) = \sum_{i=1}^{n} Sim(t_c, c_n) \tag{4.6}$$

where document vector d is the sum of concept vectors, t_c represents the concept vector of the current term, and c_n represents the concept in document d.

Finally, the TF-IDF weights were adjusted according to the term's semantic similarities to reflect both its frequency and semantic similarities to the text. The semantic term weighting was represented as:

$$Semantic\_Term\_Weight = TF - IDF \cdot Semantic\_Sim \tag{4.7}$$

4.3 Experiments

To validate the proposed term weighting method's effectiveness, we experimented with text classification by using different term weighting schemes. The experiment was implemented in Python Keras. We chose Support Vector Machines (SVM) as the classifier. Based on the term weighting score, we selected the top k terms to form the text representation. Next, we applied the text representation to the text classification tasks. We then demonstrated the accuracy of the text classification result. The performance was compared with traditional methods.

4.3.1 Datasets

The details of the datasets are listed below.

HuffPost news - This dataset was obtained from Kaggle[1]. It contains around 20,000 news headlines from HuffPost. We used both title and content as input. The details are shown in Table 4.1.

BBC news - This dataset was also obtained from Kaggle[2] and contains BBC news content. There are five categories: business, entertainment, politics, sports and technology. We used both title and content as input. The details are shown in Table 4.2.

[1]https://www.kaggle.com/rmisra/news-category-dataset
[2]https://www.kaggle.com/hgultekin/bbcnewsarchive

Table 4.1: The huffpost news data set

Category	Training	Testing	Features	Values
Politics	5759	612	Avg.Len per doc	540
Business	4841	528	Total document	20822
Entertainment	4179	485		
Sports	3959	459		
Total	18738	2084		

Table 4.2: The bbc news data set

Category	Training	Testing	Features	Values
Entertainment	342	44	Avg.Len per doc	419
Technology	358	43	Total document	2225
Business	468	42		
Politics	478	39		
Sport	456	55		
Total	2002	223		

4.3.2 Baseline

To measure the improvements of our model over previous models, we compared it with many traditional methods:

TF-IDF - TF-IDF is a classic method to calculate term weighting.

TF-ICF - TF-ICF was proposed by Wang and Zhang [57]. It replaces IDF with ICF(Inverse Category Frequency).

WTE - Semantic similarity is calculated by using Word2vec [23]. In this experiment, we used word2vec to calculate the semantic similarity between terms and document. The similarity score was used as term weighting score.

4.3.3 Parameter Setting

Based on the term weighting score, we selected top 10 and 15 terms to form the text representation. We then used SVM as classifier to perform the text classification.

Table 4.3: Text classification accuracy

Model	BBC news	HuffPost news
TF-IDF (top 10 terms)	90.58	79.12
TF-ICF (top 10 terms)	88.36	78.35
WTE (top 10 terms)	82.51	71.73
Proposed Method (top 10 terms)	91.92	80.32
TF-IDF (top 15 terms)	92.37	82.53
TF-ICF (top 15 terms)	91.46	82.77
WTE (top 15 terms)	90.58	77.59
Proposed Method (top 15 terms)	94.62	84.21

4.3.4 Result Analysis

Experimental results are shown in Table 4.3. From the result, we can see that proposed method performs better than the traditional method TF-IDF. This improved performance happens because besides term frequency, we also took advantage of semantic information to calculate term weighting. We also found out that our model performs better than all other methods. The reasons why our model achieves better results than other models are as follows: (1) we used a hybrid term weighting method that utilizes term frequency and term semantic similarity; (2) to calculate a term's semantic similarity, we used a concept-based approach. To handle concept ambiguity, we applied a named entity technique to retrieve the entity for the term, then we obtained the concept set for each term. The concept

with the highest score were selected as the corresponding concept for the term. Finally, the TF-IDF weights were revised according to the term's semantic similarities for the purpose of reflecting both frequencies and semantic similarities of the terms in the text.

4.4 Conclusions

This chapter proposed a hybrid term weighting method that utilizes frequency and concept-based terms' semantic similarities. We first used TF-IDF to calculate the frequency-based term weighting. Then we calculated semantic-based term weighting. To obtain the corresponding term's concept, we used named-entity-based concept-sense disambiguation to select the corresponding concept for term. Next, we adopted a concept-based approach to calculate the semantic similarity. Finally, the TF-IDF weights were revised according to the term's semantic-based term weighting. We then examined the effectiveness of our model in comparison to other term weighting methods. The results shown that our model performs better than other models. Hence, semantic-based term weighting is a supplement for a term-relation-based approach. In the future, we will explore more approaches that use external knowledge to improve term weighting's quality

This work focused on how to improve term weighting. However, the proposed model can also be extended to other NLP tasks, such as information retrieval task which not only needs to consider the term weighting, but also other aspect. In future, we will explore how to apply our model

those NLP tasks. Furthermore, we will investigate other approaches that use external knowledge to improve the term weighting's quality.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

This thesis presented an approach that utilizes the external knowledge bases to enhance semantic text representation. We proposed an Entity-based Concept Knowledge-Aware (ECKA) representation model that incorporates semantic information into short text representation. Furthermore, to capture category-related attentive representations from multi-level textual features, we employed a CNN-based attention network in the ECKA model and presented a hybrid term weighting method that utilizes a term's frequency and semantic similarities to determine term weighting calculation. The results of our experiments show that our proposed models significantly outperform current state-of-the-art models.

5.2 Future Work

This research opens new challenges and opportunities to explore methods of improving text representation and term weighting. Further research efforts should be directed towards some of the following issues and challenges.

Semantic Enhancement for Text Representation:

- In future research, we will explore how to apply our semantic representation model to the news recommendation task. To expand our model to the news recommendation, we may consider adding meta information (e.g., news type, country information) into news representation. Since such meta information reflects a user's reading habit, it can benefit news representation. Furthermore, as a user's interests may change from time to time, we may add the attention model to capture the aggregated representation from the user's browsing history. The aggregated representation can be used to measure the similarity between candidate news. Then, we can use the similarity score as a news recommendation criteria.
- Another potential research direction is how to apply this model to other NLP tasks, which requires considering the time and emotional context (e.g., chatbots). Since the representation learned from this model focuses on the text characteristic, we may need to integrate the context level representation into the current representation. The context level representation can be learned from the transformer layer. Moreover, to make the chatbot's responses not only based on content but also on emotion, we need to further explore how to

integrate the emotion representation into the sequence-to-sequence model.

- Though we have made some efforts to reduce computational cost, we need to do additional work. Deep neural networks have a higher computational cost than traditional methods due to their time and space complexity. Hence, we need to further explore how to reduce the time's and space's cost in deep model.

Semantic-based Term Weighting for Text Analysis:

- In future research, we will explore how to apply the proposed model to information retrieval, which is used to select and rank relevant documents from a large set of candidate documents according to the keyword user input. Our model can be applied to improve the semantic representation for both keywords and candidate documents. The keyword's concept can be obtained from our model, and it can be considered as a semantic enrichment for the keyword. The semantic representation for keywords can be obtained by leveraging the concept and keyword embedding. Then, since our model is designed to calculate the term weighting for the candidate documents, we can obtain the term weighting for each document by applying it to the candidate documents. According to the term weighting score, the top k terms can be selected to form the representation for the candidate documents. The similarity between the keyword and the candidate document can be calculated by the similarity measure function. Then the result of information retrieval can be ranked according to the similarity. Compared to the traditional information

retrieval models, our model can capture the semantic representation for both keyword and candidate document.

- We proposed using the semantic-based term weighting for text analysis. However, term weighting can be learned from different aspects to capture more implicit and explicit semantic representations. For example, integrating the concept with the BERT can obtain the more semantic weighting.

BIBLIOGRAPHY

[1] B. AGARWAL, N. MITTAL, P. BANSAL, AND S. GARG, *Sentiment analysis using common-sense and context information*, Computational intelligence and neuroscience, 2015 (2015).

[2] M. ALAM, Q. BIE, R. TÜRKER, AND H. SACK, *Entity-based short text classification using convolutional neural networks*, in International Conference on Knowledge Engineering and Knowledge Management, Springer, 2020, pp. 136–146.

[3] F. ALI, K.-S. KWAK, AND Y.-G. KIM, *Opinion mining based on fuzzy domain ontology and support vector machine: A proposal to automate online review classification*, Applied Soft Computing, 47 (2016), pp. 235–250.

[4] K. BOLLACKER, C. EVANS, P. PARITOSH, T. STURGE, AND J. TAYLOR, *Freebase: a collaboratively created graph database for structuring human knowledge*, in Proceedings of the 2008 ACM SIGMOD international conference on Management of data, 2008, pp. 1247–1250.

[5] A. BORDES, N. USUNIER, A. GARCIA-DURAN, J. WESTON, AND O. YAKHNENKO, *Translating embeddings for modeling multi-*

relational data, in Advances in neural information processing systems, 2013, pp. 2787–2795.

[6] E. CAMBRIA, S. PORIA, F. BISIO, R. BAJPAI, AND I. CHATURVEDI, *The clsa model: A novel framework for concept-level sentiment analysis*, in International Conference on Intelligent Text Processing and Computational Linguistics, Springer, 2015, pp. 3–22.

[7] L. CAO, *Metasynthetic Computing and Engineering of Complex Systems*, Advanced Information and Knowledge Processing, Springer, 2015.

[8] ——, *Data Science Thinking: The Next Scientific, Technological and Economic Revolution*, Data Analytics, Springer International Publishing, 2018.

[9] C.-H. CHEN, *Improved tfidf in big news retrieval: An empirical study*, Pattern Recognition Letters, 93 (2017), pp. 113–122.

[10] J. CHEN, Y. HU, J. LIU, Y. XIAO, AND H. JIANG, *Deep short text classification with knowledge powered attention*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 6252–6259.

[11] J. CHENG, Z. WANG, J.-R. WEN, J. YAN, AND Z. CHEN, *Contextual text understanding in distributional semantic space*, in Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 133–142.

[12] K. CHO, B. VAN MERRIËNBOER, C. GULCEHRE, D. BAHDANAU, F. BOUGARES, H. SCHWENK, AND Y. BENGIO, *Learning phrase representations using rnn encoder-decoder for statistical machine translation*, arXiv preprint arXiv:1406.1078, (2014).

[13] J. DEVLIN, M.-W. CHANG, K. LEE, AND K. TOUTANOVA, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805, (2018).

[14] M. FABIAN, K. GJERGJI, W. GERHARD, ET AL., *Yago: A core of semantic knowledge unifying wordnet and wikipedia*, in 16th International World Wide Web Conference, WWW, 2007, pp. 697–706.

[15] P. FERRAGINA AND U. SCAIELLA, *Tagme: on-the-fly annotation of short text fragments (by wikipedia entities)*, in Proceedings of the 19th ACM international conference on Information and knowledge management, 2010, pp. 1625–1628.

[16] J. FLISAR AND V. PODGORELEC, *Document enrichment using dbpedia ontology for short text classification*, in Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, 2018, pp. 1–9.

[17] J. GAO, X. XIN, J. LIU, R. WANG, J. LU, B. LI, X. FAN, AND P. GUO, *Fine-grained deep knowledge-aware network for news recommendation with self-attention*, in 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), IEEE, 2018, pp. 81–88.

[18] F. GUTIERREZ, D. DOU, S. FICKAS, D. WIMALASURIYA, AND H. ZONG, *A hybrid ontology-based information extraction system*, Journal of Information Science, 42 (2016), pp. 798–820.

[19] S. HOCHREITER AND J. SCHMIDHUBER, *Long short-term memory*, Neural computation, 9 (1997), pp. 1735–1780.

[20] L. HU, S. JIAN, L. CAO, AND Q. CHEN, *Interpretable recommendation via attraction modeling: Learning multilevel attractiveness over multimodal movie contents*, in IJCAI'2018, 2018, pp. 3400–3406.

[21] W. HUA, Z. WANG, H. WANG, K. ZHENG, AND X. ZHOU, *Short text understanding through lexical-semantic analysis*, in 2015 IEEE 31st International Conference on Data Engineering, IEEE, 2015, pp. 495–506.

[22] H. HUANG, Y. WANG, C. FENG, Z. LIU, AND Q. ZHOU, *Leveraging conceptualization for short-text embedding*, IEEE Transactions on Knowledge and Data Engineering, 30 (2017), pp. 1282–1295.

[23] D. JATNIKA, M. A. BIJAKSANA, AND A. A. SURYANI, *Word2vec model analysis for semantic similarities in english words*, Procedia Computer Science, 157 (2019), pp. 160–167.

[24] L. JI, Y. WANG, B. SHI, D. ZHANG, Z. WANG, AND J. YAN, *Microsoft concept graph: Mining semantic concepts for short text understanding*, Data Intelligence, 1 (2019), pp. 238–270.

[25] Y. KIM, *Convolutional neural networks for sentence classification*, arXiv preprint arXiv:1408.5882, (2014).

[26] J. LEHMANN, R. ISELE, M. JAKOB, A. JENTZSCH, D. KONTOKOSTAS, P. N. MENDES, S. HELLMANN, M. MORSEY, P. VAN KLEEF, S. AUER, ET AL., *Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia*, Semantic Web, 6 (2015), pp. 167–195.

[27] J. LI, Y. CAI, Z. CAI, H. LEUNG, AND K. YANG, *Wikipedia based short text classification method*, in International Conference on Database Systems for Advanced Applications, Springer, 2017, pp. 275–286.

[28] P. LI, L. HE, X. HU, Y. ZHANG, L. LI, AND X. WU, *Concept based short text stream classification with topic drifting detection*, in 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, 2016, pp. 1009–1014.

[29] Y. LI, B. WEI, Y. LIU, L. YAO, H. CHEN, J. YU, AND W. ZHU, *Incorporating knowledge into neural network for text representation*, Expert Systems with Applications, 96 (2018), pp. 103–114.

[30] H. LIU AND P. SINGH, *Conceptnet‚Äîa practical commonsense reasoning tool-kit*, BT technology journal, 22 (2004), pp. 211–226.

[31] E. LOPER AND S. BIRD, *Nltk: the natural language toolkit*, arXiv preprint cs/0205028, (2002).

[32] D. LUNDQUIST, K. ZHANG, AND A. OUKSEL, *Ontology-driven cybersecurity threat assessment based on sentiment analysis of network activity data*, in 2014 International Conference on Cloud and Autonomic Computing, IEEE, 2014, pp. 5–14.

[33] Q. LUO, E. CHEN, AND H. XIONG, *A semantic term weighting scheme for text categorization*, Expert Systems with Applications, 38 (2011), pp. 12708–12716.

[34] A. MARSTAWI, N. M. SHAREF, T. N. M. ARIS, AND A. MUSTAPHA, *Ontology-based aspect extraction for an improved sentiment analysis in summarization of product reviews*, in Proceedings of the 8th International Conference on Computer Modeling and Simulation, 2017, pp. 100–104.

[35] R. MATSUO AND T. B. HO, *Semantic term weighting for clinical texts*, Expert Systems with Applications, 114 (2018), pp. 543–551.

[36] P. N. MENDES, M. JAKOB, A. GARCÍA-SILVA, AND C. BIZER, *Dbpedia spotlight: shedding light on the web of documents*, in Proceedings of the 7th international conference on semantic systems, 2011, pp. 1–8.

[37] J. R. MENDEZ, T. R. COTOS-YANEZ, AND D. RUANO-ORDAS, *A new semantic-based feature selection method for spam filtering*, Applied Soft Computing, 76 (2019), pp. 89–104.

[38] R. MIHALCEA AND A. CSOMAI, *Wikify! linking documents to encyclopedic knowledge*, in Proceedings of the sixteenth ACM conference

on Conference on information and knowledge management, 2007, pp. 233–242.

[39] T. MIKOLOV, K. CHEN, G. CORRADO, AND J. DEAN, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781, (2013).

[40] T. MIKOLOV, I. SUTSKEVER, K. CHEN, G. S. CORRADO, AND J. DEAN, *Distributed representations of words and phrases and their compositionality*, in Advances in neural information processing systems, 2013, pp. 3111–3119.

[41] M. E. PETERS, M. NEUMANN, M. IYYER, M. GARDNER, C. CLARK, K. LEE, AND L. ZETTLEMOYER, *Deep contextualized word representations*, arXiv preprint arXiv:1802.05365, (2018).

[42] X.-H. PHAN, L.-M. NGUYEN, AND S. HORIGUCHI, *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, 2008, pp. 91–100.

[43] A. QAZI AND R. GOUDAR, *An ontology-based term weighting technique for web document categorization*, Procedia computer science, 133 (2018), pp. 75–81.

[44] A. RADFORD, K. NARASIMHAN, T. SALIMANS, AND I. SUTSKEVER, *Improving language understanding by generative pre-training*, 2018.

[45] F. REN AND J. DENG, *Background knowledge based multi-stream neural network for text classification*, Applied Sciences, 8 (2018), p. 2472.

[46] R. K. ROUL, J. K. SAHOO, AND K. ARORA, *Modified tf-idf term weighting strategies for text categorization*, in 2017 14th IEEE India council international conference (INDICON), IEEE, 2017, pp. 1–6.

[47] T. SABBAH, A. SELAMAT, M. H. SELAMAT, F. S. AL-ANZI, E. H. VIEDMA, O. KREJCAR, AND H. FUJITA, *Modified frequency-based term weighting schemes for text classification*, Applied Soft Computing, 58 (2017), pp. 193–206.

[48] M. D. P. SALAS-ZÁRATE, R. VALENCIA-GARCÍA, A. RUIZ-MARTÍNEZ, AND R. COLOMO-PALACIOS, *Feature-based opinion mining in financial news: an ontology-driven approach*, Journal of Information Science, 43 (2017), pp. 458–479.

[49] K. SCHOUTEN, F. FRASINCAR, AND F. DE JONG, *Ontology-enhanced aspect-based sentiment analysis*, in International Conference on Web Engineering, Springer, 2017, pp. 302–320.

[50] M. SCHUSTER AND K. K. PALIWAL, *Bidirectional recurrent neural networks*, IEEE transactions on Signal Processing, 45 (1997), pp. 2673–2681.

[51] R. A. SINOARA, J. CAMACHO-COLLADOS, R. G. ROSSI, R. NAVIGLI, AND S. O. REZENDE, *Knowledge-enhanced document em-*

beddings for text classification, Knowledge-Based Systems, 163 (2019), pp. 955–971.

[52] Y. SONG, H. WANG, Z. WANG, H. LI, AND W. CHEN, *Short text conceptualization using a probabilistic knowledgebase*, in Twenty-Second International Joint Conference on Artificial Intelligence, 2011.

[53] P. THAKOR AND S. SASI, *Ontology-based sentiment analysis process for social media content*, Procedia Computer Science, 53 (2015), pp. 199–207.

[54] R. TÜRKER, *Knowledge-based dataless text categorization*, in European Semantic Web Conference, Springer, 2019, pp. 231–241.

[55] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER, AND I. POLOSUKHIN, *Attention is all you need*, in Advances in neural information processing systems, 2017, pp. 5998–6008.

[56] M. VÖLKEL, M. KRÖTZSCH, D. VRANDECIC, H. HALLER, AND R. STUDER, *Semantic wikipedia*, in Proceedings of the 15th international conference on World Wide Web, 2006, pp. 585–594.

[57] D. WANG AND H. ZHANG, *Inverse-category-frequency based supervised term weighting scheme for text categorization*, arXiv preprint arXiv:1012.2609, (2010).

[58] F. WANG, Z. WANG, Z. LI, AND J.-R. WEN, *Concept-based short text classification and ranking*, in Proceedings of the 23rd ACM Inter-

national Conference on Conference on Information and Knowledge Management, 2014, pp. 1069–1078.

[59] H. WANG, F. ZHANG, X. XIE, AND M. GUO, *Dkn: Deep knowledge-aware network for news recommendation*, in Proceedings of the 2018 world wide web conference, 2018, pp. 1835–1844.

[60] J. WANG, Z. WANG, D. ZHANG, AND J. YAN, *Combining knowledge with deep convolutional neural networks for short text classification.*, in IJCAI, 2017, pp. 2915–2921.

[61] Q. WANG, Z. MAO, B. WANG, AND L. GUO, *Knowledge graph embedding: A survey of approaches and applications*, IEEE Transactions on Knowledge and Data Engineering, 29 (2017), pp. 2724–2743.

[62] Z. WANG AND H. WANG, *Understanding short texts*, (2016).

[63] Z. WANG, H. WANG, AND Z. HU, *Head, modifier, and constraint detection in short texts*, in 2014 IEEE 30th International Conference on Data Engineering, IEEE, 2014, pp. 280–291.

[64] Z. WANG, H. WANG, J.-R. WEN, AND Y. XIAO, *An inference approach to basic level of categorization*, in Proceedings of the 24th acm international on conference on information and knowledge management, 2015, pp. 653–662.

[65] Z. WANG, K. ZHAO, H. WANG, X. MENG, AND J.-R. WEN, *Query understanding through knowledge-based conceptualization*, in Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[66] W. WU, H. LI, H. WANG, AND K. Q. ZHU, *Probase: A probabilistic taxonomy for text understanding*, in Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, 2012, pp. 481–492.

[67] J. XU, Y. CAI, X. WU, X. LEI, Q. HUANG, H.-F. LEUNG, AND Q. LI, *Incorporating context-relevant concepts into convolutional neural networks for short text classification*, Neurocomputing, 386 (2020), pp. 42–53.

[68] N. YADAV AND C. R. CHOWDARY, *Feature based sentiment analysis using a domain ontology*, in Proceedings of the 13th International Conference on Natural Language Processing, 2016, pp. 90–98.

[69] Y. YANG AND J. O. PEDERSEN, *A comparative study on feature selection in text categorization*, in Icml, vol. 97, Nashville, TN, USA, 1997, p. 35.

[70] Z. YANG, D. YANG, C. DYER, X. HE, A. SMOLA, AND E. HOVY, *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, 2016, pp. 1480–1489.

[71] H. ZHANG, Y. CAI, B. ZHU, C. ZHENG, K. YANG, R. C.-W. WONG, AND Q. LI, *Incorporating concept information into term weighting schemes for topic models*, in International Conference on Database Systems for Advanced Applications, Springer, 2020, pp. 227–244.

[72] L. ZHU, G. WANG, AND X. ZOU, *Improved information gain feature selection method for chinese text classification based on word embedding*, in proceedings of the 6th International Conference on Software and Computer Applications, 2017, pp. 72–76.