Research Article

Knowledge Graph-Based Hierarchical Text Semantic Representation

Yongliang Wu, Xiao Pan, Jinghui Li, Shimao Dou, Jiahao Dong, and Dan Wei

School of Information Science and Technology, Shijiazhuang Tiedao University, Shijiazhuang, Hebei 050023, China

Correspondence should be addressed to Yongliang Wu; wuyongliang@stdu.edu.cn

Received 11 January 2023; Revised 11 November 2023; Accepted 12 December 2023; Published 12 January 2024

Academic Editor: Alexander Hošovský

Document representation is the basis of language modeling. Its goal is to turn natural language text that flows into a structured form that can be stored and processed by a computer. The bag-of-words model is used by most of the text-representation methods that are currently available. And yet, they do not consider how phrases are used in the text, which hurts the performance of tasks that use natural language processing later on. Representing the meaning of text by phrases is a promising area of future research, but it is hard to do well because phrases are organized in a hierarchy and mining efficiency is low. In this paper, we put forward a method called hierarchical text semantic representation using the knowledge graph (HTSRKG), which uses syntactic structure features to find hierarchical phrases and knowledge graphs to improve how phrases are evaluated. First, we use CKY and PCFG to build the syntax tree sentence by sentence. Second, we walk through the parse tree using the hierarchical routing process to obtain the mixed phrase semantics in passages. Finally, the introduction of the knowledge graph improves the efficiency of text semantic extraction and the accuracy of text representation. This gives us a solid foundation for tasks involving natural language processing that come after. Extensive testing on actual datasets shows that HTRSKG surpasses baseline approaches with respect to text semantic representation, and the results of a recent benchmarking study support this.

1. Introduction

The topic of natural language processing has seen tremendous advancement and widespread application in recent years, such as text classification [1], machine translation [2], social recommendation [3], and intelligence dialogue [4]. The present state of the art in NLP may be roughly broken down into three distinct phases as follows: text representation, method training, and algorithm evaluation. Each and every activity in natural language processing relies on text representation as its basis. Its purpose is to establish a mechanism for converting natural language with human intent into a structured form that is easy for computers to compare and measure. Researchers have conducted related research on text representation [5, 6]. However, language processing tasks still face significant difficulties because of the semantic variety of natural language texts. As a result, research on text-meaning representation has grown in popularity recently.

Word vectorization is the foundation of most common text representation techniques, and it is the result of the critical word extraction and word vector stages. BOW’s core tenet is that one should see the text as a collection of words and decompose the text in the document into words without considering the order, so it is liable to lose the natural text semantics and affect the performance of downstream tasks. In addition, the words appearing in the text are treated equally, which cannot reflect the importance of words in the sentence, making the semantics of sentences indistinguishable. Researchers have proposed N-gram [7] and TF-IDF [8]. The former improves the disorder problem in BOW, and the latter considers the relative importance of words in the current document and the whole corpus. Word2Vec is a classical word-embedding model that combines the context of words to obtain the vectorization of words [9]. In summary, most traditional text-representation methods employ words as text semantic representation.
units, which segment the phrase semantics and ignore the word order characteristics, resulting in the loss of text semantics.

From the perspective of semantic integrity, phrases are more suitable basic units of textual representation than words [10, 11]. We believe the following two issues can effectively improve the quality of text representation and have intense research significance. First, unlike words, phrases are not naturally segmented in natural language texts and have hierarchical combinatorial features. Therefore, the first issue to be considered is how to obtain high-quality phrases quickly and accurately in continuous text. In previous work, we have explored the topic of phrase embedding in our own studies [10] and used it in the context of question-and-answer systems [12-14], which lays the foundation for our research in this paper. Second, hierarchical phrases slow down the process of selecting suitable phrases and phrase evaluation. Through literature research [15, 16], we find that proprietary terms in various fields are mostly described by phrases, which are often inherent. For this issue, we plan to improve mining efficiency by introducing knowledge graphs into the phrase mining process.

For the issues of semantic loss of word representation and low efficiency of phrase mining, the research motivation of this paper is to solve semantic loss by representing text by phrases and introduce knowledge graphs into the phrase mining process to improve mining efficiency. In this work, we propose HTSRKG, a hierarchic text semantic representation method based on the knowledge graph. First, we construct a syntactic structure tree for each text sentence based on parsing technology, which thoroughly considers syntactic and semantic features. Second, hierarchical recursive traverses the component candidate phrases, which we will use with statistical and combination attributes to determine their overall quality. Third, we introduce the knowledge graph in the phrase mining and evaluation process to optimize phrase semantics. The innovative contributions of this paper are as follows:

(i) We combine grammatical structure and statistical features for hierarchical phrase mining, which improves the quality and efficiency of phrase mining.

(ii) We employ knowledge graphs to optimize the phrase evaluation for the first time, improving text semantic representation’s performance and efficiency.

(iii) Through extensive experiments, we verify that our method is ahead of the comparing baseline methods on downstream tasks.

Section 2 of our paper presents the background literature that informed our study. Section 3 introduces some required knowledge of the following work. Our proposed approach is laid out in great detail in Section 4. Extensive tests on actual datasets are used to confirm our method’s effectiveness in Section 5. Finally, in Section 6, we extend our findings and summarize our study.

2. Related Work

We provide a brief overview of relevant efforts in the three areas of phrase extraction, text representation, and knowledge graph in this section.

2.1. Text Representation

Text representation refers to transforming text data into vectors or matrices convenient for computer processing. It has been extensively used in several practical applications as the primary function of natural language processing, including text classification [17], text matching [18, 19], and information retrieval [20]. Researchers have proposed many text-representation models from different perspectives, which can be divided into three types according to the granularity level as follows: word-based, phrase-based, and sentence-based.

Word-based text-representation methods divide the text into sets of words that provide the basis for subsequent natural language processing tasks. BOW is a typical representative of word-based methods. It regards documents as an unordered set of words and employs count coding to represent texts by vectors. Fu et al. [21] proposed a potential BOW model to generate definitions and determine discrete potential variables’ semantics. Liu et al. [22] fused BOW with historical features for summarization generation tasks. However, counting the number of word occurrences cannot determine word frequency’s importance. The TF-IDF model was suggested by researchers as a solution to this issue [8], whose main thesis is that a word’s significance increases in direct proportion to how often it occurs in a text and decreases in direct proportion to how frequently it occurs in all papers. Yahav et al. [23] utilized TF-IDF to deal with the statistical error caused by the high correlation of the extracted text content to increase the text mining’s accuracy. In addition, some other models are also proposed, such as N-gram [7], which considers the order between words and can extract richer features. However, the word-based methods still have significant limitations because they only consider the word frequency without thoroughly combining the context information and ignore the phrases in the text, which will lead to the loss of some text semantics.

Phrases play an essential role in text representation. Phrase-based text-representation methods have also received extensive attention and performed well in downstream tasks [24]. At present, there are roughly two ideas for phrase-based text representation. The first idea considers that a phrase’s semantics are composed of words, so the adjacent words in the sequence are combined to represent a phrase. In the early days, the N-gram model was used for phrase segmentation [25]. However, although this method is simple, the accuracy and efficiency need to be improved. Researchers also put forward some ideas. Xu et al. [24] proposed a phrase representation generating technique that is attention based and can produce phrase representation from a related token representation. Another idea is to regard phrases as indivisible independent semantic units. Li et al. [26] produced a new phrase embedding by merging the pretrained phrase encoding with the embeddings for the
words that make up the phrase. Liu et al. [27] directed the converter-based model to encode important phrases by feeding it some previous information of significant terms.

Early sentence-based text representation mainly improved on the idea of Word2Vector [28] or used the encoder-decoder model to generate sentence embedding [29]. Later, researchers also tried to combine it with self-supervised learning. Klein and Nabi [30] combined self-contrast and decorrelation goals to promote learning representation through the resulting contrast. Jafariakinabad and Hua [31] proposed a self-monitoring framework for learning sentence structure representation, which mainly focused on the syntactic structure of sentences for explicit representation learning. In addition, the attention process is also used to learn sentence representations. Zhang et al. [32] employed a multiview attention model to learn sentence representation, mainly attention from various angles utilizing several view vectors. With the rise of the pretraining language model, text representation has also entered a new stage. Yan et al. [19] combined contrastive learning to employ unlabelled text and solved the collapse problem of sentence representation in BERT calculation. Tan et al. [33] proposed a semantic perception contrastive learning framework based on BERT, which explored the potential semantic space of sentences and reduced the impact of sentence surface features.

However, compared with the three methods, the phrase-based text representation method is the closest to human understanding of semantics [10, 11]. Because phrases are the representation unit of human language semantics, they use a fixed sequence of words to represent independent semantics. In this way, the word’s polysemy is avoided and does not contain multiple overlapping semantics like a sentence. General text segmentation methods like the N-gram model cannot achieve efficient and accurate phrase recognition. In contrast, phrase mining technology has attracted widespread attention. In this study, we suggest using syntactic parsing to create a sentence parsing tree and hierarchical traversal to identify the component phrases.

2.2. Phrase Extraction. The goal of phrase extraction is to mine phrases of an exorbitant quality from a vast volume of text corpora. In natural text processing, it is one of the most important jobs, and it is used in knowledge capture, information retrieval, classification construction, and topic modeling. The ongoing study may be broken down into two categories as follows: statistically-based research and learning techniques model-based research.

Early phrase-mining techniques based on statistics mainly utilized the TF-IDF model to select key phrases based on the candidate phrase set. Gu et al. [34] combined the phrase span from each document with the attention map to train the prediction model to identify quality phrases. Chen et al. [35] suggested that the meaning presentation of each and every sentence should be broken down into substructures (words as well as phrases) with as much granularity as possible and that frequent substructures should be identified based on their frequency. Subsequently, large-scale deep neural networks have brought new vitality to phrase mining. Wang et al. [36] applied the pointer network to select the token index from the input sequence as the output and then utilized LSTM as the decoder for phrase recognition. Harada and Watanabe [37] adopted syntactic distance calculated by convolution network to induce the phrase structure.

The phrase is one of the critical expressions of text semantics, so efficient and accurate phrase mining becomes essential. However, most phrases are hierarchical and the number is vast, so finding a suitable phrase mining method has become one of the crucial topics in recent years.

2.3. Knowledge Graph. The semantic network is the source of the data structure known as the knowledge graph, which is made of entities, relationships, and characteristics [38], and is represented explicitly as triples (subject predicate object). It collects all kinds of human knowledge and is widely used in diverse domains [15, 16]. Wikidata, YAGO, and DBpedia are just a few examples of the numerous large-scale knowledge graphs that scholars have constructed recently. These knowledge graphs have been widely applied for recommendation [39] and medical [40].

In the last several years, researchers have observed that knowledge graphs are capable of providing rich and organized information for language comprehension. For increasing the efficiency of text representation, semantic identification and word embedding are both implemented with the help of the knowledge graph. Kapanipathi et al. [41] proposed a language knowledge enhancement graph converter (LET) to deal with word ambiguity. Du et al. [42] integrated the knowledge graph into leap LSTM to skip irrelevant words in the input text. With the proposal and development of pretraining models (such as BERT), researchers are constantly exploring the integration of knowledge graphs into language models. He et al. [43] incorporated the previous knowledge of entities obtained from the global existing knowledge into the presentation and made use of the knowledge graph to improve word representation and the performance of recognized named entities. Liu et al. [44] incorporated the triples connected to the input document into the knowledge graph so that the BERT could turn the original sentences into a sentence tree that included the information from the knowledge graph.

On the other hand, as can be shown from the previous investigation and observation, the majority of research studies use knowledge graphs to enhance the effectiveness of entity identification in text. Still, the semantic representation of crucial phrases in text is poor [15, 20]. The main reason is that the entities in the knowledge graph are mostly proper nouns or terms. At the same time, the semantics in actual natural language are often expressed as multilevel phrases. This leads to a representation deviation between entity semantics and precise semantics. For example, “support vector machine” is a term in machine learning. We often mentioned “the parameters of support vector machines” as a semantic whole in our communication process. How to recognize and represent phrases in text effectively and
accurately has become a significant problem. This paper utilizes knowledge graphs to modify phrase structure and improve text semantic representation accuracy.

2.4. Text Classification and Text Clustering. Text classification and clustering are classic downstream tasks of text representation tasks and are usually used to analyze and verify the performance of text-representation methods.

Text classification methods based on text semantic representation usually use words as the basic unit of semantic representation. Xiong et al. [45] employed candidate retrieving and deep ranking to address the issue of the low efficiency of word-based text classification methods. Parmar et al. [46] mainly researched word-based unseen entity text classification and primarily solved the problem of unknown class classification. Zhao et al. [47] proposed a word-based hierarchical capsule network for hierarchical text classification. Wang et al. [48] proposed a word-embedding network for structured text classification to obtain the fusion embedding of hierarchical semantics dependency and graph structure in a structured text and to distill the meta-information from fusion characteristics. To sum up, the existing text classification methods are primarily based on words for semantic representation and phrase semantic units are lost.

Text clustering methods based on semantic representation often employ words as the basic semantic representation unit. Guan et al. [49] proposed a deep feature-based text clustering framework incorporating pretrained word encoders into text clustering tasks. Zheng et al. [50] improved robustness against imbalanced and noisy data based on word representation in text clustering. Saada and Nadif [51] aimed to investigate the impact of different transformations on both the isotropy and the performance based on words to assess the true impact of anisotropy. Cheng et al. [52] targeted fine-tuning the word-based pretrained language models to effectively handle the text clustering task. In short, current text clustering methods are primarily based on the semantic representation of words and the semantics of phrase representation are lost, which affects the clustering performance.

3. Preliminaries

This section shows some indispensable knowledge about our research.

Definition 1. Text representation is a primary task of document mining tasks, and the goal is to find a reasonable way to convert continuous natural language sentences into a structured shape for subsequent machine learning algorithms. Traditional methods usually employ words as the basic unit of text representation. Given a corpus $C$, its representation is $C = \{w_1, w_2, \ldots w_n\}$, where $w_i$ defines words appearing in the corpus.

Definition 2. Phrase is an ordered set of words with fixed semantics. It is represented as $p = w_1w_2\ldots w_n$. From the perspective of semantic formation, phrases include two categories, namely, intrinsic phrases and combined phrases. Inherent phrases are often derived from phrases with artificially determined semantics in specific fields, such as “neural networks” and “knowledge graph.” Combining phrases usually combines the semantics of compositions (phrase or word) to form the final semantics. For example, “minimum spanning tree” and “maximum likelihood estimation.” In this paper, we focus on the hierarchical features of phrases, which indicate that a phrase may be contained within other phrases, such as “artificial intelligence” and “interpretive artificial intelligence.”

Definition 3. Phrase mining aims to extract key phrases from continuous natural language texts to represent the core semantics of the text. Given a corpus $C$, its representation is $C = \{p_1, p_2, \ldots p_n\}$, where $p_i$ defines phrases appearing in the corpus.

Definition 4. Phrase embedding is to design a function that preserves semantic relatedness while mapping statements to a low-dimensional subspace, thus facilitating the quantification of phrase relationships, as shown in Figure 1.

Definition 5. Phrase evaluation is a process of checking the importance of phrases, which includes various indicators, frequency, consistency, completeness, and informativeness. Phrase quality usually ranges from 0 to 1 or more. The more critical a phrase in is a corpus; its quality is closer to 1.

Definition 6. Knowledge graph is a collection of human knowledge consisting of a large number of entity-relation triples, shown as $KG = \{trip_1, trip_2, \ldots trip_n\}$, where $trip_i = \{e_1, r, e_2\}$. $e_1$ denotes a professional entity term. $r$ is the relation of $e_1$ and $e_2$.

4. Hierarchic Semantic Representation by the Knowledge Graph

We introduce HTSRKG, a hierarchy text semantic representing by the knowledge graph, to enhance text semantic representation. The process is broken down into three primary steps as follows: PCFG-based parsing tree construction, hierarchical phrase mining, and a knowledge graph-based text semantic correction stage. In Figure 2, we can see its internal structure. Training text and an expanded knowledge graph are HTSRKG’s inputs. HTSRKG produces a semantically ordered collection of phrases for subsequent tasks in natural language processing.

4.1. PCFG-Based Parsing Tree Construction. To have HTSRKG ready to go, text preprocessing is a crucial step. Unfortunately, NLP activities cannot be performed directly on pure natural language text since it often includes erroneous or filthy data. In order to break natural language documents into sentence units for further syntactic structure analysis, we use the techniques of removing stop word, tokenization, and stemming.
In order to extract meaning from text, it is necessary to analyze its syntactic structure. The two main categories of syntactic parsing techniques, constituent parsing and dependency parsing, describe the most often used approaches. In this work, we use phrases as basic building blocks for a semantic representation of text. As exhibited in Figure 3, we use constituency parsing to express the language semantics by a syntax tree made up of phrases and their respective related classes. The top “ROOT” represents the root node of the syntactic parse tree. The lowest level contains all the words that make up the sentence. The intermediate nodes from “ROOT” to the leaf nodes all represent the part of speech of word combinations, e.g., “EX” indicates a demonstrative pronoun. “VBP” is a present tense verb, not a third-person singular. For other part-of-speech symbols, please refer to Penn TreeBank II (https://surdeanu.cs.arizona.edu/mihai/teaching/ista555-fall13/readings/PennTreebankConstituents.html), and this paper uses this to mine candidate phrases.

The TreeBank dataset is where we get our hands on G, the PCFG (probability-based context grammar) (Penn TreeBank syntax). G shown as \((N, \Sigma, R, S, P)\). Nonterminal symbols are represented by the set \(N\) of Penn TreeBank II component tags’ phrase-level bracket labels served as the basis for \(N\)’s components, including PP, VP, and NP. Since NP serves as a building block for both PP and VP, this work is dedicated to exploring its construction and extraction. Symbols used at the end of a message are denoted by \(\Sigma\). In grammar, carpus words that are not connected to \(N\) are called terminals. For the sake of this discussion, let us refer to \(\Sigma\) as the set of all words included in the Penn TreeBank. Rules for manufacturing are denoted by \(R\). The form \(A \rightarrow B\) of every rule \(r\) is known to exist. \(A\) stands for a node set that is not terminal. \(B\) is created by combining several components of \((\Sigma \cup B)\). To create \(R\), we gathered all of the nonterminal node extensions from the Penn TreeBank. The beginning sign is \(S\). All production rules’ probabilities are included in \(P\). The expression of a rule \(r\) is \(A \rightarrow B[p]\). \(p\) represents the probability of this rule and its range is 0–1. The probability of every nonterminal expansion, when added together, must equal 1, \(\sum_{r \in \mathbb{R}} P(A \rightarrow B) = 1\). Any rule of \(A\)’s expansion is \(p\).

We can determine the likelihood of every criterion in \(R\) with given Penn TreeBank by calculating the occurrences of extension and then formatting, which is illustrated as follows:

\[
P(A \rightarrow B) = \frac{|A \rightarrow B|}{|A|}.
\]

4.2. Hierarchical Phrase Mining. Algorithm 1 uses the concept of hierarchical traversal based on the parse tree it builds to extract the essential phrases from sentences. Text corpus \(C\) has been processed. The \(t\).content refers to the meaning of \(t\). The tag tree/node of tree is expressed by the letter \(t\).able. After parsing a text, we put it into a hierarchy with each node representing a phrase. The subtrees for each node are represented as \(t\).subtree( ) and each node has many of them. Lines 3 through 20 go over each sentence in the corpus one by one. The parse tree’s hierarchical traversal is carried out at lines 6 through 18. The phrase “saving and statistics” is completed in lines 7 through 12. The key stage of hierarchical traversal, which is implemented in line 13 to line
4.3. Knowledge-Based Text Semantic Correction. Text’s phrase composition is then obtained from the grammatical structure of sentences, following the combined phrase mining step. In order to refine the text’s semantic representation, Algorithm 2 uses the knowledge graph for fixing the mined phrases. Algorithm 2 takes as input a quality-sorted set of phrases and a larger knowledge base. We give higher weight to word combinations appearing in the knowledge base because they are high-quality phrases humans have verified. For terms in professional fields that are not included in the knowledge base, we still use syntactic parsing results for mining because, as domain terms, they generally exist as independent syntactic components in sentences, such as subjects and objects. Therefore, they can still be extracted through syntactic analysis. Lines 2–7 prioritize adding knowledge graph-related terms to the updated phrase collection. Lines 8–10 reorder the revised results based on the quality of the remaining sentences. This is why Algorithm 2 makes use of the knowledge graph to help refine the phrase description of the document, thereby keeping the sentence’s semantics more intact.

5. Comparative Experiments and Analysis

5.1. Datasets. We conduct studies to test the efficacy of the knowledge base with improved semantics on document categorization and document aggregation works. This study does so by introducing the following two types of outward collected datasets: knowledge base and document sets with classes.

Many people think of the knowledge base as the repository of human epistemology because it accurately depicts the intricate webs of connections that hold together the entities of the actual world. It is often shown as a triple with three parts, i.e., two things and a connection between them. For instance, “Beckham,” “profession,” “football player” would indicate that football player is Beckham’s line of work. We chose the well-known YAGO and DBpedia knowledge graphs. A massive semantic knowledge base built on WordNet and GeoNames from Wikipedia, YAGO (https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago) is a powerful tool. It comprises almost 120 million relations among more than 10 million entities (such as people, organizations, and cities). For the purpose of extracting structured material from the information provided in different Wikimedia projects, DBpedia (https://www.dbpedia.org/resources/knowledge-graphs/) is an explicit knowledge graph that now comprises over 228 million items. In this study, we do a minimal amount of preprocessing on the aforementioned knowledge graphs and choose a small subset of entity triples to conduct our tests with. Table 1 displays the results of the statistical analysis.

We choose five popular text clustering and classification datasets and undertake downstream comparison tests to evaluate the efficacy of text semantic representation.

There are a total of 2225 documents included in the BBC dataset (https://mlg.ucd.ie/datasets/bbc.html), all of which are related to news articles that appeared on the newspaper’s website in one of the five different categories between 2004 and 2005.

The 20 Newsgroups dataset (https://qwone.com/~jason/20Newsgroups/) contains over 20,000 news texts that have been uniformly distributed among 20 topics. The
Reuters-21578 (https://archive.ics.uci.edu/dataset/137/reuters-21578+text+categorization+collection) is a corpus of articles from the Reuters wire service in 1987 that is often used in linguistics research. The RCV1-RCV2 (https://archive.ics.uci.edu/ml/datasets/Reuters+RCV1+RCV2+Multilingual+Multiview+Text+Categorization+Test+collection) is a text corpus and associated news category data in English, often used in text categorization and other NLP applications. Users’ ratings and comments on various product types sold on Amazon are archived in the Amazon dataset (https://jmcauley.ucsd.edu/data/amazon/index_2014.html). It is widely considered a benchmark for text clustering and classification problems. Table 2 displays data on text collections.

5.2. Experimental Environment Configuration. All of this paper’s experiments are written in Python. Windows with a 32 GB Intel Core i7-10700k CPU serves as the testing platform. This is the basic concept behind how our experiment was built. We maximize the quality of the data by eliminating stop words, stemming, and other common errors in the preparation step after obtaining experimental datasets. In this work, we build upon our prior studies [10]...
by using text clustering and classification jobs to prove the
relevance of knowledge bases in the meaning expression of
expression. To avoid the influence of selection randomness on
the experimental results, we duplicate the abovementioned
eperiments ten times and set the average index as the final
result shown in the latest manuscript based on cross vali-
dation. In each round of experiments, we randomly selected
80% of the experimental dataset as the training dataset. The
remaining 20% is used as the validation set.

5.3. Comparison Methods and Evaluation Indicators. We
conducted split-up tests on text categorization and text
clustering to validate our suggested strategy. We chose four
well-known text categorization methods to use as reference
points in our comparison tests.

- LTW [55] optimizes supervised classification by learning
  a word-based weight function.
- DSA [56] learns word-specific contextual features for
different words to improve text classification performance.
- DCRF [57] implements legal text classification by do-
  main features and random forests.
- LCNHTC [47] is a hierarchical capsule network for the
  hierarchical text classification task, which has the ability to
distinguish similar categories.
- Phrase2Vec [10] utilizes phrases for text semantic rep-
  resentation and improves text classification performance
  based on phrase embedding. It lacks knowledge correction
  in phrase mining and is our preliminary research basis of
  this paper.

We employ precision, recall, and F1-score as the index test.
Precision represents the proportion of genuinely correct sam-
ple among all the samples judged to be accurate, as shown as
(2). Recall represents the probability of the correct example
being judged among all proper examples, denoted as (3). TP
means the number of true positive samples. FP means the count
of false positive samples. TN means the count of true negative
samples. FN is the count of false negative samples. The F1-score
is defined as the proportion of correct classifications as (4).

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (2)
\]
\[
\text{Recall} = \frac{TP}{TP + FN} \quad (3)
\]
\[
F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (4)
\]

5.4. Experimental Analysis of Text Classification. We per-
formed two sets of tests to validate the knowledge graph
representation-based text representation approach.

Text classification algorithms’ experimental results on
many test datasets are displayed in Table 3. The result shows
that the HTSRKG-based text categorization technique is
superior to the baseline approaches. This outcome occurs for
three main reasons. As a first step, HTSRKG uses syntactic
parsing methods to analyze the text’s structural composition.
Second, HTSRKG uses phrases to encode text, conserving
semantic elements as much as possible. Last but not least,
HTSRKG fixes textual hierarchical phrases using the
knowledge graph, enhancing the precision of the text rep-
resentation. While LTW’s technique of learning word
weights enhances text categorization, this approach to
representing texts ignores the structural aspects of sentences,
leading to text semantic loss. While DSA is great at clas-
sifying texts based on the context of individual words, it fails
to take into account that phrases made up of numerous
words may have their own unique semantics. Fusion of
domain characteristics is what DCRF does to make text
categorization more accurate. However, the text categori-
ization performance is hindered by the domain knowledge
base’s reliance on specialist language in place of colloquial
and common terms. LCNHTC solves hierarchical text
classification based on word construction capsule network,
ignoring the semantic characteristics of phrases. Our earlier

In document cluster jobs, four popular baselines are
picked out for comparative experiments. We choose four
popular text-clustering approaches as the gold standards
against which to run our studies.

- HINT [58] employs a heterogeneous information net-
  work combined with multisource text to improve the text
  clustering effect.
- TC-DWA [52] is a novel BERT-based method that
  formulates a self-training objective and enhances text
  clustering with a dual word-level augmentation technique.

Our approach enhances the efficiency of Phrase2Vec-
based text clustering by refining the semantic representation
of text using knowledge graph rectification. Purity is the
evaluation index to measure how well text clusters are
constructed.

\[
Purity(\Omega) = \frac{1}{N} \sum_{\omega \in \Omega} \max_{c \in C}(\omega), \quad (5)
\]

where \(\omega\) is the total volume of samplings in \(\omega\) with the class
\(c\). We may think of \(C\) as the set of all text labels that are
accurate. \(\Omega\) denotes the group of anticipated designations. If
the clustering quality is high, the purity value is near to 1;
else, it is near to 0.

Table 2: Text classification and clustering dataset feature list.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Volume</th>
<th>Text</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBC</td>
<td>2.81 M</td>
<td>2113</td>
<td>5</td>
</tr>
<tr>
<td>20Newsgroups</td>
<td>17.64 M</td>
<td>21000</td>
<td>20</td>
</tr>
<tr>
<td>Reuters-21578</td>
<td>29.3 M</td>
<td>21600</td>
<td>5</td>
</tr>
<tr>
<td>RCV1-RCV2</td>
<td>161.2 M</td>
<td>111920</td>
<td>6</td>
</tr>
<tr>
<td>Amazon</td>
<td>8.4 G</td>
<td>40000000</td>
<td>24</td>
</tr>
</tbody>
</table>
effort, called Phrase2Vec, lacked HTSRKG’s knowledge graph rectification method. Only HTSRKG outperforms it in terms of classification accuracy. HTSRKG is an improvement on the early research on the Phrase2Vec framework. Compared with Phrase2Vec, HTSRKG adds a knowledge base correction stage to the phrase mining module to improve the accuracy and efficiency of phrase mining. HTSRKG pays more attention to extracting hierarchical phrases, and the goal of Phrase2Vec is phrase embedding. Our hypothesis, based on a comparison of the experimental findings of HTSRKG-DBpedia as well as HTSRKG-YAGO, is that the size of the fact graph has correct influences on text classification performance. In summary, HTSRKG uses core phrases to express text semantics and integrates knowledge graphs with semantic rectification. Experiments comparing different methods reveal that HTSRKG yields better results for downstream text categorization.

5.5. Efficiency Analysis of Text Classification. While focusing on the text classification performance of HTSRKG, we also carried out related experiments to verify its running efficiency. Figure 4 demonstrates the text classification runtimes of comparative methods on different datasets. Comparing the overall running time, HTSRKG has certain advantages on different datasets. There are three main reasons for this performance. In the first place, the effectiveness of text representation is greatly enhanced by the phrase mining approach based on syntactic parsing, which efficiently minimizes the amount of candidate phrases. Secondly, the hierarchical traversal method selects the optimal semantic phrases, thereby saving the time required for combined phrase matching. Thirdly, we adopt the knowledge graph to optimize the phrase mining and phrase evaluation process, which shortens the evaluation time of knowledge entities. LTW is word-based for text representation, so it spends much time on subsequent text classification tasks. DSA employs the bag of words for text representation, which consumes much time for text classification. DCRF is aware of the impact of domain terms on operational efficiency, but still ignores the role of combined phrases in text representation efficiency. The word-based capsule network training in LCNHTC takes a lot of time. Phrase2Vec is the basis of HTSRKG, except for the lack of knowledge graph correction stage. Therefore, its running efficiency is second only to HTSRKG. HTSRKG-DBpedia and HTSRKG-YAGO take different amounts of time to complete, but we find that the volume of the knowledge graph makes a big difference in this regard. The time involved grows proportionally with the size of the knowledge graph. Finally, HTSRKG reduces the time needed to classify texts and improves their accuracy by representing them using hierarchy phrases and knowledge networks.

5.6. Experimental Results of Text Clustering. We also perform comparison studies on text clustering activities to confirm the effect of text presentation on subsequent tasks. In Table 4, we can see how various baselines performed on text clustering tasks using various datasets. Compared to other contrastive algorithms, HTSRKG achieves better results when grouping texts. The fundamental rationale is that HTSRKG improves the groundwork for further natural language processing tasks by using the knowledge graph to update phrase-based text representation. HINT improves the performance of text clustering by fusing multisource features through heterogeneous information networks. Semantic similarity calculations are negatively impacted by word-based text representation because they fail to capture the meaning of phrases. Regularized asymmetrical non-negative matrix factorization (RANMF) uses word-to-word similarities to improve text clustering, but it overlooks the importance of phrases in the language representation. When it comes to text clustering, PostSim only takes into account document content similarity, ignoring text semantics and understanding correlation, which might lead to inaccurate conclusions. TC-DWA enhances text clustering based on bidirectional words but still ignores the role of phrases in semantic representation. Phrase2Vec and HTSRKG share the similar text representation pipeline, but HTSRKG refines phrase representation using a knowledge graph. In terms of text clustering, Phrase2Vec is just slightly less effective than HTSRKG. Based on our analysis of the differences between the HTSRKG-DBpedia and HTSRKG-YAGO experimental outcomes, we believe that the knowledge graph selection also plays a role in HTSRKG’s overall performance. According to the results of the preceding investigation, HTSRKG contributes positively to the enhancement of downstream clustering methods.

5.7. Efficiency Analysis of Clustering Task. We execute tests and provide the elapsed time of comparison techniques to confirm the effect of textual encoding on text clustering performance, as shown in Figure 5. Overall, HTSRKG
outperforms the comparison methods in terms of clustering efficiency. There are three factors that play a significant role in this result. First, the phrase mining method based on syntactic parsing saves the filtering time of meaningless candidate phrases, thereby improving the efficiency of text clustering. Second, hierarchical phrases have actual semantics and are less numerous than word-based text-representation methods. Finally, the knowledge graph effectively reduces the mining and evaluation time of inherent phrases in text clustering and improves the overall performance. HINT applies words as the basic unit of text representation, so it consumes more time to handle the text clustering process. RANMF employs regularized asymmetric matrix decomposition to achieve text classification, but the calculation process of word semantic similarity takes much time. PostSim utilizes text similarity to implement text clustering but ignores the promotion effect of knowledge entities. TC-DWA pretrains are based on dual-word integration, which consumes a lot of time. Both HTSRKG and Phrase2Vec use phrase similarity to increase text representation efficiency. However, HTSRKG also makes use of a knowledge graph, whilst Phrase2Vec does not. Based on our analysis of the experimental findings from HTSRKG-DBpedia as well as HTSRKG-YAGO, we hypothesize that a more comprehensive knowledge base is helpful in the textual clustering work. Based on the abovementioned analysis, HTSRKG improves the text clustering effect while maintaining the competitiveness of the running efficiency.

In conclusion, we compared the performance of classification tasks and text clustered on many datasets and analysed the findings in detail. Experiment findings show
that HTSRKG is superior to baseline approaches in text categorization and text clustering and that it also has benefits in text semantic representation.

5.8. Discussion. The core of HTSRKG is to optimize phrase mining efficiency through knowledge graphs for improving downstream natural language processing tasks. Experimental results demonstrate that HTSRKG is ahead of the baseline methods in text classification and clustering. In the practical application process, discussing the following issues will be of particular benefit to future research.

The first issue that needs to be discussed is the selection of knowledge graphs. A knowledge graph is a symbolic representation of a structured semantic knowledge base. It contains descriptions of accepted terms in various domains and thus can assist in phrase mining efficiency. Common knowledge graphs include DBpedia, YAGO, Freebase, BabelNet, Wikidata, and ConceptNet. In this paper, YAGO and DBpedia are selected to participate in the comparative experiment of this paper in combination with the difficulty of acquiring knowledge graphs and the popularity of scientific research. From the perspective of theoretical essence and experimental experience, we believe that any knowledge graph fusion will have better phrase mining results than no knowledge base fusion. In addition, in practical applications, it is better to choose a domain-specific knowledge graph, which we speculate will greatly improve the performance of phrase representation. For example, if the algorithm in this article is applied to the semantic representation of medical texts, BIOS (https://bios.idea.edu.cn/) (a knowledge graph in the medical field) may be a better knowledge base.

Followed by application scenario limitations of the algorithm, the essence of HTSRKG’s phrase mining is based on the syntactic structure. Therefore, it prefers text domains that conform to standard grammar specifications. The performance of HTSRKG may be subject to certain limitations in some application areas where syntactic specifications are low or where syntactic specifications cannot be guaranteed, such as conversation record representation in the field of instant messaging.

Finally, we objectively discuss the deficiencies and limitations of our experiments. Since HTSRKG mainly employs knowledge graphs to modify the syntactic parsing phrase mining process, our experiments mainly focus on datasets that satisfy grammatical specifications. In addition, phrase mining is an upstream of natural language processing tasks. Our experiments verified the promotion effect of HTSRKG on text classification and text clustering tasks. Its impact on other natural language processing tasks needs to be further confirmed.

6. Conclusions

In the current research, we provide HTSRKG, a knowledge-based approach to hierarchical text semantic representation. To begin, we build grammatical parse trees of texts using PCFG and CKY approaches. Second, we get hierarchical representations of texts by walking through the parse tree to
find phrases. We then integrate the knowledge base to improve phrase quality, setting the stage for further natural language processing. The extensive experimental findings show that HTSRKG offers benefits in text semantic representation, suggesting a feasible path for natural document semantics mining.

Large models based on pretraining have been proven to be the most effective language models. However, their pretraining process is mainly based on words (tokens). In future work, we believe that (1) it is attractive to apply phrases in pretraining to improve its efficiency and (2) other phrase-based downstream natural language processing tasks should also be more competitive.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (62106157), the Natural Science Foundation of Hebei Province (F2021210002), the Natural Science Foundation of Hebei Province (F2021210005), and the Outstanding Youth Foundation of Hebei Education Department (BJ2021085).

References


R. Li, Q. Yu, S. Huang, L. Shen, C. Wei, and X. Sun, "Phrase embedding learning from internal and external information based on autoencoder," *Information Processing & Management*, vol. 58, no. 1, Article ID 102422, 2021.


J. Shang, J. Liu, M. Jiang, X. Ren, C. Voss, and J. Han, "Automated phrase mining from massive text corpora," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 437–450, 2019.


