Hindawi
Journal of Applied Mathematics
Volume 2021, Article ID 5264090, 16 pages
https://doi.org/10.1155/2021/5264090

Research Article

Key *n*-Gram Extractions and Analyses of Different Registers Based on Attention Network

Haiyan Wu,[1] **Ying Liu**,[2] **Shaoyun Shi**,[3] **Qingfeng Wu**,[4] **and Yunlong Huang**[5]

[1]*Zhejiang University of Finance and Economics, Hangzhou 310018, China*
[2]*School of Humanities, Tsinghua University, Beijing 100084, China*
[3]*Department of Computer Science and Technology, Institute for Artificial Intelligence, Tsinghua University, Beijing 100084, China*
[4]*Chengdu Polytechnic, Chengdu 610095, China*
[5]*Beijing Normal University, Beijing 100875, China*

Correspondence should be addressed to Ying Liu; yingliu@mail.tsinghua.edu.cn

Received 10 July 2020; Revised 1 December 2020; Accepted 25 April 2021; Published 18 May 2021

Academic Editor: Mariano Torrisi

Key n -gram extraction can be seen as extracting *n*-grams which can distinguish different registers. Keyword (as $n = 1$, 1-gram is the keyword) extraction models are generally carried out from two aspects, the feature extraction and the model design. By summarizing the advantages and disadvantages of existing models, we propose a novel key *n*-gram extraction model "attentive *n*-gram network" (ANN) based on the attention mechanism and multilayer perceptron, in which the attention mechanism scores each *n*-gram in a sentence by mining the internal semantic relationship between words, and their importance is given by the scores. Experimental results on the real corpus show that the key *n*-gram extracted from our model can distinguish a *novel*, *news*, and *text book* very well; the accuracy of our model is significantly higher than the baseline model. Also, we conduct experiments on key *n*-grams extracted from these registers, which turned out to be well clustered. Furthermore, we make some statistical analyses of the results of key *n*-gram extraction. We find that the key *n*-grams extracted by our model are very explanatory in linguistics.

1. Introduction

A register refers to the common vocabulary, sentence structure, rhetorical means, and other characteristics of the language used when people use language to communicate in various social activities for different persons in different environments [1]. With the development of the information age, much register information is produced in production and life. In various Internet applications, the register has played a pivotal role. To better realize automatic text processing, we need to distinguish different registers. As a component of texts, words in the sentence contain rich semantic information, which play very important roles in distinguishing different registers. However, previous studies have demonstrated that *n*-gram based words have better results than words in register classification tasks [2].

Key n -gram extraction can be thought of as extracting *n*-grams to distinguish different registers.

The existing models are mainly based on words, and there are few studies on the extraction of key *n*-grams ($n \geq 2$). Many keyword extraction models have been put forward and have achieved significant effect, due to the development of deep learning models and the attention mechanisms [3–6]. To some extent, each feature extraction model has its advantages and disadvantages.

In terms of keyword extraction, previous scholars mainly proceeded from two aspects, one is the feature extraction and the other is the model design. The features extracted are mainly the word frequency, *term frequency-inverse document frequency* (TF-IDF), *latent Dirichlet allocation* (LDA), synonym set, NP phrases, syntactic information, word2vec, or other specific domain knowledge, such as tags and candidate

keywords [7]. The model design for these features are mainly from three aspects, *statistical language models*, *graph-based models*, and *machine learning models*.

1.1. Statistical Language Models. These models combine linguistic knowledge with statistical methods to extract keywords. Such keyword extraction models are based on the word frequency, POS, lexical chains, *n*-grams, etc. [6, 8, 9]. The advantages of these methods are simple implementation and effective extraction of keywords. Unfortunately, the features chosen by these methods are often based on frequency or countability, without considering the semantic relationship between words and sentences.

These methods lead to some problems, etc.; high-frequency words sometimes are not the keywords, for example, a lot of stop words in linguistics appeared many times (usually, most stop words in Chinese are auxiliary words), but they are not important words for registers. Even though some models select high-frequency words by removing stop words, they are not accurate in the semantic expression of the registers. Especially in a novel with a lot of dialogues in it, we know that in conversation, according to the context, many words are often omitted. If the stop words are removed, the meaning of the sentence is changed completely. Similar problems exist in the features using TF-IDF methods.

1.2. Graph-Based Models. Compared with statistical language models, these models map linguistic features to graph structures, in which words in sentences are represented by nodes in graphs and the relationship between words is represented by edges. Then, the linguistic problems are transformed into graphical problems and the graphical algorithms are applied to feature extraction. In recent years, researchers have tried to use the graphical model to mine keywords in texts. Biswas et al. proposed a method of using collective node weight based on a graph to extract keywords. Their model determined the importance of keywords by calculating the impact parameters, such as centrality, location, and neighborhood strength, and then chose the most important words as the final keywords [10]. Zhai et al. proposed a method to extract keywords, which constructed a bilingual word set and took it as a vertex, using the attributes of Chinese-Vietnamese sentences and bilingual words to construct a hypergraph [11].

These models based on graphs transform abstract sentences into intuitive images for processing and use the graph algorithm to extract the keyword. But the disadvantage is that these algorithms are based on the strong graph theory knowledge, which requires researchers to have strong linguistic knowledge and graph theory knowledge. Only in this way can these two theories be well connected. Besides, the graphs built from the texts usually have thousands or even millions of nodes and relations, which brings efficiency problem to the graph algorithms.

1.3. Machine Learning Models. With the development of the Internet, as the size of the corpus grows larger, there are more and more corpus-based research [12, 13]. It is also an inevitable trend to use a machine learning model to mine its internal laws. Many scholars employed machine learning models to extract keywords. Uzun proposed a method based on Bayesian algorithm to extract keywords according to the frequency and position of words in the training set [14]. Zhang et al. proposed extracting keywords from global context information and local context information by SVM algorithm [15]. Compared with the statistical language models, these early machine learning algorithms based on the word frequency, location information, and global and local context information have made significant improvements in feature extraction. In fact, from the features selected by these models, scholars have tried to consider the selection of features from more aspects. It is just that these features need to be extracted artificially [14–17]. With the development of a computer hardware and the neural network, more complex and efficient models emerged, that is, a deep learning model. Meanwhile, various feature representation methods appeared, such as word2vec and doc2vec. Many scholars began to use deep learning models to extract keywords. Wang and Zhang proposed a method based on a complex combination model, a bi-directional long short-term memory (LSTM) recurrent neural network (RNN), which has achieved outstanding results [3–5]. It can be said that keyword extraction based on a deep learning model not only improved the accuracy of keyword extraction significantly but also enriched the corresponding feature representation. The disadvantage is that the models like the LSTM model has a high requirement for computer hardware and generally needs a long time to train them.

Attention mechanism is proposed by Bahdanau et al. in 2014 [18]. The models with the attention mechanism are widely used in various fields for its transparency and good effects in aggregating a bunch of features. Then, Bahdanau applied the attention mechanism to the machine translation system, which improved the accuracy of their system significantly. In this process, attention mechanism was used to extract the important words in sentences [18]. Pappas and Popescu-Belis proposed a document classification method that applied the attention mechanism to extract words distinguishing different documents, and the classification accuracy rate was greatly improved [19]. The significantly improved classification accuracy implied that the words extracted by the attention mechanism can distinguish different documents well. Similarly, the application of the attention mechanism in other fields also proves this point [20].

By analyzing and summarizing the advantages and disadvantages of these models, we propose a simple and efficient model based on attention mechanism and multilayer perceptron (MLP) to extract key *n*-grams that can distinguish different registers. Here, we call this model the "attentive *n*-gram network"(ANN) for short, whose structure is shown in Figure 1. The model ANN consists of eight parts, the *input layer*, *embedding layer*, *n -gram vector*, *attention layer*, *n -gram sentence vector*, *concatenation*, *classification*, and *output*. In other words, the *input layer* is the sentence we want to classify, the *embedding layer* is to vectorize the words in the sentence, and the *n -gram vector* is to convert the word vector into the corresponding *n*-gram representation. The *attention layer* is to score *n*-grams in the sentence. The *n -gram sentence vector* is a weighted summation of *n*-gram

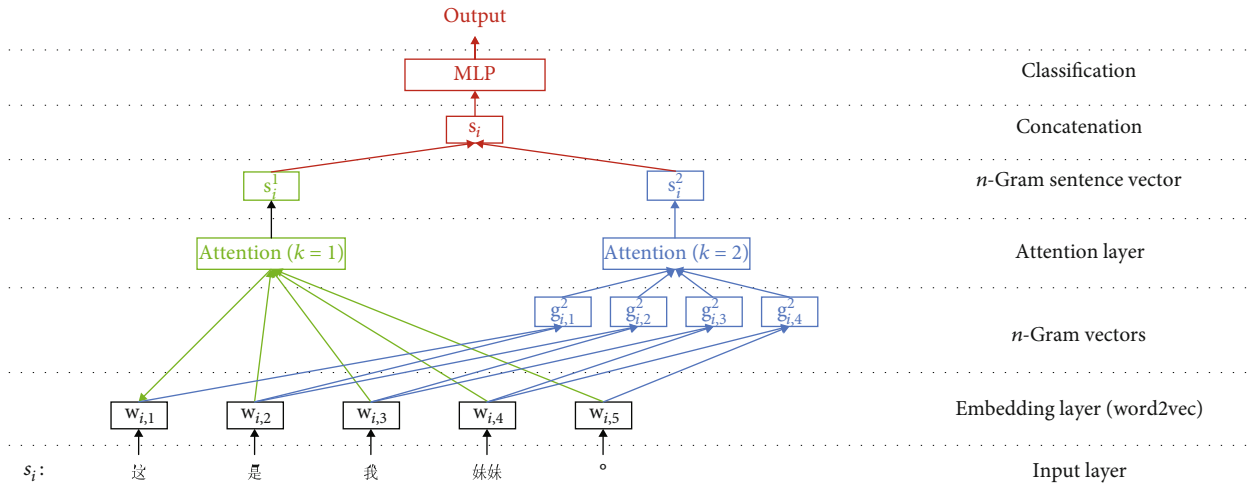


FIGURE 1: The overview of the attentive n-gram network. As an example, the input sentence is $s_i = \{这是我妹妹。\}$. The *embedding layer* converts the words in s_i into the corresponding vector $s_i = \{\mathbf{w}_{i,1}, \mathbf{w}_{i,2}, \mathbf{w}_{i,3}, \mathbf{w}_{i,4}, \mathbf{w}_{i,5}\}$. Then, n *-gram vectors* come from the concatenation of these word vectors. In the figure, as an example, the 2-gram vectors $s_i^2 = \{\mathbf{g}_{j,1}^2, \mathbf{g}_{j,2}^2, \mathbf{g}_{j,3}^2, \mathbf{g}_{i,4}^2\}$ come from the concatenation of every two adjacent word vectors. When $k = 1$, the *attention layer* scores each word $\mathbf{w}_{i,j}$ in the sentence s_i to obtain the score $a_{i,j}^1$. Similarly, when $k = 2$, the score corresponding to each 2-gram $\mathbf{g}_{i,j}^2$ in s_i is $a_{i,j}^2$. Weighting the sum of all $\mathbf{w}_{i,j}(\mathbf{g}_{i,j}^2)$ by the weights $a_{i,j}^1(a_{i,j}^2)$, we obtain the *sentence vector* $\mathbf{s}_i^1(\mathbf{s}_i^2)$. In the *concatenation layer*, $\mathbf{s}_i^1$ and $\mathbf{s}_i^2$ are concatenated together to get the sentence vector $\mathbf{s}_i$, which is the input to the *MLP classifier*. After *classification* layers, we get the *output*, probabilities of the sentence belonging to each category. The symbols are described in detail in Section 2.1. The attentive n-gram network (ANN) structure with 1,2-grams.

vectors to form a sentence vector and the result of the *attention layer*. *Concatenation* concatenates sentence vectors from n-grams with different n as inputs to the classifier. *Classification* is a classifier, and the *output layer* includes three parts, the category of sentences, n-grams, and n-gram-corresponding scores. In Figure 1, we will use an example to illustrate it. Experimental results show that our model ANN achieves significant and consistent improvement comparing with the baseline model. In particular, our work has contributed to the following aspects:

(1) Using attention mechanism to extract key n-grams which can distinguish different registers

(2) Compared with machine learning methods such as SVM and Bayesian, the classification accuracy has been significantly improved by using ANN based on semantic information

(3) With the training process of ANN, attention mechanism has low scores on stop words, which can automatically filter the stop words

2. Methodology

2.1. Attentive n-Gram Network. In computer science, deep learning has become a popular method and has shown its powerful modeling ability in many areas, such as computer vision and natural language processing [21]. Among the basic neural network design patterns, there is a structure called attention mechanism which can automatically analyze the importance of different information. In the field of natural language processing, such as in machine translation, people use the attention mechanism to calculate the source keywords [18].

In our case, the task is to analyze which keywords or 2-gram phrases bear key information for differentiating registers. We first conduct a classification task on texts of different registers and apply the attention mechanism to keywords. Attention mechanism aims to calculate the importance of the words helping to identify the registers, which the higher weights will be assigned to prompt the classification task. Words with higher weights are more important to the register, in contrast to those that appear in every registers, e.g., stop words.

Formally, we can suppose to have a word dictionary $W = \{w_1, w_2, \cdots, w_n\}$ and a set of sentences $S = \{s_1, s_2, \cdots, s_m\}$. Each sentence s_i consists of a sequence of words $s_i = \{w_{i,1}, w_{i,2}, \cdots, w_{i,l_i}\}$, where $l_i = |s_i|$ is the length of the sentence s_i. Here, we highlight vectors in bold. For example, $\mathbf{w}_i, \mathbf{s}_i$ are the vectors of word w_i and the sentence vector of sentence s_i. Word vectors in our model can be randomly initialized or pretrained word2vec, which corresponds to the *embedding layer* in Figure 1.

2.1.1. Attention Mechanism on n-Grams. The attention mechanism in our model takes n-gram vectors as inputs and returns sentence vectors as outputs. In particular, the vectors of k-grams are formed by the concatenation of k word vectors. For example, the sentence $s_i = \{w_{i,1}, w_{i,2}, \cdots, w_{i,l_i}\}$ can also be represented as k-grams $s_i = \{g_{i,1}^k, g_{i,2}^k, \cdots, g_{i,l_{i+1-k}}^k\}$, where $g_{i,j}^k$ is the jth k-gram of the sentence and its vector is $\mathbf{g}_{i,j}^k = \mathbf{w}_{i,j}|\mathbf{w}_{i,j+1}|\cdots|\mathbf{w}_{i,j+k-1}$ ("|" here means concatenation),

n -gram vectors in Figure 1. Then, the attention network first uses a fully connected layer to calculate the latent attention vector $\mathbf{u}_{i,j}^k$ of each k-gram $g_{i,j}^k$:

$$\mathbf{u}_{i,j}^k = \tanh\left(\mathbf{A}^k \mathbf{g}_{i,j}^k + \mathbf{b}^k\right), \tag{1}$$

where $\mathbf{A}^k \in \mathbb{R}^{t \times kv}, \mathbf{b}^k \in \mathbb{R}^t$ are the parameters of the attention network and t denotes the hidden layer size of the attention network and v is the size of the word vectors (kv is the size of k-gram vectors). tanh is the activation function which introduces nonpolynomial factors to the neurons [22]. It has been proved that multilayer feed forward networks with a nonpolynomial activation function can approximate any function [23]. Therefore, it is common that people use a nonpolynomial activation function after a fully connected layer. The $\mathbf{u}_{i,j}^k$ is the hidden attention vector which contains the information of word importance. Then, a weighted sum is conducted over the latent attention vector:

$$u_{i,j}^k = \mathbf{h}^{k^T} \mathbf{u}_{i,j}^k, \tag{2}$$

where $\mathbf{h}^k \in \mathbb{R}^t$ are the weights over the dimensions of $\mathbf{u}_{i,j}^k$ and the parameters of the attention network. The result $u_{i,j}^k$ is the score attention mechanism giving to the k-gram $g_{i,j}^k$. Note that $u_{i,j}^k \in (-\infty, \infty)$, if scores of different k-grams are directly used to do a weighted sum over all word vectors to form the sentence vector, the length and scale of the sentence vectors will be out of control. To normalize the weights, a softmax function is conducted over all $u_{i,j}^k$ (In mathematics, the softmax function, also known as softargmax [24] or normalized exponential function [25], is a function that takes as input a vector of K real numbers and normalizes it into a probability distribution consisting of K probabilities.):

$$a_{i,j}^k = \frac{\exp\left(u_{i,j}^k\right)}{\sum_j \exp\left(u_{i,j}^k\right)}, \tag{3}$$

where $a_{i,j}^k \in (0, 1)$ is the attention weight of the k-gram $g_{i,j}^k$ in sentence s_i. Note that $\sum_j a_{i,j}^k = 1$. Each of the words in the sentence can be scored by equations (1), (2), and (3). For example, $s_i = \{$这是我妹妹。$\}$, in Figure 1, when $k = 1$, the score a_i^1 for each word in sentence s_i is $a_i^1 = \{a_{i,1}^1, a_{i,2}^1, a_{i,3}^1, a_{i,4}^1, a_{i,5}^1\}$. Similarly, when $k = 2$, the score corresponding to each 2-gram in s_i is $a_i^2 = \{a_{i,1}^2, a_{i,2}^2, a_{i,3}^2, a_{i,4}^2\}$. In other words, equations (1), (2), and (3) belong to the attention layer, through which n-gram scores in a sentence are scored, corresponding to the *attention layer* in Figure 1.

The k-gram sentence vector $\mathbf{s}_i^k$ is formed as follows:

$$\mathbf{s}_i^k = \sum_j a_{i,j}^k \mathbf{g}_{i,j}^k. \tag{4}$$

In general, the sentence vector $\mathbf{s}_i^k$ in k-gram comes from a weighted sum of the k-gram vectors $\mathbf{g}_{i,j}^k$. But the weights are dynamically generated through the attention network. Different k-grams have different weights in different sentences. The attention network will learn how to evaluate their importance and will return the weights during the training process.

Specifically, when $k = 1$, the sentence vector $\mathbf{s}_i^1 = \sum_j a_{i,j}^1 \mathbf{w}_{i,j}$ is a weighted sum of word vectors $\mathbf{w}_{i,j}$. To take different n-grams into consideration, we concatenate the sentence vector $\mathbf{s}^k$ in different k. For example, in our further experiments, when considering both words and 2-grams, e.g., ANN (1,2-gram), the final sentence representation is as follows:

$$\mathbf{s}_i = \mathbf{s}_i^1 | \mathbf{s}_i^2. \tag{5}$$

This part corresponds to *concatenation* of Figure 1. The final representations of sentence vectors $\mathbf{s}_i$ are then fed into higher layers for language register classification.

2.1.2. Register Classification. We utilize a multilayer perceptron (MLP)[21] to classify the registers. *MLP* is a kind of feed forward artificial neural network, which consists of at least three layers of nodes: the input layer, the hidden layer, and the output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. *MLP* uses the supervised learning technique called back propagation to train and distinguish different registers. In our paper, this module takes the sentence vector $\mathbf{s}_i$ as inputs and returns the probabilities of s_i coming from different registers as outputs; this part corresponds the *classification* in Figure 1, whose structure is shown in the red part of Figure 1.

Although the task of our model is the classification of registers, what we are really focused on is the different keywords used in different registers, represented by the attention weights. It is not necessary to design a complex or elaborated classification module because what we want is actually a powerful attention network, as mentioned in Section 2.1.1. Supposing that C is the set of all different registers and $|C|$ is the number of classes. In an efficient and effective way, it uses two fully connected layers to build the classification module:

$$\mathbf{p}_i = \mathbf{M}_2 \tanh\left(\mathbf{M}_1 \mathbf{s}_i + \mathbf{b}_1\right) + \mathbf{b}_2, \tag{6}$$

where $\mathbf{M}_1 \in \mathbb{R}^{t \times v_s}, \mathbf{b}_1 \in \mathbb{R}^t, \mathbf{M}_2 \in \mathbb{R}^{|C| \times t}, \mathbf{b}_2 \in \mathbb{R}^{|C|}$ are the model parameters, v_s is the size of sentence vector $\mathbf{s}_i$, and t is the size of the hidden layer. Then, $\mathbf{p}_i$ has the size of $|C|$, which are the unnormalized probabilities of s_i belonging to different registers. To normalize the probabilities, a softmax layer is conducted on the $\mathbf{p}_i$:

$$p\left(c_j \mid s_i\right) = \frac{\exp\left(p_i^{(j)}\right)}{\sum_j \exp\left(p_i^{(j)}\right)}, \tag{7}$$

where $p_i^{(j)}$ is the jth value of $\mathbf{p}_i$ and $p(c_j \mid s_i)$ is the probability of the sentence s_i belonging to the class c_j. To give the

prediction, we let the c_j corresponding to the maximum $p(c_j \mid s_i)$, e.g., $\max_{c_j} p(c_j \mid s_i)$, be the predicted class.

The model is trained through *cross entropy loss* [26], a well-known classification loss in machine learning:

$$\text{loss} = -\sum_{s_i} \log(p(y_i \mid s_i)), \tag{8}$$

where the y_i is the label (real class) of the sentence s_i and the loss function is used to optimize the model. Usually, the closer the loss is to 0, the better the model will be. In our work, we use the Adam function as the optimizer [27].

2.2. Extraction Key n-Gram. After training, ANN gives low weights to all of the n-grams in the sentence because these n-grams are all short common relations frequently occurring in many sentences, which are not representative. Suppose that feature f_i occurs in m documents $d_1, \cdots, d_m$ and its weights in the m documents are $\alpha_i^1, \cdots, \alpha_i^m$, then, the feature importance β_i can be a weighted average of α_i:

$$\beta_i = \frac{1}{m} \sum_j \alpha_i^j \cdot \log_2(l_j), \tag{9}$$

where l_j is the number of input features of document d_j (e.g., number of words when input features are words). It is used to normalize the importance because getting a high weight in a long document is more difficult. Then, the features can be sorted according to the importance β_i and the features with higher importance than a predefined threshold 1.0 are selected.

2.3. Type/Token Ratios (TTR). TTR is the ratio of the type to token in the corpus. The so-called type number refers to how many different words are in the text. The token number refers to how many words are in the text. To some extent, the ratio of the type to token reflects the richness of words of the text. But the TTR calculated in this way is influenced by the length of the article, so here, we use a modified method to calculate TTR, *Herdan's log TTR* [28, 29]. The formula is as follows:

$$\text{Herdan}'\text{s log TTR} = \frac{\log\text{type}}{\log\text{token}}. \tag{10}$$

In our further experiments, we need to calculate the Herdan′s log TTR of different registers to measure their richness, namely, Herdan′s log $\text{TTR}_{\text{novel}}$, Herdan′s log TTR_{news}, and Herdan′s log $\text{TTR}_{\text{text book}}$.

3. Experiment

3.1. Datasets. Experiments and further linguistic analyses are conducted on three corpus datasets. The Novel contains 20 texts, among which 12 are novels written by Mo Yan and 8 novels by Yu Hua. The novels by Mo Yan are *Cotton Fleece*, *Breast and Buttocks*, *Red Sorghum*, *Mangrove*, *Dionysian*, *Life and death Fatigue*, *Thirteen Steps*, *Herbivorous Family*, *41 Guns*, *Sandalwood Penalty*, *Paradise Garlic Song*, and *Frog*. Those by Yu Hua are *Seventh Day*, *Classical Love*, *To Live*, *Reality*, *Brothers*, *Brothers-2*, *Xu Sanguan Sells Blood*, and *Shouting in Drizzle*. In general, Mo Yan and Yu Hua's novels are mainly based on depicting character stories, further revealing the shortcomings behind the social background and the fate of the characters themselves.

The news is a public dataset (https://www.sogou.com/labs/resource/list_yuliao.php), which covers ten topics including domestic, international, sports, social, stock, hot spots, education, health, finance, and real estate. This is a public corpus, and many scholars have used this corpus to do researches.

The textbook mainly includes LuXun's novel "KongYiji", "Hometown", "AQ True Biography", and "Blessing", as well as Lao She's "Camel Xiangzi", Shakespeare's play "Romeo and Juliet", Gorky's "Sea Swallow", "How Steel Is Made", "Honest Children", Zhu Ziqing's "Prose", "Back", "Hurry", "The Analects", etc. It can be seen that the *text book* is a collection of registers, mainly selected by educational articles for students to learn.

The statistics of these datasets are shown in Table 1. Here, the train data and test data are from 0.8 and 0.2 of the *novel*, *news*, and *text book*, which we use to train and test models, respectively. Moreover, to train the model in a better way, we divided the datasets into different proportions, that is, the training set and the test set were 0.7 : 0.3, 0.8 : 0.2, and 0.9 : 0.1, respectively. It is found that the accuracy of the model is as high as shown in Table 2 when the ratio of the training set and the test set is 0.8 : 0.2.

Table 1: Statistics of the dataset.

Dataset	Sentence count	Train data	Validation data	Test data
Novel	117,353	93,882	11,736	11,735
News	29,754	23,803	2,796	2,795
Text book	24,119	19,295	2,412	2,412

Table 2: Classification accuracy based on different models.

Model	Train data	Test data
CNN (1-grams/keywords)	96.45	93.21
CNN (2-grams)	96.27	93.19
CNN (1,2-grams)	96.66	93.85
ANN (1-grams/keywords)	97.85	93.63
ANN (2-grams)	97.04	93.04
ANN (1,2-grams)	98.18	94.88

3.2. Research Procedures. Our experiments are divided into these steps, as shown in Figure 2. Next, we describe each part of the flow chart 2 in detail.

(1) Corpus preprocessing includes the *corpus set*, *preprocessing*, and *corpus vectorization*. *Preprocessing* uses toolkits to clean data and segment words and

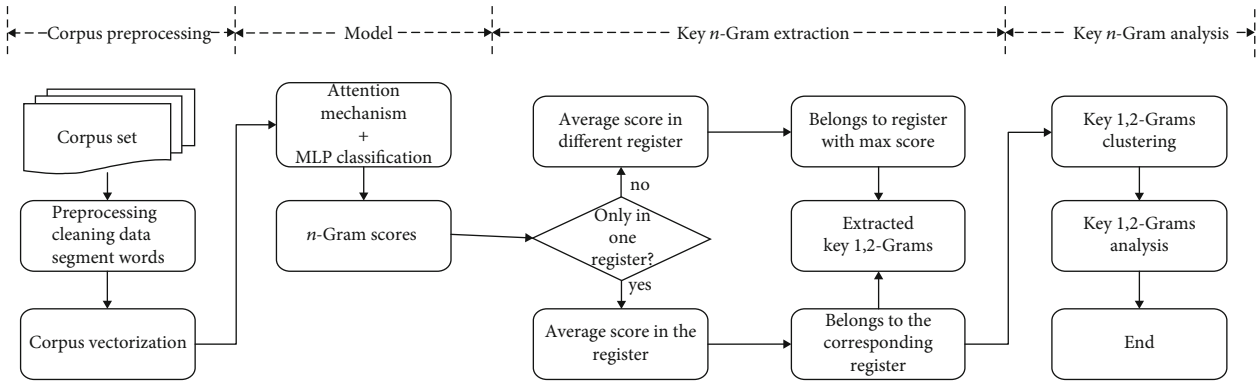
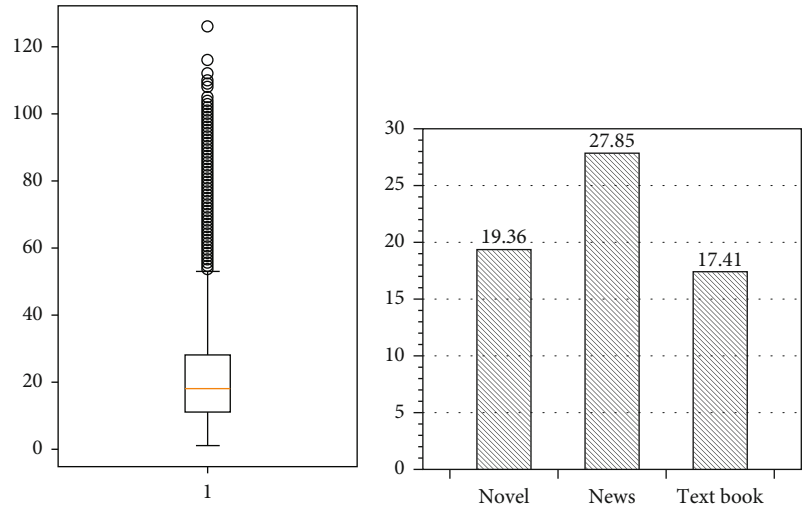


Figure 2: Keyword and 2-gram extraction flow chart.



(a) Boxplot of the sentence length (b) Average sentence length distribution

Figure 3: Sentence length boxplot and average sentence length histogram.

sentences. The main toolkits are *Python3.6* (https://www.python.org/downloads/release/python-360/) and *Stanford CoreNLP* (https://stanfordnlp.github.io/CoreNLP/). *Corpus vectorization* is to translate each sentence into the corresponding word number progressive representation. The result corresponds to the "input layer" shown in Figure 1

(2) The model consists of two parts, *attention mechanism* and *MLP classification*. The function of the *attention mechanism* is to score every n-gram in every sentence by using equations (1), (2), and (3), which are shown in Section 2.1. *MLP classification* is a classifier for stylistic classification. These two parts correspond to the *attention layer* in Figure 1; the working process of these two parts is described in Figure 1

(3) Key n-gram extraction averages the scores of n-gram in each register, whether they appear or occur many times. When n-gram appears in three registers at the same time, the key n-gram with the highest score is regarded as the key n-gram of the register in which it belongs

(4) Key n-gram analyses are composed of *key 1,2-gram clustering* and *key n -gram analyses*, in which *key 1,2-gram clustering* clusters the key n-grams extracted from the previous step. *Key n -gram analyses* not only carries out linguistic analyses on the clustering results of *key 1,2-gram clustering* but also statistical analyses on the key n-grams extracted from the previous step

In addition, the most important task is to find out the key n-grams of each register.

3.3. Experimental Settings. To train our model, we employ the grid search to select the best combination of parameters for the model. These parameters include learning rate $\in\{0.001, \mathbf{0.01}, 0.1, 1, 10\}$ and the batch size $\in\{32, 64, \mathbf{128}, 256, 512\}$. Also, our inputs are the sentence vectors, so we need to set the length of each sentence. According to Figure 3(a), we find that the quarter of the sentence length is 10, the average sentence length is 20, three quarters of the sentence length is 40, and the longest sentence is 128. Since our corpus is composed of three registers, we also calculate the average sentence

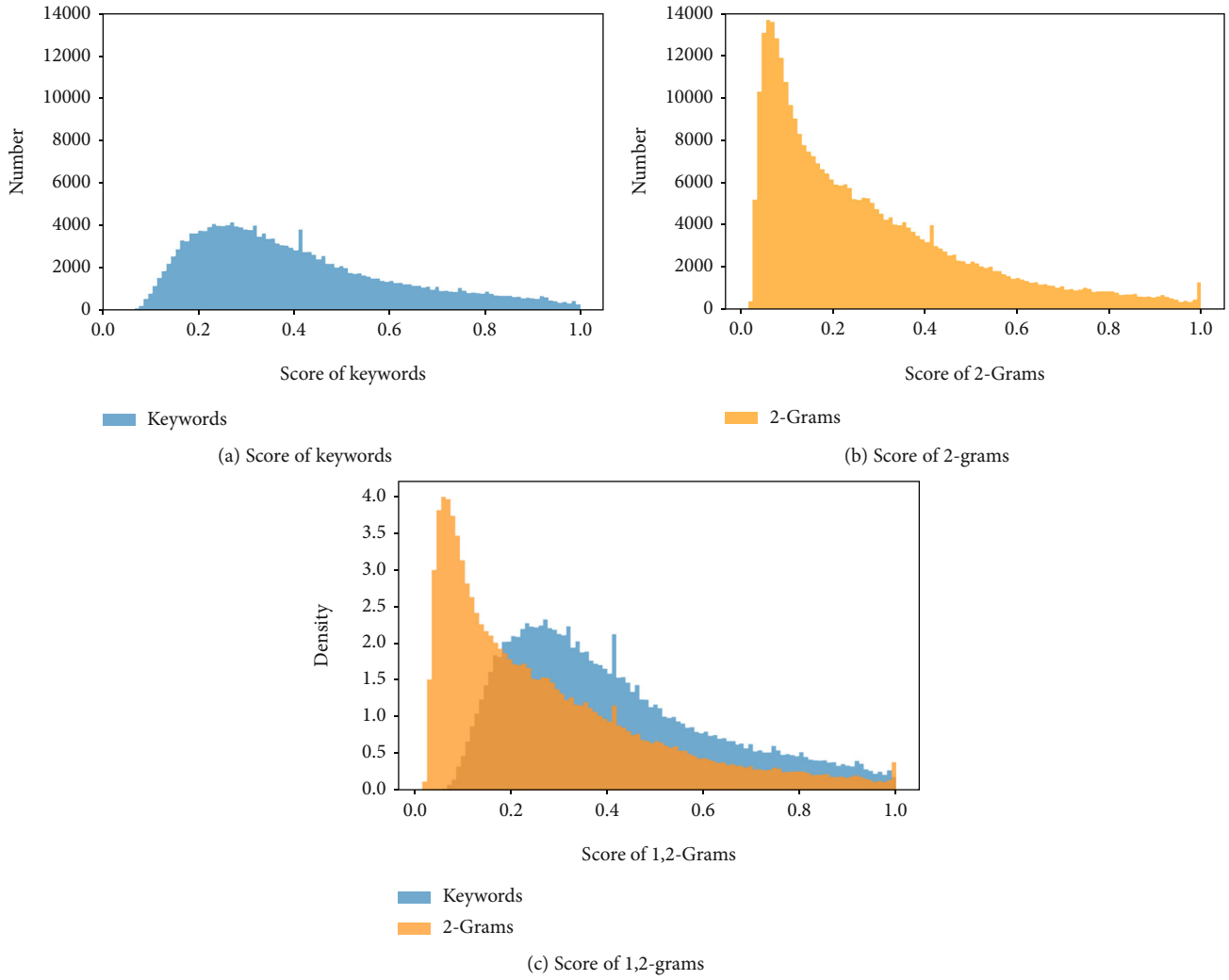


FIGURE 4: Score density interval histogram distribution of the model ANN with keywords and 2-grams.

length of each register, which functions as our reference values to choose the sentence length parameter. From Figure 3(b), the average sentence lengths of the *novel* and *text book* are close to 20 and the average sentence length of the *news* is close to 30. Hence, we set the sentence length set $\in\{10,20,30,40,50,80,100,120,130\}$. The sentence vector size is the total number of words in these three registers; the word size is 32. The best combination of parameters are shown in black bold. For other parameters that have less impacts on our model, we adopt the default values.

To reduce the impact of different sizes of the corpus, we adopt the random sampling method and take corpora of equal size as the training set and test set for our model. Here, we adopt *accuracy* to evaluate our model. *Accuracy* measures how many of the sentences predicted as positive are effectively true positive examples. Besides, when $n = 1$, which means the keyword extraction, we use keywords instead of 1-grams, whose structure is the green part of Figure 1. When $n = 2$, we use 2-grams, which is shown in the blue part of Figure 1.

3.4. Result and Result Analyses. In Figures 4(a) and 4(b), "number" refers to the cumulative number of features in a certain range and "density" means the number of features in a certain interval. From Figure 4(a), we find that the distribution of keyword scores is mainly concentrated in the interval [0.1, 0.5]. In Figure 4(b), the score of 2-gram is distributed in the interval [0.0, 0.4]. In Figure 4(c), in interval [0.3, 1.0], the scores of keywords are greater than that of 2-grams.

Combined with Table 3, we find that in interval [0.0, 0.1], keywords account for 0.0079 and 2-grams account for 0.2543, which indicates that about 0.2543 of 2-grams has little effect on classification. In interval [0.0, 0.3], the keyword stands at 0.3741 and 2-gram accounts for 0.6396; this shows that more than half of 2-grams play a small role in stylistic classification. From the score interval [0.3, 1.0], the proportions of keywords and 2-grams are 0.6259 and 0.3604, respectively, which further show that keywords play a more prominent role in classification than 2-grams.

Furthermore, in Figure 4(c), we find that 2-gram has prominence in the attachment of point 1.0, which indicates

Table 3: Rational distribution of score interval of keywords and 2-grams.

Ratio interval	[0.0,0.1]	[0.1,0.2]	[0.2,0.3]	[0.3,0.4]	[0.4,0.5]	[0.5,0.8]	[0.8,0.9]	[0.9,1.0]
$\text{Ratio}_{\text{keywords}}$	0.0079	0.1438	0.2223	0.1922	0.1473	0.2161	0.0412	0.0291
$\text{Ratio}_{2-\text{grams}}$	0.2543	0.2253	0.1601	0.1169	0.0816	0.1214	0.0221	0.0184

that some 2-grams with higher scores play a good role in stylistic classification.

Our experiments are divided into three groups, namely, *ANN (keywords)*, *ANN (2-grams)*, and ANN (1,2-grams), whose structures consist of the green and the blue parts of Figure 1. Through these models, we can extract the keywords and 2-grams for the *novel*, *news*, and *text nook*. The experimental results are compared with the baseline, and the results are shown in Table 2 on the test set and train data.

3.5. Linguistic Analyses. We mainly analyze the differences among three registers from four aspects, corpus content analyses, lexical richness analyses, keyword analyses, and key 2-gram analyses.

3.5.1. Corpus Content Analyses. To better understand the key *n*-grams of different registers, we mainly analyze the content characteristics of the *novel*, *news*, and *text book*. Their specific statistics are shown in Table 1.

(i) *Novel*. We choose Mo Yan and Yu Hua's novels as our collection of novels. Mo Yan is the 2012 Nobel Prize winner, whose novels often use bold words and colorful words. The sentences in his novels are rich in style, which contains long sentences, compound sentences, and simple sentences, and the author described things in a way that is unconstrained. The language of Yu Hua's novels is profoundly influenced by Western philosophical language, whose traits are simplicity, vividness, fluidity, and dynamic

(ii) *News*. As a register of recording and reporting facts, the *news* usually has several characteristics, such as authenticity, timeliness, and correctness. Authenticity means that the content must be accurate. Timeliness means that the contents is time limited. Correctness means that the reporting of time, place, and characters must be consistent with the facts

(iii) *Text Book*. As a kind of register to impart knowledge to students, the *text book* focuses on training the listening, speaking, reading, and writing skills of students, with the aim of broadening their vision and knowledge scope. So, there are many kinds of articles in a *text book* for students to learn, which mainly contains prose, novels, inspiration, patriotism, ideological and moral education, and other stories

3.5.2. Lexical Richness Analyses. According to equation (10), on the same-size corpus, the higher the TTR value is, the richer the words are. We calculate the TTR of the *novel*, *news*, and *text book*, respectively:

$$\begin{aligned}
\text{Herdan}'\text{s log TTR}_{\text{novel}} &= \frac{\text{logtype}_{\text{novel}}}{\text{logtoken}_{\text{novel}}} = 0.8122,\\
\text{Herdan}'\text{s log TTR}_{\text{news}} &= \frac{\text{logtype}_{\text{news}}}{\text{logtoken}_{\text{news}}} = 0.7574,\\
\text{Herdan}'\text{s log TTR}_{\text{text book}} &= \frac{\text{logtype}_{\text{text book}}}{\text{logtoken}_{\text{text book}}} = 0.7703.
\end{aligned} \tag{11}$$

Since $\text{Herdan}'\text{s log}_{\text{novel}}$ is greater than $\text{Herdan}'\text{s log}_{\text{text book}}$ and greater than $\text{Herdan}'\text{s log}_{\text{news}}$, comparatively speaking, the *novel* has the richest vocabulary, followed by the *text book*, and the *News*.

3.5.3. Keyword Analyses. We analyze the differences among the *novel*, *news*, and *text book* from the proportion of POS and syllable. The statistical distribution of POS is shown in Figure 5(a) and the proportion of syllable distribution shown in Figure 5(b). The data in Figure 5(a) and 5(b) are based on a training set and test set. In Figure 5(a), we can obtain POS in each register from high to low as follows:

(i) *Novel*. *NN*, *VV*, *NR*, *AD*, *JJ*, *VA*, *SP*, *CD*, *OD*, and *CS*.

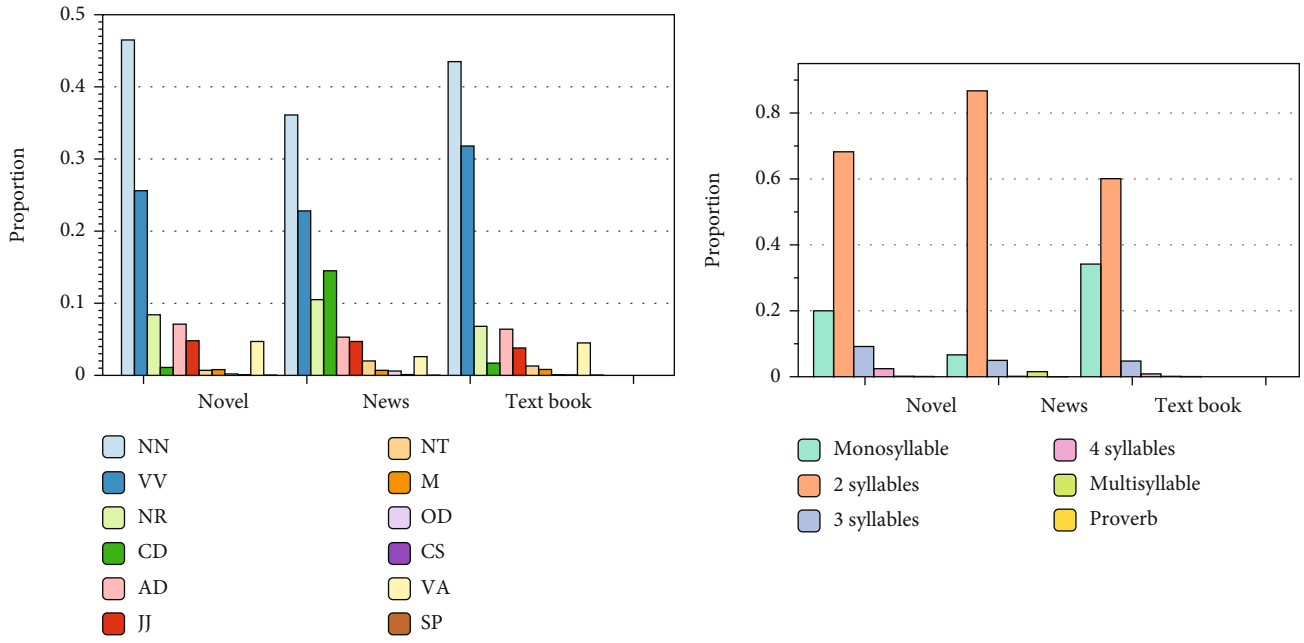
(ii) *News*. *NN*, *VV*, *CD*, *NR*, *AD*, *JJ*, *VA*, *NT*, *M*, and *OD*

(iii) *Text Book*. *NN*, *VV*, *NR*, *AD*, *JJ*, *VA*, *CD*, and *SP*

In Figure 5, we find that the number of nouns (NN) in each register is the highest. To better analyze, we subdivide nouns (NN) into small parts according to their semantic information shown in Table 4.

The specific meanings of abbreviations in Tables 5–7 are given in Tables 4 and 8. The abbreviations in Table 4 are designed by ourselves and the contents of Table 8 are from the Chinese Treebank Marker of Pennsylvania [30]. We will analyze the distribution of each POS in the *novel*, *news*, and *text book*. Take the following POS as examples, as follows.

(1) *POS-NN*. In Figure 5(a), we find that the proportion of nouns (NN) ranks the highest in the *novel*, then in the *text nook*, and the lowest in the *news*. Combined with Table 5, we find that there are more than 12 kinds of nouns (NN) in the *novel*, such as NPE, RN, PA, GN, TN, BN, and EN. These nouns (NN) refer to characters, events, time, descriptions, etc., which correspond to the content characteristics of Section 3.5.1. In *news*, these nouns (NN) mainly include NT, NPE, PSC, RN, OR, CN, etc., which are the names of the main organization, domain noun,



(a) Keyword POS distribution

(b) Keyword syllable distribution

Figure 5: Keyword POS and syllable distribution.

Table 4: Abbreviation list.

Abbreviation	Implication	Abbreviation	Implication
AFPP	Address for a person with physical defect	NOI	Name of the industry
BN	Body noun	NPE	Name of people
CCN	Company character name	NPN	Natural phenomenon noun
CN	Country name	NRNL	Noun related to natural landscape
DSN	Domain scope name	NRTU	Noun related to the universe
EN	Event name	OR	Organization name
FD	Functional department	PA	People appellation
GN	Group name	PRT	Price related to term
LN	Location name	PSC	Production supply chains
MUD	Money used to develop economy	RN	Role name
NOA	Name of the animal	TN	Thing name
NOC	Name of the company	TRO	The ratio of one price to another

occupations, time, group organization, etc., which are from Table 6. Therefore, we find that the *news* focuses on a wide range of groups, not individuals. These nouns (NN) in the *text nook* are names, time, plants, animals, events, natural phenomena, etc.; the specific examples of these abbreviations are shown in Table 7. Hence, we find that textbooks focus on describing people, things, etc.

(2) POS-VV. In Figure 5(a), verbs (VV) are the most in the *text book*, followed by the *novel*, and the last by the *news*. Combining Tables 5–7, we find that verbs (VV) in the *novel* are mainly body-related verbs, such as such as "笑" (laugh), "哭" (cry), "跑" (run), "走" (walk), "跳" (jump), and "唱" (sing). Among them, "说" (say) and "问" (ask) are related to the mouth, "走" (walk) and "跑" (run) are related to the feet, and "抱" (embrace) to the hands; this is related to the characteristics of the *novel*. In the *news*, verbs (VV) are mainly dummy verbs and continuous verbs. For example, "进行" (do) is a dummy verb and "上涨" (go up) is a continuous verb. *News* uses these verbs to express its formal and solemn tone. *Text book* includes not only body verbs but also personalized verbs; the latter are rich in the *text book* because of the wide range of registers selected in the *text book*

(3) *POS-CD*. Also, in Figure 5, we find that *CD* is the most in the *news*, followed by the *text book*, and the *Novel*. As demonstrated in Table 6, we find that there are a lot of numbers in the *news*. It can be said that quantitative figures are used in the *news* to express

Table 5: Novel keyword list (part).

Category	Examples
NPE	“许三观” (Xu Sanguan), “李光头” (Li Guangtou), “福贵” (Fugui), “余占鳌” (Yu Zhan'ao)
RN	“上官” (Shangguan), “司令” (Commander), “铁匠” (Blacksmith), “军官” (Officer)
PA	“大哥” (Brother), “母亲” (Mother), “奶奶” (Grandma), “表哥” (Cousin)
GN	“老人们” (Old people), “姑娘” (Girl), “男人” (Men), “小孩” (Children)
TN	“木筏” (Rafts), “酒” (Wine), “乌云” (Dark clouds), “石桥” (Stone bridge), “小船” (Boat)
BN	“鼻子” (Nose), “手指头” (Finger), “脚” (Feet), “尾巴” (Tail), “眉” (Brow), “眼窝” (Eye socket)
AFPP	“瞎子” (Blind man), “哑巴” (Mute), “瘸子” (Crippled)
EN	“整容” (Cosmetic surgery), “打猎” (Hunting), “自杀” (Suicide), “接生” (Deliver)
NT	“后半夜” (After Midnight), “晌午” (Midday), “傍晚” (Nightfall), “明早” (Tomorrow Morning)
ON	“砰砰砰” (Bam bam bam), “咕咕” (Coo), “吱吱” (Squeak), “咔咔” (Crack)
PN	“哪儿” (Where), “咱们” (We), “大家” (We all), “他们” (They)
AD	“为什么” (Why), “何况” (Moreover), “怎么” (How), “可是” (But)
VA	“通红” (Very red), “丰满” (Fullness), “零零碎碎” (Odds and ends), “活泼” (Lively)
VV	“笑” (Laugh), “哭” (Cry), “跑” (Run), “走” (Walk), “跳” (Jump), “唱” (Sing)
NR	“东北” (Northeast), “墨水河” (The Moshui River), “刘吹手” (Liu blow hand)
JJ	“巨大” (Enormous), “黑暗” (Darkness), “自由” (Freedom), “嘶哑” (Hoarse)
CD	“一半” (Half), “大量” (Lots of), “几” (Some), “十几” (About dozen)
SP	“呢” (Ne), “吧” (Ba), “啦” (La), “嘛” (Ma), “呀” (Ya), “了” (Le), “的” (De)

Table 6: News keyword list (part).

Category	Examples
NT	“2004 年” (2004 Years), “目前” (Li Guangtou), “80 年代” (80s), “5 月” (May)
NPE	“莫扎特” (Mozart), “加尔布雷斯” (Galbraith), “贾冬梅” (Jia Dongmei)
PSC	“投资者” (Investor), “消费者” (Customer.), “生产商” (Producer), “开发商” (Developers)
RN	“分析师” (Analyst), “ 记者” (Reporter), “学者” (Scholar), “老师” (Teacher)
OR	“商场” (Marketplace), “超市” (Girl), “ 公司” (Company), “银行” (Bank)
FD	“国储局” (State bureau), ““财政部” (Treasury), “税务局” (Tax bureau), “国务院” (State Department)
TRO	“投资率” (Investment rate), “汇率” (Exchange rate), “利率” (Interest rate)
NOC	“新浪” (Sina), “谷歌” (Google), “迪奥” (Dior), “腾讯” (Tencent)
CN	“新加坡” (Singapore), “印度” (India), “英国” (United Kingdom), “中国” (China)
CCN	“员工” (Employee), “经理” (Manager), “总监” (Majordomo), “总裁” (CEO)
PRT	“股价” (Share price), “房价” (House price), “油价” (Oil price)
DSN	“经济” (Economy), “政治” (Politics), “环境” (Environment), “文化” (Culture)
NOI	“制造业” (Manufacturing), “房地产业” (Estate industry), “农业” (Agriculture)
M	“种” (Kind), “吨” (Tons), “米” (Meters), “名” (Ming), “平方公里” (Square kilometres)
MUD	“现金” (Cash), “薪水” (Salary), “工资” (Wages), “资金” (Funds)
VV	“表决” (Vote), “上涨” (Go up), “授权” (Authorized), “进行” (Do)
VA	“有效” (Effective), “一致” (Unanimous), “快速” (Fast), “发达” (Developed)
AD	“正式” (Official), “共同” (Common), “最终” (Ultimately), “全部” (All)
CD	“4.88 亿” (488 million), “5%” (Five percent), “70.69” (Seventy point six nine)
JJ	“宏观” (Macroscopical), “持续” (Continue), “全面” (Overall), “强力” (Power)

what is mentioned, rather than vague words, such as approximate grade words, “一半” (half) and “大量” (lots of), which can be often found in the *novel* and *text book*. In addition, the correctness of the *news* is also reflected in using a large number of numerals

In Figure 5(b), we find that the distribution of syllabic words in different registers ranges from high to low as follows:

(i) *Novel*. 2 syllables, 4 syllables, 3 syllables, monosyllable, multisyllable

Table 7: Text book keyword list (Part).

Category	Examples
NPE	“阿Q” (ah Q), “列宁” (LieNin), “喜儿” (Xier), “罗密欧” (Romeo), “闰土” (RunTu)
TN	“山村” (Mountain village), “斗笠” (Straw hat), “树”(Tree)
NOA	“蜜蜂” (Bee), “黄鹂” (Orioles), “兔子” (Rabbit)
NRTU	“极光” (Aurora), “光速” (Speed of light), “宇宙” (Universe), “星系” (galaxy)
NRNL	“高原” (Hibernation), “死海” (Splash), “河” (Peristalsis)
NPN	“雨” (Hibernation), “暴风雨” (Rainstorm), “乌云” (Peristalsis),“风沙” (Wind sand)
NPE	“爱迪生” (Edison), “牛顿” (Newton), “董存瑞” (Dong Cunrui)
RN	“天文学家” (Astronomer), “科学家” (Scientist), “哲学家” (Philosopher)
LN	“巴黎” (Paris), “三峡” (Three Gorges), “鲁镇” (Lu town)
NT	“古代” (Ancient times), “冬天” (Winter), “迄今” (So fa)
AD	“因而” (Thus), “此外” (Besides), “然而” (However), “其次” (Secondly)
CS	“倘若” (IF), “尽管” (Although), “要是” (If only)
VV	“听见” (Hear), “飞溅” (Splash), “蠕动” (Peristalsis), “说” (Say)

Table 8: POS from Penn Treebank.

POS	Implication	POS	Implication
AD	Adverb	NT	Temporal noun
CD	Cardinal number	OD	Ordinal number
CS	Subordinating conjunction	ON	Onomatopoeia
JJ	Noun modifier other than nouns	SP	Sentence final particle
M	Measure word (including classifiers)	VA	Predicative adjective
NN	Common nouns	VV	Verbs
PN	Pronouns		

(ii) *News*. 2 syllables, 4 syllables, monosyllable, 3 syllables, multisyllable

(iii) *Text Book*. 2 syllables, 4 syllables, 3 syllables, monosyllable, multisyllable

We analyze the distribution of each syllable in each register, taking these syllables as examples, as follows:

(1) *Monosyllable*. In Figure 5(b), monosyllabic words are the most in the *novel*, followed by the *text book*, and the *news*. As shown in Tables 5–7, we find that most of the monosyllabic words in the *novel* are body-related words. These verbs are related to specific parts of the body. According to the content of the *novel* in Section 3.5.1, we know that it is consistent with the characteristics of the *novel* which mainly depicts the specific actions of the characters. In the *text book*, because there are many novels, there are more monosyllables in the *text book*. With the simplification of Chinese phonetics, homonyms have significantly increased. If monosyllabic words are still widely used in the *news*, it will inevitably lead to misunderstanding, which hinders the role of language as a tool. Therefore, more accurate polysyllabic words are used in the *news*

(2) *2 Syllables*. In Figure 5(b), we find that disyllabic words are the most frequent in the *news*, followed by the *text book*, and the last by the *novel*. Combined with Tables 5–7, we find that the *news* uses disyllabic words to express a formal and solemn tone. For example, “表决” (vote), “申明” (instruction), etc., instead of “说” (say) in the *novel* and *text book*. In addition, there are more disyllabic verbs in the *news*, in the *novel*, and in the *text book*; disyllabic words are mostly nouns (NN), such as “鼻子” (nose) and “眼窝” (eye socket).

3.5.4. Key 2-Gram Analyses. In Figure 6, we can conclude that the main 2-gram structure of each register is from high to low, as follows:

(i) *Novel*. $NN/NR + VV$, $^{***} + SP$, $JJ/VA + DEG/NN$, $VV + AS$, $DT + NN$, $PN/NN/NR + NN$, $NN/NR + VA$, $CD + M$, and *PROVERB* (http://library.umac.mo/ebooks/b26028347.pdf)

(ii) *News*. $PN/NN/NR + NN$, $VV + NN$, $NN/NR + VV$, $VV + VV$, $VV + CD$, $AD + VV$, $CD + M$, $NT + NN$, $NT + VV$, and $NT + NT$

(iii) *Text Book*. $NN/NR + VV$, $^{***} + SP$, $NN/NR + VA$, $JJ/VA + DEG/NN$, $AD + VV$, $CC + NT$, $VV + AS$, and $P + VV/NN$

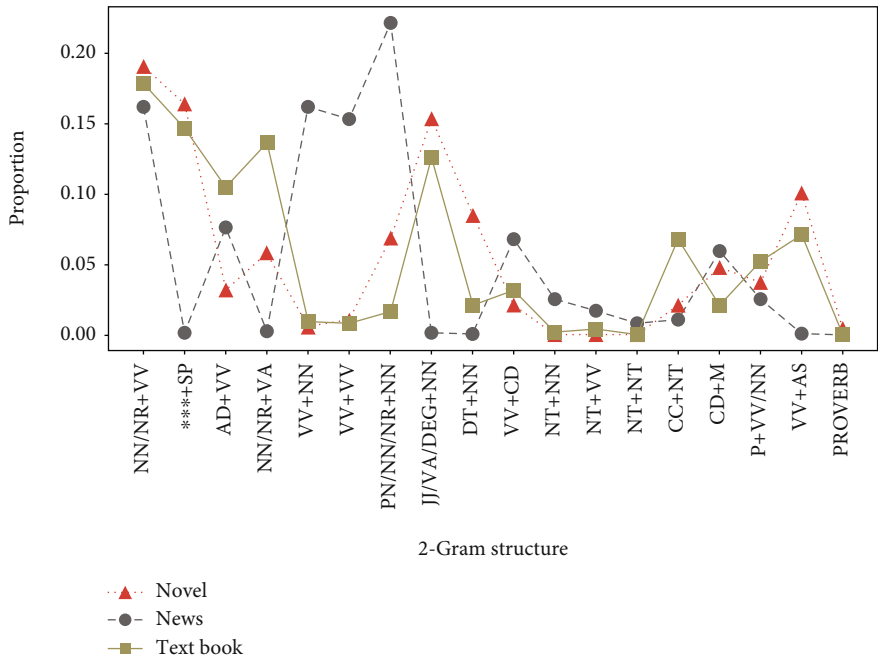


Figure 6: 2-gram structure proportion distribution.

Table 9: Novel 2-gram list (part).

Structure	Examples
NN/NR + VV	“母亲抱出” (Mothers hold out), “屁股扭了扭” (Ass twisted)
	“淤泥吞噬” (Sludge Swallowing), “池塘倾斜” (Pool slanting)
	“身体抖动” (body shaking), “余占鳌说” (Yu Zhanxi said)
*** + SP	“走啊” (Walk), “是吗” (yes?), “是吗” (Wrong!), “黄瞳呢” (Huang Tong?)
AD + VV	“不可以” (Can not), “当场昏倒” (Collapsed on the spot), “快跑” (Run fast)
NN/NR + VA	“河水清澈” (Water clear), “姿态妩媚” (Charming gesture), “夜晚宁静” (Quiet at night)
VV + NN	“翻白眼” (Show the whites of one’s eyes), “截住它” (Stop it)
PN + NN	“我奶奶” (My Grandma), “你公公” (Your father-in-law), “她父亲” (Her father)
JJ/VA + DEG/NN	“严肃的” (Serious), “湿淋淋的“ (Wet), “慈善面孔” (Philanthropy face)
VE + NN	“没有纸灰” (No powder), “有相貌” (Have Face), “无意义” (Meaningless)
VV + AS	“跑了” (Ran), “醒了” (Wake up), “饱了” (Full)
NN + NN	“天黑情景” (Dark scene), “柳生心中” (Speed of)
DT + NN	“这酒” (This wine), “哪个鹞子” (Which kites)
PROVERB	“真金不怕火炼” (True gold fears not the fire)
	“癞蛤蟆想吃天鹅肉” (Toad wants to swallow a swan)

Here, *** + *SP* denotes a sentence or a phrase ending with sp. Combining Tables 9–11, we analyze the distribution of each 2-gram structure in different registers, mainly taking these 2-gram structures as examples, as follows:

(1) $NN/NR + VV$. In Figure 6, we find that the proportion of this structure is the highest in the *novel* and *text book* and the lowest in the *news*. In the *novel*, the examples of the structure $NN/NR + VV$ are shown in Table 9. Combining Section 3.5.1, we find that the *novel* contains many dialogues. For the *text book*, some novels were selected in it, so there are also many conversations. Referring to Tables 9 and 11, structural $NN/NR + VV$ can be regarded as a description of the action and behavior of NN or NR, etc. In Section 3.5.3, we know that the verbs (VV) in structure $NN/NR + VV$ are body-related verbs, which are consistent with the characteristics of the *novel* that mainly describes the action of the characters. Contrary to this, in the *news*, there are many dummy verbs, such as “进行” (do) shown in Table 10. The reason for using such dummy verbs in the *news* is that these verbs are consistent with its own serious register. For the *text book*, as there are

Table 10: News 2-gram list (part).

Structure	Examples
NN/NR + VV	“委托人申明” (Principal claim), “陈渝表示” (Chen Yu said), “他认为” (Principal claim)
PN/NN/NR + NN	“市场经济” (Market economy), “政治热点” (Political hotspot), “试点阶段” (Pilot stage)
VV + VV	“发布实施” (Publish and implement), “参与讨论” (Participate and discuss), “组织执行”(organize and carry out)
VV + CD	“上涨20.26%” (Go up 20.26%), “下降45%” (Descend 45%), “获得2.5” (Achieve 2.5)
VV/VC + OD	“成为第三” (Become a third), “是第一” (Is first), “排位第一” (Pole position)
VV + NN	“表决结果” (Vote result), “进行产销” (Producing production and sales)
AD + VV	“特此公告” (Announcement), “尚未裁决” (Not verdict), “正式发布” (Official release)
NT + NN	“近期价格” (Recent prices), “目前公司” (At present the company)
NT + VV	“2005 年实现” (2005 year realize), “10 日推出” (10, launch)
NT + NT	“2005 年12 月” (December 2005), “5 月10 日” (On May 10)

Table 11: Text book 2-gram list (part).

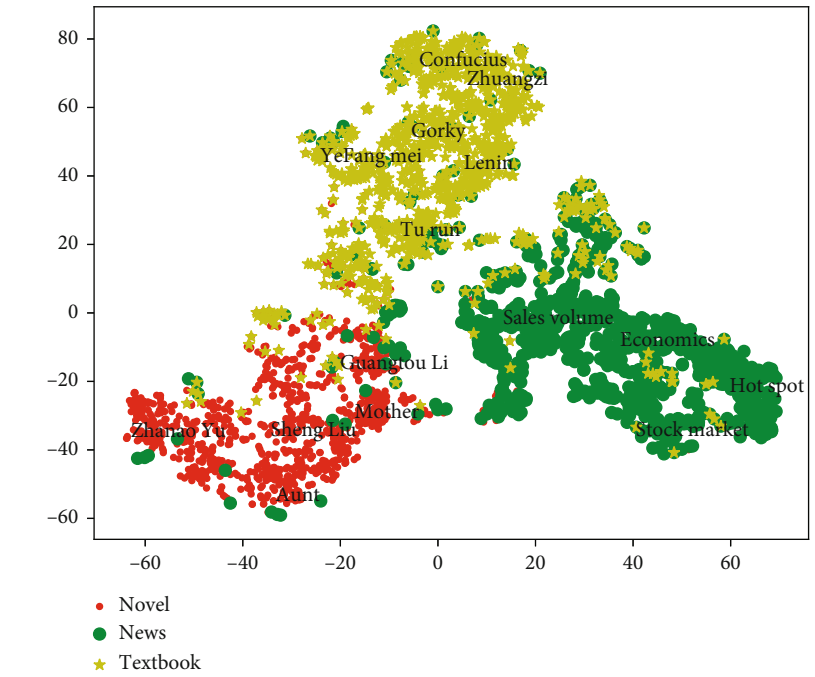
Structure	Examples
NN/NR + VV	“皇帝想” (Emperor wanted), “门铃响” (Doorbell rang), “咱们走” (Let us go)
*** + SP	“对嘛” (That’s right?), “无耻啊” (Shameless!), “后来呢” (Later?)
AD + VV	“很明白” (Understand), “没法说” (Cannot say),“再变旧” (Become old again)
NN/NR/AD + VA	“颜色鲜艳” (Color bright), “小姑娘腼腆” (Little girl shy) “脸色惨白” (Pale face), “感情激动” (Emotional excited)
VV + NN	“拥抱我” (Hug me), “帮帮我” (Help me), “救救孩子” (Save children)
PN/NN/NR + NN	“杜教授” (Professor du), “你父亲” (Your father), “树木绿草” (Trees, green grass)
CC + NT	“或者星期一” (Or on Monday), “和废水” (And waste water), “还是工作” (Or work)
JJ/VA + DEG/NN	“老人家” (Old man), “好主意” (Good idea), “旧衬衣” (Old shirt), “严格的” (Strict)
CD + M	“十一点” (Eleven o’clock), “一粒” (A grain of), “很多根” (Many roots)
P + VV/NN	“往下说” (Go ahead), “像丛林” (Like a jungle)
DT + NN	“哪个混蛋” (Which bastard), “有些园林” (Some gardens)

a lot of novels in it, the structure $NN/NR + VV$ is the same as that in the *novel*

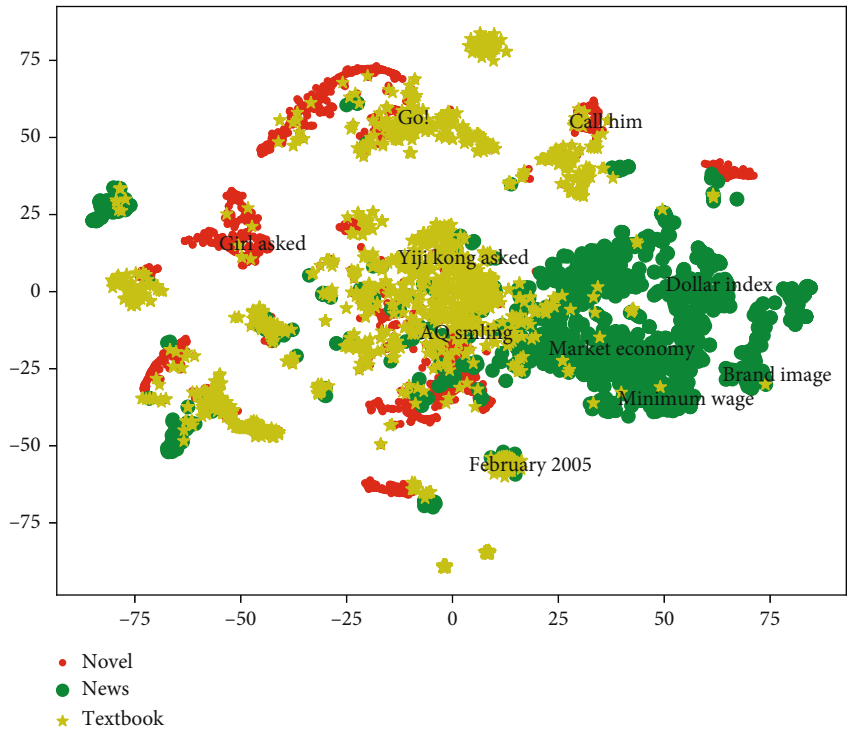
(2) $PN/NR/NN + NN$. In Figure 6, we find that the structure $PN/NR/NN + NN$ is the most in the *news*, then in the *novel*, and in the *text book*. In conjunction with Table 10, we find that the examples of structure $PN/NR/NN + NN$ are composed of two disyllabic words, such as “市场经济” (market economy) and “试点阶段” (pilot stage). Wang and Zhang once pointed out that such a structure of “disyllabic words + disyllabic words” has pan-temporal characteristics. That is to say, the structure of “disyllabic words + disyllabic words” is widely used in the *news*, which can describe things in a more accurate way from a higher angle [31]. This is in line with the characteristics of the *news*. Therefore, there are a large number of such two-syllable structure used in the *news*

(3) $NN/NR + VA$. In Figure 6, the structure $NN/NR + VA$ is the most in the *text book*, followed by the *novel*, and last by the *news*. From the perspective of the whole content of Table 11, comparing with the *news*, we find that the *text book* description is more meticulous, such as “脸色惨白” (pale face), “旧衬衣” (old shirt), “严格的” (strict), and “哪个混蛋” (which bastard), which are shown in Table 11. Combining with the contents of the *text book* in Section 3.5.1, the trait of the *text book* is the more meticulous kind of descriptions, which can better help students learn and improve their writing ability. Besides, as a formal written language, the *news* is simple and serious. The language of the *novel* and *text book* are more casual and flexible; therefore, in the dialogue between the *novel* and the *text book*, this kind of structure often appears and the noun of the structure is always omitted

3.6. Cluster Verification. To verify the effect of our extracted keywords and 2-grams, we use the *t*-SNE [32] method to cluster keywords and 2-grams. The input of *t*-SNE is *n*-grams trained by the attention network, which are high-dimensional vectors. Compared with other clustering



(a) Keyword cluster distribution



(b) 2-gram cluster distribution

Figure 7: Keyword and 2-gram clustering based on *t*-SNE algorithm.

methods, the *t*-SNE clustering method can distinguish high-dimensional data very well and has a good visualization effect. The clustering result is shown in Figure 7. Here, we only show several main keywords and 2-grams in Figure 7.

From the results of keyword clustering in Figure 7(a) and 2-gram clustering in Figure 7(b), we find that the effect of keyword clustering is better than that of 2-gram clustering, which is consistent with our conclusion in Section 4. In Figure 7, we find that the *news* is more concentrated and the core is centered on "经济" (economy). The *novel* can be divided into two groups, because our novel corpus consists of works written by two authors, Mo Yan and Yu Hua, as indicated by the red circle in Figure 7(a); each red circle represents one class. The *text book* is more scattered. This is

because the theme of the *text book* corpus is diversified, so the clustering of the *text book* contains several patches. Next, we analyze it from the left to the right and from the bottom to the top.

From the left side of Figure 7(a), the names of these protagonists are mainly found in Mo Yan's novels. The reason is that Mo Yan's novels focus on the anti-Japanese war and the Cultural Revolution. The right side of the *novel* shows the protagonists of Yu Hua's works. Yu Hua's works mainly focus on the era after the reform and opening up of China; therefore, Yu Hua's work is closer to the *news*. For the *text book*, we found that the clustering algorithm groups "鲁迅" (LuXun)'s work together and "孔子" (Confucius) and "庄子" (Zhuangzi) are also grouped together, as well as works related to "罗密欧" (Romeo) and "朱丽叶" (Juliet). On the one hand, we can see that "余华" (Yu Hua) and "鲁迅" (LuXun)'s collections are relatively close, which shows that Yu Hua and "鲁迅" (LuXun)'s writing registers are relatively similar. We found that after graduating from high school in 1977, "余华" (Yu Hua) entered the Beijing "鲁迅" (LuXun) College of Literature for further study and he might, therefore, have been influenced by "鲁迅" (LuXun)'s writing register during his studies. On the other hand, the *text book* is similar to the *news*, especially "列宁" (Lenin), "梅兰芳" (Mei Lanfang) etc. We found that these keywords are related to the theme of patriotism and thus are closer to the *news*. From the clustering results above, word vectors trained by our model have a significant effect.

Unlike keyword clustering, the *novel* and *text book* are not well distinguished in 2-gram clustering and the clustering results of the *novel* and *text book* are relatively discrete. The reason is that there are dialogues in both the *novel* and *text book*, especially the structures such as "NN + VV" and "*** + SP" shown in Figure 6, which leads to a poor distinction between fiction and textbooks. There are few dialogues in the *news*, and the structures of "NN + NN" and "VV + VV" shown in Figure 6 for the *news* are very significant.

In fact, in our paper, the attention network has two functions. One is to extract n-gram keywords that can distinguish the novel, news, and text book; the other is to obtain the vectorization of each n-gram by training the attention network.

4. Conclusion and Future Work

We propose a model attentive n-gram network (ANN) for key n-gram extraction. Our model makes full use of the spatial semantic information of words, and the attention mechanism scores each n-gram in the sentence. With the increasing accuracy of training models, the attention mechanism scores each word more accurately. In the experiment, the classification accuracy of our model is significantly higher than the baseline accuracy. In particular, our model is not limited to 1,2-grams, but n of n-grams is also applicable to 3, 4, 5, and 6 as well. In the future, we will conduct further explorations in the following two directions:

(i) We will further explore the factors that influence the attention mechanism, such as the length of sentences and the occurrence number of keywords, to improve the analyses of the characteristics of each register

(ii) We will also extend phrase structures to sentences and paragraphs to explore registers. In this way, we can study the register more comprehensively from keywords, phrases, phrase structures, sentences, and paragraphs

Data Availability

Part of this article USES the data set is available, such as News (https://www.sogou.com/labs/resource/list_yuliao.php), and the novel and textbook are protected.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the Tsinghua University Humanities and Social Sciences Revitalization Project (2019THZWJC38), the Project of Baidu Netcom Technology Co. Ltd. Open Source Course and Case Construction Based on the Deep Learning Framework PaddlePaddle (20202000291), the Distributed Secure Estimation of Multi-sensor Systems Subject to Stealthy Attacks (62073284), the Multi-sensor-based Estimation Theory and Algorithms with Clustering Hierarchical Structures (61603331), and the National Key Research and Development Program of China (2019YFB1406300).

References

[1] A. G. Turkina, O. Y. Vinogradova, N. D. Khoroshko, and A. I. Vorobyov, "Russian register of patients with chronic myeloid leukemia," *Gematologiya i transfuziologiya*, vol. 52, no. 2, pp. 7–11, 2007.

[2] P. Náther, *N-gram Based Text Categorization*, Lomonosov Moscow State University, 2005.

[3] Y. Wang and J. Zhang, "Keyword extraction from online product reviews based on bi-directional lstm recurrent neural network," in *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pp. 2241–2245, Singapore, 2017.

[4] B. T. Hung, "Vietnamese keyword extraction using hybrid deep learning methods," in *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, pp. 412–417, Ho Chi Minh City, Vietnam, 2018.

[5] Y. Wen, H. Yuan, and P. Zhang, "Research on keyword extraction based on word2vec weighted textrank," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 2109–2113, Chengdu, China, 2016.

[6] G. Ercan and I. Cicekli, "Using lexical chains for keyword extraction," *Information Processing & Management*, vol. 43, no. 6, pp. 1705–1714, 2007.

[7] L. Peng, B. Wang, S. Zhiwei, C. Yachao, and L. Hengxun, "Tag-textrank: a webpage keyword extraction method based on tags," *Journal of Computer Research and Development*, vol. 49, no. 11, pp. 2344–2351, 2012.

[8] A. Hulth, "Improved automatic keyword extraction given more linguistic knowledge," in *Conference on Empirical Methods in Natural Language Processing*, 2003.

[9] S. S. Kathait, S. Tiwari, A. Varshney et al., "Unsupervised key-phrase extraction using noun phrases," *International Journal of Computer Applications*, vol. 162, no. 1, pp. 1–5, 2017.

[10] S. K. Biswas, M. Bordoloi, and J. Shreya, "A graph based keyword extraction model using collective node weight," *Expert Systems with Applications*, vol. 97, pp. 51–59, 2018.

[11] J. Zhai, S. Gao, Z. Yu et al., "Keywords extraction in Chinese–Vietnamese bilingual news based on hypergraph," *International Journal of Distributed Sensor Networks*, vol. 14, no. 11, 2018.

[12] B. Chang and S. Yu, "Corpus technology and application," *Language Study*, vol. 5, pp. 43–51, 2009.

[13] W. Zhan, "Chinese linguistics research in the age of big data," *Journal of Shanxi University*, vol. 5, pp. 70–77, 2013.

[14] Y. Uzun, *Keyword Extraction using Naive Bayes*, Bilkent University, Department of Computer Science, Ankara, Turkey, 2005, http://www.cs.bilkent.edu.tr/~guvenir/courses/CS550/Workshop/Yasin_Uzun.pdf.

[15] K. Zhang, H. Xu, J. Tang, and J.-Z. Li, "Keyword extraction using support vector machine," in *WAIM*, vol. 4016 of Lecture Notes in Computer Science, pp. 85–96, Springer, 2006.

[16] J. Feng, F. Xie, X. Hu, P. Li, J. Cao, and W. Xindong, "Keyword extraction based on sequential pattern mining," in *Proceedings of the third international conference on internet multimedia computing and service*, pp. 34–38, 2011.

[17] C. Zhang, "Automatic keyword extraction from documents using conditional random fields," *The Journal of Computer Information Systems*, vol. 4, no. 3, pp. 1169–1180, 2008.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, http://arxiv.org/abs/1409.0473.

[19] N. Pappas and A. Popescu-Belis, "Multilingual hierarchical attention networks for document classification," 2017, http://arxiv.org/abs/1707.00896.

[20] Y. Ding, L. Yang, H. Luan, and M. Sun, "Visualizing and understanding neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1150–1159, 2017.

[21] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[22] B. L. Kalman and S. C. Kwasny, "Why tanh: choosing a sigmoidal function," in *Proceedings 1992 IJCNN International Joint Conference on Neural Networks*, pp. 578–581, 1992.

[23] M. Leshno, V. Y. Lin, A. Pinkus, and S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function," *Neural Networks*, vol. 6, no. 6, pp. 861–867, 1993.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT press, 2016.

[25] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[26] P.-T. De Boer, "A tutorial on the cross-entropy method," *Annals of Operations Research*, vol. 134, no. 1, pp. 19–67, 2005.

[27] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, http://arxiv.org/abs/1412.6980.

[28] C. W. Hess, H. T. Haug, and R. G. Landry, "The reliability of type-token ratios for the oral language of school age children," *Journal of Speech, Language, and Hearing Research*, vol. 32, no. 3, pp. 536–540, 1989.

[29] D. Li and K. Wang, *A Corpus-Based Study of Lexical Patterns in Chinese-English Simultaneous Transmission*, Modern Foreign Languages, 2012.

[30] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, "Building a large annotated corpus of english: the penn treebank," *Computational Linguistics*, vol. 19, no. 2, pp. 313–330, 1993.

[31] R. W. Langacker, "Linguistic manifestations of the space-time (dis) analogy," in *Space and Time in Languages and Cultures: Language, Culture, and Cognition*, pp. 191–216, 2012.

[32] G. C. Linderman and S. Steinerberger, "Clustering with t-sne, provably," 2017, http://arxiv.org/abs/1706.02582.