

Article

The Status and Trend of Chinese News Forecast Based on Graph Convolutional Network Pooling Algorithm

Xiao Han ^{1,2}, Jing Peng ³, Tailai Peng ^{1,2}, Rui Chen ^{1,2}, Boyuan Hou ^{1,2}, Xinran Xie ^{1,2} and Zhe Cui ^{1,2,*}

¹ Chengdu Institute of Computer Applications, Chinese Academy of Sciences, Chengdu 610041, China; hanxiao16@mails.ucas.ac.cn (X.H.); boykaptl@gmail.com (T.P.); ray2020@foxmail.com (R.C.); houboyuan19@mails.ucas.ac.cn (B.H.); xiexinran19@mails.ucas.ac.cn (X.X.)

² University of Chinese Academy of Sciences, Beijing 100049, China

³ School of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China; pengj@cuit.edu.cn

* Correspondence: cuizhe@casit.com.cn; Tel.: +86-138-0808-8762

Abstract: It is always a hot issue in the intelligence analysis field to predict the trend of news description by pre-trained language models and graph neural networks. However, there are several problems in the existing research: (1) there are few Chinese data sets on this subject in academia and industry; and (2) using the existing pre-trained language models and graph classification algorithms cannot achieve satisfactory results. The method described in this paper can better solve these problems. (1) We built a Chinese news database predicted by more than 9000 annotated news time trends, filling the gaps in this database. (2) We designed an improved method based on the pre-trained language model and graph neural networks pooling algorithm. In the graph pooling algorithm, the Graph U-Nets Pooling method and self-attention are combined, which can better solve the analysis of the problem of forecasting the development trend of news events. The experimental results show that the effect of this method compared with the baseline graph classification algorithm is improved, and it also solves the shortcomings of the pre-trained language model that cannot handle very long texts. Therefore, it can be concluded that our research has strong processing capabilities for analyzing and predicting the development trend of Chinese news events.

Keywords: natural language processing; graph convolution neural network (GCN); graph pooling algorithm



Citation: Han, X.; Peng, J.; Peng, T.; Chen, R.; Hou, B.; Xie, X.; Cui, Z. The Status and Trend of Chinese News Forecast Based on Graph Convolutional Network Pooling Algorithm. *Appl. Sci.* **2022**, *12*, 900. <https://doi.org/10.3390/app12020900>

Academic Editors: Andrea Prati, Carlos A. Iglesias, Vincent A. Cicirello and Luis Javier García Villalba

Received: 15 November 2021

Accepted: 7 January 2022

Published: 17 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, knowledge graphs, graph neural networks, and pre-trained language models have become more and more important under the development of deep learning. The traditional concept of knowledge graph was proposed by Google. It mainly uses named entity recognition, knowledge extraction, and visualization technologies to turn complex knowledge into a relatively friendly new data structure, thereby using knowledge-extracted triples to perform graph construction and graph feature extraction and subsequent applications. The graph neural networks can use the structure of the connected graph in the knowledge graph to learn and extract features, to achieve the goals of various downstream tasks. In recent years, researchers have applied knowledge graphs to the intelligent question-answering systems, recommendation systems, intelligent search engines, and other fields. In particular, the application of convolutional neural networks, recurrent neural networks, graph neural networks, and deep reinforcement learning in the knowledge graph makes it easier for researchers to extract the feature information in the knowledge graph. In the meantime, as the theory of graph convolutional neural networks is developing, researchers have discovered that graph convolutional neural networks can be applied in many fields. For example, graph-level task prediction technology for protein molecules in nature [1], node-level task traffic congestion prediction technology under the

same road section and different time background [2], node-level task three-dimensional structure component assembly technology [3], and so on. In the field of natural language processing, graph convolutional neural networks also have many creative applications. For example, graph convolutional networks can be used to analyze user speeches in social networks and build user portraits. On shopping sites, graph convolutional networks can be used to analyze user preferences and predict the products that users want to buy while recommending [4]. For traditional text classification tasks, Text GCN using knowledge graphs and graph convolutional neural networks has made great progress in this regard [5]. The author constructs a graph on the entire corpus, takes words and entities as nodes in the graph, uses co-occurrence relationship information to construct edges in the graph, and then regards the text classification problem as a node classification problem. At the same time, in the past two years, researchers have carried out a variety of studies following the path of Text GCN.

With the development of pre-trained language models in the past few years, researchers have increasingly used pre-trained language models to implement text-related tasks. Compared with the previous language models, the pre-trained language model effectively learns global semantic information and significantly improves the accuracy of natural language processing tasks. Among them, representative pre-trained language models such as ELMo [6], GPT [7–9], and BERT [10] have further improved the effect of pre-trained language models. ELMo uses a bidirectional LSTM model, which consists of one forward and one backward language model. The objective function takes the maximum likelihood of these two directions and summates each middle layer of the bidirectional language model. Later, when performing supervised NLP tasks, ELMo can be directly used as features to be spliced to the word vector input of the specific task model or the highest-level representation of the model. After that, the OpenAI released the GPT (Generative Pre-Training) language model in order to build an unsupervised general language model in a variety of natural language processing tasks. The biggest highlight of GPT is that it is different from ELMo's bidirectional LSTM model, but uses a Transformer encoder [11] that can capture long-distance information. When we use GPT, the model can be supervised and fine-tuned according to specific downstream tasks. Coincidentally, the BERT model proposed by Google combines the advantages of ELMo and GPT, uses the Transformer encoder while adding a bidirectional structure, and proposes two new natural language processing target tasks, MLM (Masked Language Model) and the task of predicting the next sentence. These pre-trained language models refreshed the list of natural language processing tasks as soon as they were proposed. After fine-tuning the pre-trained language models, researchers can adapt to their respective downstream tasks. Of course, the pre-trained language model also has some shortcomings. During pre-training, due to the need to grasp the global semantic information, the training overhead is usually the square of the training text length. Therefore, the aforementioned pre-trained language models usually limit the text length to 512.

Many researchers use news text to construct news knowledge graphs and use graphs to judge news development trends. At the same time, other researchers use pre-trained language models to fine-tune to achieve news text classification and development trend prediction. The method of using news knowledge graphs can handle long texts, but the effect is relatively poor; the method of using pre-trained language models has good results, but it is incapable of processing long texts. What's more, few researchers study the development trend of Chinese news texts. The lack of a fully-labelled database and technical obstacles make it an unsolved problem for researchers to use Chinese news texts to predict news development trends.

To better solve the above problems, we have made two efforts. We first used web crawler technology to build a database of Chinese news text trends. This database contains nearly 9000 labelled news data, including news time, news headlines, news sources, news text, and news trend labels, which fill the gaps in the current analysis and trend prediction database of Chinese news texts. Secondly, we designed a news text trend prediction method

that uses knowledge extraction technology, graph convolutional neural networks, and pre-trained language models. We use the open-source code on GitHub named TextGrapher [12] to generate a homogenous knowledge graph of news text. The open-source code uses the TF-IDF algorithm to extract keywords and high-frequency words of the news text and uses named entity recognition and knowledge extraction to extract ternary relationship groups and build news knowledge graphs. At the same time, we use the Chinese pre-trained language model of HFL (Joint Laboratory of Harbin Institute of Technology and iFLYTEK Research) [13] to obtain the embedding vector of the named entity as the initial value of the node of the graph convolutional neural network and use the graph convolutional neural network to extract the entire graph feature and design the graph-level task. At the same time, the graph pooling algorithm is improved to make the graph embedding vector better retain the hierarchical features, and the convolved graph embedding vector can be used for comparison and prediction of news trends. The experimental results show that the effect of this method compared with the baseline graph classification algorithm is improved, and it also solves the shortcomings of the pre-trained language model, which cannot handle very long texts.

2. Methods

The overall architecture of the method we designed is shown in Figure 1. There are three steps from left to right. The first step is the input of news text. In Figure 1, five news texts are taken as an example. They are marked as T1, T2, T3, T4, and T5. The open-source code TextGrapher transforms five news texts into a homogenous knowledge graph. In the second step, the tensor composed of these graphs is first obtained through BERT as Service to obtain the word embedding vector of each node as the initial value of the node of the graph convolutional neural network. In the third step, we use the graph convolutional neural network and its pooling method to obtain the embedding vector of each knowledge graph and form these vectors into a graph embedding vector matrix. The cross-entropy loss function is used to train the graph convolutional neural network and update the parameters to realize the classification of news texts with different labels.

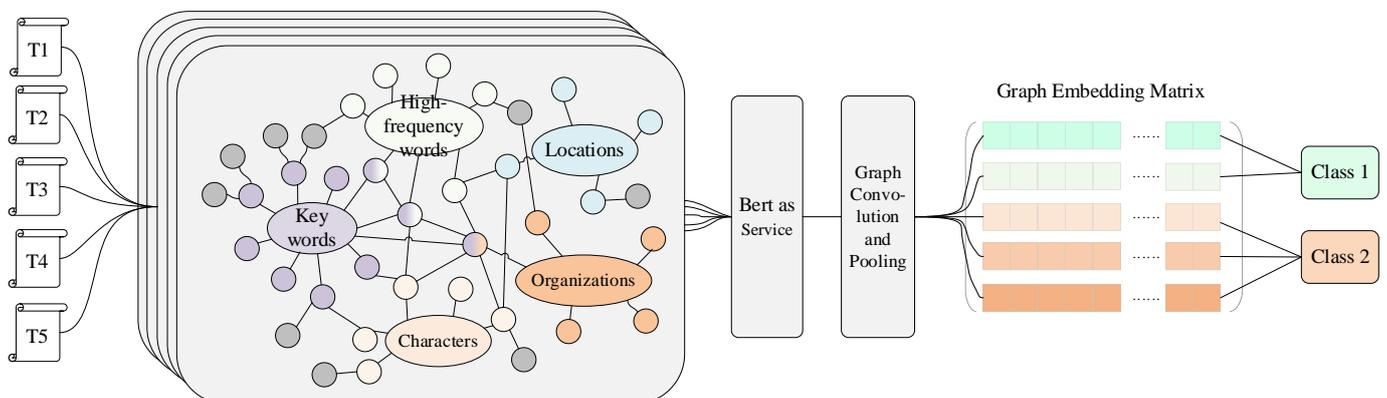


Figure 1. Schematic of our method. There are five news texts for example.

2.1. Graph Convolutional Neural Network

Graph convolutional neural network [14] is developed from graph neural network [15]. According to the definition of graph neural network, we formally define the set of graphs $G = (g_1, g_2, \dots, g_n)$. For each graph $g = (V, E)$, where V is the set of points in the graph. E is the set of edges in the graph. For any point $v \in V$, there is an edge connected to itself, that is $(v, v) \in E$. For each graph, there is a matrix $X \in \mathbb{R}^{i \times j}$ that represents the initial feature set of each point in the graph, where i represents the number of nodes in the graph and j represents the dimension of each node's feature. We will discuss how to obtain the initial state in Section 2.2. For each graph, there is a corresponding adjacency matrix A and degree matrix D . Then, the matrix $\tilde{A} = D^{-\frac{1}{2}}AD^{\frac{1}{2}}$ is the normalized symmetric adjacency

matrix of the graph. According to the definition of GCN, the convolution formula for the first layer of the graph is:

$$H^{(1)} = \text{ReLU}(\tilde{A}XW_0) \quad (1)$$

In this formula, $W_0 \in \mathbb{R}^{j \times k}$ represents the weight matrix. $\text{ReLU}(\cdot)$ represents the activation function, defined as $\text{ReLU}(x) = \max(0, x)$. Iterating the number of GCN layers can get deeper features. By using the GCN hidden state update formula, our hidden state update formula can be obtained:

$$H^{(t+1)} = \text{ReLU}(\tilde{A}H^{(t)}W_0) \quad (2)$$

where t represents the number of iterations. In particular, $H^{(0)} = X$.

2.2. Node Initial State and Knowledge Graph Structure

In the application of GCN, there is no fixed method to obtain the initial state of nodes. Generally, researchers use existing pre-trained models to obtain vector representation of words or phrases, such as CBOW (Continuous bag-of-words Model), Or Globe (Global Vectors of Word Representation). Each of them has advantages and disadvantages. After the BERT pre-trained model was proposed, researchers had the idea of using the pre-trained language model to obtain the word embedding vector. The BERT as Service [16] project that turns BERT into a service is a method constructed based on this idea. In addition, we can also use any pre-trained model of BERT as our server-side pre-trained processing model. For example, the pre-trained language model of English relies more on the method of masking words, while the basic unit of Chinese expression of semantics is more dependent on phrases. Therefore, BERT-wwm is used to mask the whole word. In order to better recognize the semantics of Chinese text, we use the Chinese BERT-wwm-ext pre-trained model released by HFL (Joint Laboratory of Harbin Institute of Technology and iFLYTEK Research). Compared with other Chinese BERT-wwm models, the BERT-wwm-ext model uses a larger corpus and increases the number of training steps to further improve the effect of the pre-trained model.

We also make some improvements to the news knowledge graph networks formed by extracting triples. For news texts with different lengths, after a large number of knowledge extraction of news texts, we found that the connection relation of these triples was not as close as expected. For a specific news text, because there are too few connections, it is very likely that 2 to 4 independent subgraphs will appear. There will be no logical connection between the subgraphs. In order to solve this problem, we create five virtual nodes in the knowledge map. The semantics they represent are keywords, high-frequency words, news characters, news locations, and news organizations. Using the TF-IDF algorithm and named entity recognition, we can get the attributes of each entity in the knowledge graph, and connect them with five virtual nodes as much as possible. Other entities outside the scope of these five virtual nodes can access the knowledge graph by the traditional entity relation extraction triplet method. For example, in the knowledge graph in 1, entities connected to these five virtual nodes are marked with the same color as corresponding virtual nodes, while entities not belonging to these five virtual nodes are marked with gray. Obviously, for these 5 virtual nodes, we set their initial vectors to 768 dimensions that are the same as the output dimension of Bert As Service, and assign each dimension's value to 0.

2.3. Improved Method of Graph Pooling Based on Graph U-Net and Self-Attention Mechanism

For the machine to quickly and accurately determine the class of the graph, we need a method to get the embedding vector of the graph. In the traditional method, we can reduce the size of the feature map and increase the range of the receptive field by pooling. Commonly used graph pooling algorithms such as SortPool [17], DiffPool [18], Graph U-Nets [19], and SagPool [20]. Graph U-Nets refers to the structure of U-Net

and designs an up-sampling and down-sampling method. It uses graph convolution and down-sampling to extract high-dimensional features, and then uses up-sampling and skip-connections to restore low-dimensional features. SagPool uses the self-attention mechanism to calculate and delete relatively unimportant nodes in the graph, aggregate high-dimensional features, and propose the concepts of global pooling architecture and hierarchical pooling architecture.

However, after the experiment, we find that the accuracy of the existing methods has the problem of low accuracy when solving practical problems. We propose a new graph pooling method based on self-attention mechanism and the Graph U-Nets method. It combines the structure of graph u-net and applies a self-attention mechanism in the output layer, which can better get the graph embedding vector we are interested in, so as to improve the effect of graph classification. As shown in Figure 2, following the structure of Graph U-Nets, we adopt a pooled block structure composed of 2-layer graph down-sampling and graph convolution. We define gPool as:

$$y = \frac{X^\ell p^\ell}{\|p^\ell\|} \tag{3}$$

$$index = rank(y, k) \tag{4}$$

$$\tilde{y} = sigmoid(y(index)) \tag{5}$$

$$\tilde{X}^\ell = X^\ell(index, :) \tag{6}$$

$$A^{\ell+1} = A^\ell(index, index) \tag{7}$$

$$X^{\ell+1} = \tilde{X}^\ell \odot (\tilde{y} \mathbf{1}_C^T) \tag{8}$$

where p is a trainable projection vector. Given a node i and its feature vector x_i , the projection of x_i in the p direction is $y_i = \frac{x_i p}{\|p\|}$. When a graph g is given, X represents the feature vectors of all nodes in the graph and ℓ represents the number of layers of graph convolution during the gPool operation. Similar to the TOP-K algorithm, $rank(y, k)$ is a method of selecting nodes, k represents the number of k maximum values in y , so $index$ is a new graph composed of y including k indexes. $X^\ell(index, :)$ and $A^\ell(index, index)$, respectively, represent the feature matrix and adjacency matrix in the new graph pooling of the ℓ -th layer. The number of their rows and columns is shown in parentheses. \tilde{y} means that the sigmoid function is used for nonlinear activation function after taking the top k values. $A^{\ell+1}$ and $X^{\ell+1}$, respectively, represent the adjacency matrix and feature matrix passed to the next layer. $\mathbf{1}_C^T$ represents a feature vector whose all elements are 1 and the size equals to C . Through gPool operation, the subgraphs composed of more important nodes in the upper layer can be selected and the new subgraph can be used for graph convolution, so as to achieve the purpose of aggregation and dimensionality reduction.

We define the gUnpool operation as:

$$X^{\ell+1} = distribute(0_{N \times C}, X^\ell, index) \tag{9}$$

where $index$ is the node directory containing the corresponding gPool operation from N nodes to k nodes. X^ℓ is the matrix composed of the eigenvectors of the current graph. $0_{N \times C}$ is an empty set of vectors that corresponds to nodes not selected by gPool and adds the 0 vector to the corresponding position. In other words, the $distribute(\cdot)$ function actually assigns the nodes that have not been selected by the corresponding gPool operation to the 0 vector and restores them to the new graph.

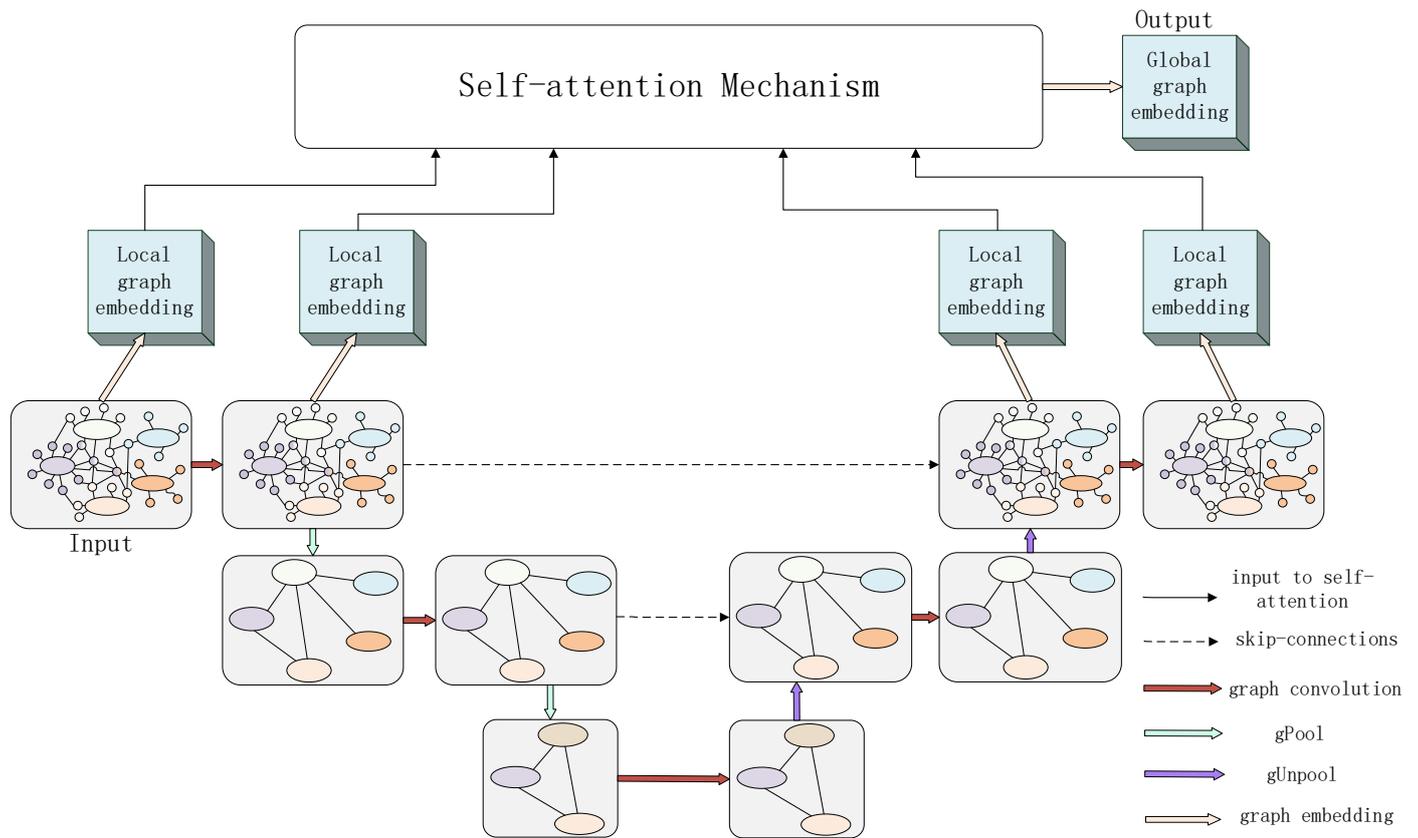


Figure 2. Schematic diagram of improved graph pooling method based on graph u-net and self-attention mechanism.

In addition to applying the design of Graph U-Nets, we also take the two graphs before gPool operation of the first layer and the two graphs after gUnpool operation of the last layer in U-Net. We transform these four graphs into a graph embedding vector and define the four inputs of the self-attention mechanism as X_1, X_2, X_3, X_4 , respectively, which is defined as the input sequence $X = (X_1, X_2, X_3, X_4)$. We input them into the self-attention module, as follows:

$$\begin{aligned} Q &= W_Q X \in \mathbb{R}^{d_3 \times N} \\ K &= W_K X \in \mathbb{R}^{d_3 \times N} \\ V &= W_V X \in \mathbb{R}^{d_2 \times N} \end{aligned} \quad (10)$$

By referring to the self-attention mechanism, we can use the three matrices of $Q, K,$ and V to calculate the hidden state:

$$H = \text{attention}((K, V), Q) = \sum_{j=1}^N \text{softmax}(s(\mathbf{k}_j, Q)) \mathbf{v}_j \quad (11)$$

During training, we only need to use the cross-entropy loss to compare the existing label with the hidden state of the output, and constantly update the parameters of the graph convolutional neural network to make the loss function converge. In actual prediction, we find a set of hidden states of the trained data, and use the same parameters to predict the hidden state of the new input. Then, we find the closest pair of hidden states from the training data and use the label of known data as the label prediction value of new data.

3. Results

3.1. The Design of Dataset

The purpose of writing this article is to solve how to qualitatively analyze and predict the current situation and development trend of news in practice. This project is supported by the Chinese Sichuan Science and Technology Program Department of Science and Technology. There are three states for a piece of news, positive, negative, or no impact, so the labels of these three states are marked as 1, 2, and 0, respectively. The purpose is to detect the current situation of the news and make judgments and early warnings of major security incidents. We use web crawler technology to crawl Chinese news, and 9925 pieces of data were crawled on Chinanews.com. After that, 9381 pieces of data were left after removing duplicates. In order to accurately label data, we have formulated the following labeling rules:

- News impact is labelled 1 for positive, 2 for negative, and 0 for no impact;
- If the impact of the news cannot be judged, the judgment should be based on the events behind the news;
- If the impact cannot be determined through the event, it is uniformly labelled as 0.

There are four examples as follow:

1. Example 1: The headline of a piece of news is: "Kim Yo-jung condemns the Korean Foreign Minister's remarks on the prevention of the epidemic: Pouring cold water on the relationship between the DPRK and South Korea." We label this news as 2;
2. Example 2: The headline of a piece of news is: "Report shows: China's agricultural economy continues to improve and farmers' income grows steadily". We label this news as 1;
3. Example 3: The headline of a piece of news is: "The Central Bank of China has invested 80 billion yuan in a week. Institutions said: The probability of funding relaxation at the end of the month is greater". We label this news as 0;
4. Example 4: The headline of a piece of news is: "Xin Xuan responds again to alleged fraud in selling goods: it is a fact, we will refund one for three." We label this news as 0.

In example 4, the first part of the news describes how Xin Xuan falsified, and the second part describes how Xin Xuan compensates to consumers, so it is hard to intuitively judge the impact of this news. Therefore, we label it as 2 on the grounds that the event behind it is not positive.

In the dataset, a total of three taggers participated in labeling data, and they completed their own tasks independently. Finally, we decided to adopt a voting mechanism to vote the data marked by different taggers. The principle is that if the judgments of the three taggers are consistent, we think their label is true. If two news items are consistent, the other is different, or the labels of the three taggers are different, we will determine the final trend judgment of the news through joint discussion among the authors. In the dataset, there are 3573 pieces of data labelled as 0; 1902 pieces of data labelled as 1, and 3906 pieces of data labelled as 2. We also counted 2524 news content with a text length of more than 512 and 6857 news content with a text length of more than 512.

3.2. Baseline and Result

The evaluation work is to randomly select 90% of the dataset as the training set, and the remaining 10% as the test set. Test data will not be used during model training. In order to better verify the effect of the method we designed, we selected the current mainstream image classification method and some pre-trained language models as a comparison. Graph classification methods include SortPool, DiffPool, gPool, SAGPool, and our improved source Graph U-Nets. The direct text classification results of the pre-trained language model are also included in the indicators, including BERT-base (Chinese), BERT-large (Chinese), and BERT-wwm-ext (Chinese), three Chinese pre-trained language models. The evaluation index is the accuracy of classification. During the test, all the data marked by the three testers will be tested independently, and whether the ability to process long texts

can be included in the assessment indicators. In terms of hyperparameter settings for the baseline, we choose to refer to the SAGPool. The experimental results are shown in Table 1. The method we designed is better than the graph classification methods, but not better than the pre-trained language models.

Table 1. Results of learning experiments in terms of node classification accuracies on our dataset. Attention graph U-Nets denotes our proposed model.

Accuracy of Classification	Precision	Recall	F1-Measure
SortPool	72.22%	75.09%	73.63%
DiffPool	77.21%	75.56%	76.37%
gPool	74.65%	70.31%	72.42%
SAGPool	80.51%	78.24%	79.36%
Graph U-Nets	78.16%	81.11%	79.61%
Attention Graph U-Nets (gensim)	78.60%	80.55%	79.56%
Attention Graph U-Nets (Bert)	81.43%	80.24%	80.83%
BERT-base (Chinese)	91.53%	89.00%	90.25%
BERT-large (Chinese)	91.44%	91.59%	91.51%
BERT-www-ext (Chinese)	92.07%	93.80%	92.93%

However, we all know that it is unfair to let the graph classification method directly compare the classification results with the pre-trained language model because it involves the loss of information from the text to the graph. Moreover, the pre-trained language model is obtained after a lot of computing power, time and data training, and as the length of the text increases, the complexity of training the pre-trained language model is a square level increase, so most pre-trained languages models set the maximum length of the processed text to 512. In the experiment, directly using the pre-trained language model for news text classification is to discard the text that cannot be processed, and directly process the news text with the text length less than 512. Therefore, it can be concluded that the model we have designed has strong capabilities in both accuracy and processing long-text news.

According to the requirements of our scientific research projects, in addition to classifying news trend forecasts, we also need to display them visually, so we use knowledge graphs to visualize them. Regarding the question of why not directly use the pre-trained language model, but also use the pre-trained language model to obtain the initial embedding vector, we just used the pre-trained language model to obtain the initial vector. In fact, using any other method to obtain the word embedding vector, such as the gensim Chinese word embedding vector method, we can also obtain the initial embedding vector of the node, but it does this. The effect is a little worse than using BERT. We can also see that from Table 1 Attention Graph U-Nets (gensim) and Attention Graph U-Nets (BERT).

4. Conclusions

Compared with the use of pre-trained language models, our method is indeed not good enough, but our method can solve the shortcomings of the limited text input length of pre-trained language models. We believe that the pre-trained language model is limited by the length of the text, and there is no good solution for text classification with a length greater than 512. Normally, the text is divided into several texts with a length of 512 before processing. Even if the current new pre-trained language models such as XL-Net [21] break through the limitation of text length, it is achieved by increasing the computing performance and increasing the space occupied by the video memory. The longer the text length, the greater the difficulty of pre-trained and the difficulty increases in proportion to the square. When processing this part of the data in the experiment, we only use the first

512 words as input to the BERT model, and the other parts can only be discarded, because in Chinese news, the first half of the news likely describes an event, and the follow-up describes the solution or plan of the event. The trend of the description of the event and the solution is the opposite, so the input is divided into sections. The method of the BERT model is not suitable. However, we propose to use graphs to solve this problem to convert texts whose length exceeds the limit into knowledge graphs, extract high-dimensional features and use them to achieve classification, without being restricted by length. In other words, the longer the text length, the more useful triples can be extracted from the knowledge graph we construct, and the relationships between these triples can be used to classify them.

Of course, this also leads to the second problem. Knowledge graphs do lose part of the information during the conversion process. Compared with text, knowledge graphs can only extract the main structure of sentences and construct triples. For some attributive adverbials and other information, the loss is still serious. Our solution to this is as mentioned in Section 2.2. In the knowledge graph, we make up five virtual nodes, which are keywords, high-frequency words, news characters, news locations, and news organizations, and set the initial vector of these nodes to a zero vector to increase the connectivity and relevance in the knowledge graph. We found that relatively short news text (for example, the text length is less than 100) does not work well for classification. The reason is that the text length is too short, which is not conducive to us extracting enough entities to build triples and knowledge graphs.

In summary, this paper has made two contributions. (1) We construct a Chinese news data set predicted by more than 9000 annotated news time trends. (2) We design an improved method based on a pre-trained language model and graph pooling algorithm, which can better balance accuracy and processing long-text news.

Author Contributions: Conceptualization, X.H. and Z.C.; software, X.H.; validation, J.P.; formal analysis, X.X.; resources, R.C.; data curation, T.P.; writing—original draft preparation, X.H.; writing—review and editing, X.H.; visualization, B.H.; supervision, Z.C.; project administration, Z.C.; funding acquisition, Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Sichuan Science and Technology Program, grant number “2020YFG0009” and Sichuan Science and Technology Program, grant number “2019ZDZX0005”.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: You can get our dataset at <https://share.weiyun.com/YLLKim53> (accessed on 9 January 2022).

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

References

1. Senior, A.; Jumper, J.; Hassabis, D.; Kohli, P. AlphaFold: Using AI for Scientific Discovery. DeepMind. 2018. Available online: <https://deepmind.com/blog/alphafold> (accessed on 9 January 2022).
2. Feng, N.; Guo, S.N.; Song, C.; Zhu, Q.C.; Wan, H.Y. Multi-component Spatial-temporal Graph Convolution Networks for Traffic Flow Forecasting. *J. Softw.* **2019**, *30*, 759–769. (In Chinese). Available online: <http://www.jos.org.cn/1000-9825/5697.html> (accessed on 9 January 2022).
3. Huang, J.; Zhan, G.; Fan, Q.; Mo, K.; Shao, L.; Chen, B.; Guibas, L.; Dong, H. Generative 3d part assembly via dynamic graph learning. *arXiv* **2020**, arXiv:2006.07793.
4. Wang, H.; Zhang, F.; Zhang, M.; Leskovec, J.; Zhao, M.; Li, W.; Wang, Z. Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2019): n. pag, Anchorage, AK, USA, 4–8 August 2019.
5. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7370–7377.
6. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.

7. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018. Available online: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (accessed on 14 November 2021).
8. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
9. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *arXiv* **2020**, arXiv:2005.14165.
10. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
11. Aswani, A.V.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; 2017; pp. 5998–6008. Available online: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf> (accessed on 14 November 2021).
12. GitHub—Liuhuanyong/TextGrapher: Text Content Grapher Based on Keyinfo Extraction by NLP Method. Available online: <https://github.com/liuhuanyong/TextGrapher> (accessed on 8 November 2021).
13. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Yang, Z.; Wang, S.; Hu, G. Pre-training with whole word masking for Chinese bert. *arXiv* **2019**, arXiv:1906.08101. [[CrossRef](#)]
14. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2017**, arXiv:1609.02907.
15. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The Graph Neural Network Model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61. [[CrossRef](#)] [[PubMed](#)]
16. GitHub—Hanxiao/Bert-as-Service: Mapping a Variable-Length Sentence to a Fixed-Length Vector Using BERT Model. Available online: <https://github.com/hanxiao/bert-as-service> (accessed on 8 November 2021).
17. Zhang, M.; Cui, Z.; Neumann, M.; Chen, Y. An end-to-end deep learning architecture for graph classification. In *Thirty-Second AAAI Conference on Artificial Intelligence*; 2018. Available online: <https://ojs.aaai.org/index.php/AAAI/article/view/11782> (accessed on 14 November 2021).
18. Ying, R.; You, J.; Morris, C.; Ren, X.; Hamilton, W.L.; Leskovec, J. Hierarchical graph representation learning with differentiable pooling. *arXiv* **2018**, arXiv:1806.08804.
19. Gao, H.; Ji, S. Graph u-nets. In *International Conference on Machine Learning*; PMLR; 2019; pp. 2083–2092. Available online: <http://proceedings.mlr.press/v97/gao19a/gao19a.pdf> (accessed on 14 November 2021).
20. Lee, J.; Lee, I.; Kang, J. Self-attention graph pooling. In *International Conference on Machine Learning*; PMLR; 2019; pp. 3734–3743. Available online: <http://proceedings.mlr.press/v97/lee19c.html> (accessed on 14 November 2021).
21. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*; 2019; Volume 32. Available online: <https://proceedings.neurips.cc/paper/2019/hash/dc6a7e655d7e5840e66733e9ee67cc69-Abstract.html> (accessed on 14 November 2021).