A Multi-Model-based Approach to Corpus Text Pre-Screening

Zuhao Wu
College of Foreign Languages and Literature
Fudan University
Shanghai, China
22210120030@m.fudan.edu.cn

Yude Bi*
College of Foreign Languages and Literature
Fudan University
Shanghai, China
biyude@fudan.edu.cn

***Abstract*—The construction of a self-built corpus for a research topic necessitates the pre-screening of acquired texts. During the construction process, certain acquired texts may not be directly relevant to the research subject due to various influencing factors. While both manual and algorithmic methods can be employed for pre-screening, manual screening becomes impractical when dealing with a large volume of text. As such, we propose a multi-model-based approach utilizing TextRank, TF-IDF, and KNN algorithms for pre-screening corpus texts. And the effectiveness of this method will be validated through rigorous evaluation.**

Keywords—pre-screen, corpus, TextRank algorithm, TF-IDF algorithm, KNN algorithm

I. INTRODUCTION

When constructing a corpus around a specific research topic, certain segments of the corpus can contain text that is not directly relevant to the topic due to various influencing factors. Consequently, it becomes imperative to pre-screen the corpus to eliminate any interfering text before proceeding with further analysis. This step is crucial to ensure the integrity and reliability of subsequent research endeavors.

There have been limited studies conducted on screening methods for excluding topic-irrelevant text in self-built corpora. Most of these studies primarily focus on presenting the study results, with only a few briefly discussing the screening mechanisms employed. The majority of these studies utilize manual screening methods. For instance, Yifan Zhu and Kaibao Hu [1] investigated the semantic features and tendencies of the word "*Bei*" using a corpus compiled by the Center for Translation and Intercultural Studies at Shanghai Jiao Tong University and news articles collected by the authors. They conducted meticulous manual screening and verification to ensure the thematic relevance of the corpus. In another study, Duanyang Li and Zhijun Wang[2] examined the English register characteristics of customs news, utilizing WCO NEWS published by the World Customs Organization. Additionally, GULZIYRE·Aniwar and Zhihuan Kuang[3] analyzed the construction of a Chinese-Kazakh parallel news corpus by examining the characteristics of news web pages. These web pages exhibited a simple structure without interfering elements, ensuring the acquired corpus's quality.

In the realm of KNN algorithm research, scholars have focused on enhancing the sample features as well as refining the algorithm itself. Xiaobo Tang, Juan Zhu, and Fenghua Yang[4] achieved this by quantitatively labeling the sentiment categories of samples, enabling KNN to make informed judgments about input text. Xianying Huang, Liyuan Xiong, Yingtao Liu, and Qindong Li[5] improved the classification efficiency of the KNN algorithm by employing CHI to split and refine the existing training set. This approach yielded notable enhancements in classification performance. Zhihua Wang, Shaoting Liu, and Qi Luo[6] combined the K-modes algorithm with KNN, resulting in advantageous outcomes for classification tasks. This integration showcased improved accuracy and effectiveness. Yang Song, Hailong Wang, Lin Liu, and Dongmei Pei[7] merged kernel principal component analysis (KPCA) with an enhanced version of the KNN algorithm. This hybrid approach significantly augmented the efficiency of KNN-based classification.

As evident from the studies mentioned above, most corpus research endeavors employ a strategy that prioritizes ensuring the thematic relevance of the corpus sources. This involves manual screening prior to acquiring the corpus, thereby minimizing the likelihood of topic-irrelevant text from being included. However, this approach still faces challenges in terms of low efficiency when dealing with large volumes of diverse sources, inconsistent text formatting, and wide-ranging subject areas. Particularly for topics that are multi-sourced and comprehensive coverage, manual screening becomes impractical. In the context of KNN algorithm research, the primary focus lies in enhancing the algorithm itself and refining input features.

Hence, in this paper, we proposed a method of pre-screening self-built corpus text, particularly when the corpus possesses characteristics such as extensive coverage, diverse sources, and a wide domain. To tackle this issue, we propose a multi-model approach integrating TextRank, TF-IDF, and KNN based on machine learning algorithms. This approach aims to efficiently pre-screen the corpus by identifying text that aligns with the specified topic.

II. CONSTRUCTION OF MULTI-MODEL

In this paper, we propose the construction of a multi-model utilizing the TextRank, TF-IDF, and KNN algorithms. Specifically, TextRank considers the contextual relationships between words in the text and deduces topic-relevant terms based on link weights. TF-IDF comprehensively considers the frequency of a word within a specific text and its occurrence across all texts. The KNN algorithm assesses the distance between a given sample and other samples based on their respective features, enabling classification into appropriate categories.

The TextRank model, inspired by the PageRank algorithm, incorporates the link relationships between words to determine the weight of each word[8]. The weight is calculated based on the number of links pointing to that word. The equation for determining the weight is presented as (1).

$$WS(V_i)=(1-d)+d\sum_{V_j\in in(V_i)}\frac{W_{ji}}{\sum_{V_k\in out(V_j)}W_{jk}}WS(V_j) \quad (1)$$

*Yude Bi is the corresponding author.

In the TextRank model, the corpus is represented as a graph. Each point in the graph, denoted as Vi, represents a sentence. The set of points that point to Vi is represented as in(Vi), while the set of points that point out from Vi is denoted as out(Vi). The weight of sentence i, denoted as WS(Vi), is calculated based on the similarity between sentences. Specifically, Wji represents the similarity between two sentences, while WS(Vj) represents the result of the previous calculation. The damping factor, typically set to 0.85, is denoted as d.

The TF-IDF (Term Frequency - Inverse Document Frequency) model determines the significance of a word within a text by considering its frequency in the given text and its frequency across other texts[9]. It comprises two components: TF, which represents Term Frequency, and IDF, which represents Inverse Document Frequency. These components can be represented by (2) and (3) respectively:

$$TF_{i,j}=\frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2}$$

$$IDF_i=\log\frac{|D|}{1+|j:t_i\in d_j|} \tag{3}$$

In (2) and (3), several variables are defined as follows:

(a) nij represents the number of occurrences of the word ti in the text dj.

(b) |D| denotes the total number of texts in the corpus.

(c) |j:ti∈dj| represents the number of texts containing the word ti.

Based on these definitions, it can be observed that when a word exhibits a high frequency within a specific text and a relatively low frequency across the entire corpus, it can be considered as the topic word of that particular text. This relationship is represented by (4).

$$TF\text{-}IDF=TF\cdot IDF \tag{4}$$

The KNN algorithm, short for K-Nearest Neighbor, is a supervised machine learning classification algorithm. It belongs to instance-based learning. When given an input sample, the KNN algorithm calculates the distance between the input sample and the K nearest samples in the training set. It then determines the category of the input sample based on the majority category among those K samples. Before training, the KNN algorithm needs to determine the distance metric to use, the value of K, and the decision rule. Several distance metrics are available for KNN, including Manhattan, Euclidean, and Cosine Distance. Euclidean Distance is commonly chosen as the distance metric for KNN. For two samples A (x1, x2, ..., xk) and B (y1, y2, ..., yk), the Euclidean distance between them can be calculated using (5).

$$D(x,y)=\sqrt{\sum_{k=1}^{n}(x_k-y_k)^2} \tag{5}$$

The K value in the KNN algorithm determines the number of nearest pre-existing samples to consider when classifying an input sample. Selecting an appropriate K value is crucial as a small K value can lead to overfitting, while a large K value may result in underfitting. The choice of K value depends on the specific training scenario and should be adjusted accordingly.

The decision rule in the KNN algorithm involves methods such as majority voting and distance-weighted voting. In this paper, we adopt the majority voting approach as the decision rule. This means that the algorithm determines the category of the input sample by directly considering the majority class among the K-nearest neighboring samples.

Once the three basic elements (distance metric, K value, and decision rule) are defined, the training process for the KNN algorithm can commence. The following are the fundamental steps of the KNN algorithm:

(a) Input the sample ak.

(b) Calculate the distances between sample ak and the existing sample set D(a1, a2, ..., ak-1).

(c) Select the K points with the minimum distances calculated in step (b).

(d) Determine the frequency of each category among these K points.

(e) Assign the category with the highest frequency to the sample ak.

These steps outline the core procedure of training the KNN algorithm.

Based on the characteristics of each model, we propose constructing a multi-model consisting of TextRank, TF-IDF, and KNN. The TextRank model allows us to rank words within a single text based on link weights, but it cannot compare TextRank values across different texts. Therefore, we suggest selecting the top TextRank words (referred to as TextRank high-ranking words) from each text and creating a collection of TextRank high-ranking words.

Next, we calculate the TF-IDF value for the TextRank high-ranking words in each text relative to the overall collection of high-ranking words. This process can be seen as a normalization step. The resulting high-weight words in each text possess two attributes: TextRank ranking and TF-IDF value.

To train the KNN model, a subset of samples are manually judged and labeled. These labeled samples are then used to train the KNN model specifically for the corpus.

Finally, utilizing the trained KNN model, all the texts in the corpus can be filtered and screened based on their similarity to the labeled samples.

The multi-model structure for integrating TextRank, TF-IDF, and KNN is depicted in Fig.1. The implementation of this multi-model is carried out using Python 3.8 environment. The TextRank and TF-IDF calculations will be performed based on their respective definitions, while the KNN part will utilize the Sklearn library. Sklearn incorporates the Kdtree module, which leverages a tree structure to handle the computational challenges associated with large-scale corpora. The construction of the multi-model follows these steps:

(a) Using the TextRank algorithm, independently calculate the TextRank values for all words within each text in the corpus. Sort the words based on their TextRank values and select the top X words.

(b) Cluster the top X words from all texts into a set. Calculate the TF-IDF values for these top X words within each text.

(c) At this stage, we have obtained the TextRank rankings and relative TF-IDF values for the words in each text.

(d) Treat the TextRank rankings and relative TF-IDF values as sample features. Manually select and label several positive and negative examples for evaluation, then use these labeled samples to train the KNN algorithm.

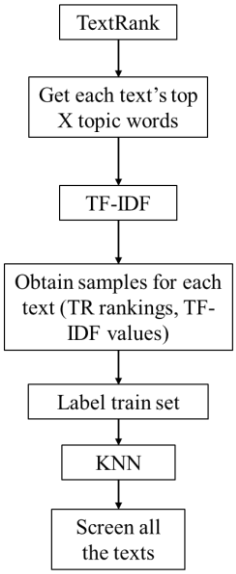

Fig. 1. The multi-model structure

III. Corpus Construction and Model Training

A. Corpus Preparation

In this research paper, we focus on two self-built corpora to evaluate the effectiveness of the multi-model for processing long-texts and short-texts. The first corpus consists of news reports related to the COVID-19 vaccine in the mainstream media of South Korea, which we refer to as "COVID-19 vaccine news." The second corpus comprises opinions about Japan discharging nuclear-contaminated water from the *Sina Weibo* platform in China, referred to as "*Weibo*". To construct these corpora, we obtained a collection of COVID-19 vaccine news articles and *Weibo* posts. We ensured the text length by randomly sampling and manually screening the content. We then labeled a certain number of samples as positive or negative.

Next, we performed segmentation, data cleansing, and stop word processing on all texts in both corpora. This preprocessing step helps to refine the data and remove irrelevant information. Afterward, we obtained a subset of 2,000 *Weibo* posts and 1,000 COVID-19 vaccine news articles for training and testing purposes.

Table I showcases texts from the two corpora before processing, while Table II displays texts after the preprocessing steps.

TABLE I. Example of Texts Before Processing

Area	Example
News	세계보건기구, WHO 가 코로나 19 확산 우려 속에 선박의 자유로운 입항 허가를 촉구했습니다. 집단감염으로 해상에 격리된 크루즈선 '다이아몬드 프린세스'호와 관련해 일본 정부와 접촉하고 있다고 도 밝혔습니다. 보도에 이종수 기자입니다.코로나 19 집단감염으로 일본 요코하마항에 정박한 크루즈선 '다이아몬드 프린세스'호.세계보건기구, WHO 는 다이아몬드 프린세스호에 탑승한 모든 승객의 건강을 지키기 위해 일본 정부와 국제해사기구, 선주 등과 지속해서 접촉하고 있다고 밝혔습니다.그러면서 코로나 19 확산 우려로 5 개 나라로부터 입항을 거부당한 크루즈선 '웨스테르담'의 입항을 캄보디아 정부가 허가했다며 선박의 자유로운 입항 허가를 촉구했습니다 …
Weibo	日本 7800 吨核污水即将入海，240 天就能到达中国沿海，1200 天后，覆盖整个北太平洋，包括中国在内的整个区域国家，没有一个国家能够幸运的避免，很多人不知道核污水的危害，核污水中含有放射性物质，会严重危害海洋生物，导致海洋生态系统的失衡，放射性物质对人体健康的危害非常大，可能会导致癌症，遗传缺陷等严重问题，在未来的 30 年里我们又该何去何从，而明天就是未来的 30 年的第一天，这一天，全世界都是命运共同体#日本核废水##日本政府正式决定福岛核废水排海##日本政府正式决定福岛核废水排海#

TABLE II. Example of Texts After Processing

Area	Example
News	세계보건기구/NNP 확산/NNG 우려__01/NNG 속__01/NNG 선박__02/NNG 자유롭/VA 입항/NNG 허가__01/NNG 촉구하/VV 집단감염/NNG 해상__02/NNG 격리되/VV 크루즈선/NNG 다이아몬드/NNG 프린세스/NNP 관련하/VV 일본__02/NNP 정부__08/NNG 접촉하/VV 밝히/VV 보도__04/NNG 이종수/NNP 기자__05/NNG 집단감염/NNG 일본__02/NNP 요코하마항/NNP 정박하/VV 크루즈선/NNG 다이아몬드/NNG 프린세스/NNP 세계보건기구/NNP 다이아몬드/NNG 프린세스/NNG 탑승하/VV 승객/NNG 건강__03/NNG 지키__01/VV 일본__02/NNP …
Weibo	7800 吨 入海 240 天 到达 中国 沿海 1200 天后 覆盖 北 太平洋 包括 中国 在内 区域 国家 国家 幸运 危害 中 含有 放射性物质 严重危害 海洋生物 导致 海洋 生态系统 失衡 放射性物质 人体 健康 危害 导致 癌症 遗传 缺陷 未来 30 年里 何去何从 明天 未来 30 年 第一天 全世界 命运 共同体

We adopt the hold-out method when dividing the dataset into train and test sets. This approach involves splitting the dataset D into two parts: the train set S and the test set T. The model is trained on S and then evaluated using T to assess its performance. To ensure the validity of the dataset, we employ stratified sampling. We divide the dataset into train and test sets using a 70:30 ratio, with 70% of the samples allocated to the train set and 30% to the test set. The specific number of texts in each corpus used in the experiment is as follows:

TABLE III. Train Set and Test Set of Each Corpus

	News Train Set	**News Test Set**	**News All**	***Weibo* Train Set**	***Weibo* Test Set**	***Weibo* All**
Amount	700	300	1000	1400	600	2000
Proportion	70%	30%	100%	70%	30%	100%
Positive Sample	350	150	500	700	300	1000
Negative Sample	350	150	500	700	300	1000

In our approach, the TextRank rankings are computed for each text, while the TF-IDF values are calculated based on the set of high-weighted words from all texts. Therefore, before training the model, we need to calculate the TextRank and TF-IDF values for all texts in the dataset, including the training set, test set, and texts to be filtered. It is important to note that the lengths of the texts may vary. In cases where the total number of processed words in a text is lower than the required TextRank ranking, we will assign a TF-IDF value of 0 to the missing words. This step ensures consistent input dimensions for the model calculation.

The TF-IDF values based on TextRank rankings do not affect the training, testing, and subsequent filtering stages. These values remain fixed throughout the process and do not change during training. After calculating the TextRank values and TF-IDF values, each sample in the dataset will have data in the format shown in Table IV.

TABLE IV. EXAMPLE OF THE SAMPLE DATA

	TextRank Ranking 1	**TextRank Ranking 2**	**…**	**TextRank Ranking m**
Sample 1	0.981	0.225	…	0.685
Sample 2	1.254	0.841	…	0.614
…	…	…	…	…
Sample n	1.115	0.687	…	1.225

B. *Model Training*

In the constructed multi-model, two hyperparameters require determination during the training process: the TextRank ranking of each text and the K value for the KNN algorithm. These hyperparameters directly impact the model's performance.

To ensure reasonable values for these hyperparameters, we restrict the TextRank ranking and K value to fall within the range of 2 to 20. This restriction is imposed to prevent overfitting, as both a TextRank ranking and a K value of 1 would lead to overfitting.

After training the multi-model, we evaluate the effects of the TextRank rank and K-value on the model's ability to screen long and short texts. The results of this evaluation are represented in Fig.2 and Fig.3, which provide insights into how variations in these hyperparameters influence the multi-model's screening capabilities.

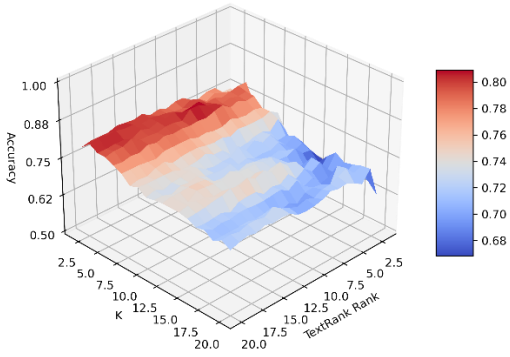

Fig. 2. The Effect of TextRank Ranking and K-Value on the Accuracy of Long Text Screening (3D Figure)

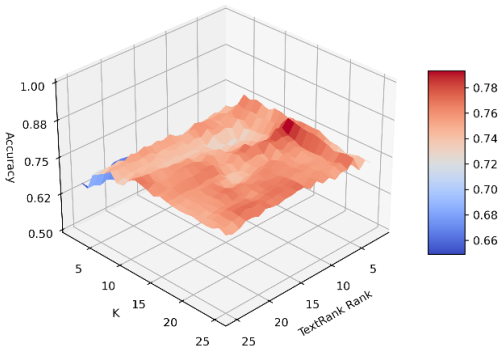

Fig. 3. The Effect of TextRank Ranking and K-Value on the Accuracy of Short Text Screening (3D Figure)

During the training process, we varied the K value and TextRank ranking from 2 to 20 and evaluated the model using the accuracy score as the evaluation metric. Since the number of positive and negative samples in the dataset is balanced, the accuracy score provides a suitable measure of the model's performance.

After conducting 361 calculations (19 values for K and 19 values for TextRank), we obtained the three-dimensional variations, which are represented in Fig.2 and Fig.3. To gain further insights, we created two projections perpendicular to the plane direction for each three-dimensional figure, resulting in Fig.4 and Fig.5.These projected figures provide a clearer visualization of how changes in the K value and TextRank ranking affect the multi-model's screening capabilities.

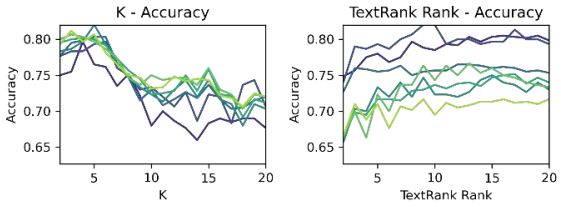

Fig. 4. The Effect of TextRank Ranking and K-Value on the Accuracy of Long Text Screening (Projection)

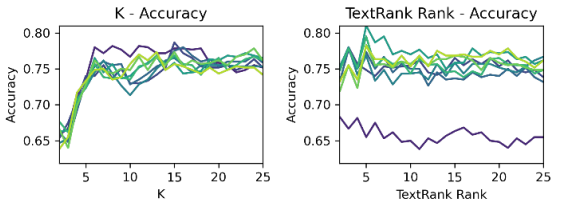

Fig. 5. The Effect of TextRank Ranking and K-Value on the Accuracy of Short Text Screening (Projection)

The combination of Fig.2, Fig.3, Fig.4, and Fig.5, selecting a K value of 10 and a TextRank ranking value of 5 for long text screening, achieves an accuracy rate of 82%. Selecting a K value of 5 and a TextRank ranking value of 15 for short text screening yields an accuracy rate of 80.17%. Once the hyperparameters are determined, the model can efficiently screen all texts.

IV. MODEL TESTING AND EVALUATING

In Section III, the computation of TextRank and TF-IDF values for the entire dataset, the division of the dataset into train and test sets, and the training of the model were

conducted. In this section, we validate the obtained KNN model using the test set and construct a confusion matrix based on the validation results.

To validate the model, we apply it to the test set and evaluate its performance. One way to assess the model's performance is by constructing a confusion matrix based on the validation results. The confusion matrix provides insights into the model's ability to classify positive and negative samples correctly. By analyzing the confusion matrix, we can evaluate metrics such as accuracy, precision, recall, and F1 score, which provide a comprehensive understanding of the model's performance on the test set.

Based on the information provided in Table I, we can see that the size of the first test set is 300 texts, and the size of the second test set is 600 texts. Using the trained KNN model, we applied it to screen both test sets. As a result, we obtained two confusion matrices, which are presented as Tables V and VI.

TABLE V. CONFUSION MATRIX OF LONG TEXT SCREENING

Confusion Matrix		**Predict Results**	
		True	False
Real Result	True	138	16
	False	38	108

TABLE VI. CONFUSION MATRIX OF SHORT TEXT SCREENING

Confusion Matrix		**Predict Results**	
		True	False
Real Result	True	247	53
	False	66	234

Based on the two confusion matrices (Tables V and VI), we can calculate the precision score (as shown in (6)), recall score (as shown in (7)), and F1 score (as shown in (8)) for both corpora. In these equations, TP represents true positives (where the model predicts positive and the actual result is also positive), TN represents true negatives (where the model predicts negative and the actual result is also negative), FN represents false negatives (where the model predicts negative but the actual result is positive), and FP represents false positives (where the model predicts positive but the actual result is negative). Using the values from the confusion matrices, we can calculate the precision score, recall score, and F1 score based on the following formulas:

$$pre = \frac{TP}{TP+FP} \tag{6}$$

$$rec = \frac{TP}{TP+FN} \tag{7}$$

$$F1 = \frac{2 \cdot pre \cdot rec}{pre+rec} \tag{8}$$

The precision score represents the proportion of true positive predictions out of all positive predictions, while the recall score measures the proportion of true positive predictions out of the actual positive samples. The F1 score combines precision and recall, providing a balanced measure of their harmonic mean.

As indicated in Table VII, the multi-model consisting of TextRank+TF-IDF+KNN demonstrates good performance in both long and short text topic screening tasks. Since an equal number of positive and negative samples were used during training, the achieved accuracy scores hold meaningful value. Furthermore, the accuracy scores for long and short texts are relatively balanced, indicating consistent performance across different text lengths.

TABLE VII. EVALUATE RESULTS

	Accuracy	**Precision**	**recall**	**F1**
News	0.82	0.8961	0.7841	0.8363
Weibo	0.8017	0.8233	0.7891	0.8058

The test results for long and short texts reveal that the precision score is generally higher than the recall score, implying a higher rate of correct recognition when identifying relevant texts. This also suggests the model's high sensitivity towards topic-related texts.

Table VIII presents an example that highlights a judgment error made by the program. In this case, both a news article and a *Weibo* post were incorrectly classified as belonging to the topic. The news article primarily focuses on the Government's decision to supplement the budget, with limited relevance to the COVID-19 vaccine. Similarly, although the *Weibo* post mentions the discharge of nuclear-contaminated water from Japan, its main focus lies on the stock market.

TABLE VIII. EXAMPLES FROM JUDGMENT ERROR

Area	**Example**
News	당정이 코로나 19(신종 코로나 바이러스) 3 차 추가경정예산의 규모와 세부사업에 협의했다. 또 디지털뉴딜과 그린뉴딜 등 '한국판 뉴딜' 육성 방향도 큰 틀에서 공감대를 이뤘다. 조정식 더불어민주당 정책위의장은 1 일 오전 국회에서 진행된 당정협의 직후 브리핑을 열고 "3 차 추경 예산 전체규모와 방향, 중점 내용에 대한 당정 간 공감대가 있었다"며 "세입경정과 국채발행 규모 모두 협의했다"고 말했다. 당정은 우선 금융고용안정패키지를 재정 측면에서 적극 뒷받침하기로 했다. 이를 위해 소상공인 긴급자금 10 조원 항공·해운·정유 주력산업 대상 채권증권시장안정펀드 조성 30 조 7 천억 비우량회사채 CP 20 조원 매입을 협의했다. 또 일자리·생계불안 등에 대한 사회적 안전망 강화대책으로 무급유직 요건 완화 등 고용유지지원금 확대 및 대상자 58 만명 추가 비대면 일자리·청년·디지털 일자리 등 55 만개 긴급일자리 공급 저신용근로자 대학생·미취업청년 금융애로 해소 예술체육인·국가유공자 보조금 확대를 결정했다. 민주당이 파악한 민생예산도 3 차 추경에 반영하기로 했다. 2022 년까지 전국 모든 공공장소 4 만 1000 곳 와이파이 단계적 설치 온누리상품권 2 조·지역사랑 상품권 3 조 추가발행 전국 유.초.중학교 대상 그린스마트학교 전환 시범사업 추진 등이다.
Weibo	今日大盘跌了 41 个点，完全不过分啊，毕竟昨天涨了那么多，今天还回去合情合理。另外，今天太平洋跌停，为什么呢？在大 A，獐子岛扇贝的事情大家耳熟能详了吧。所以呢。太平洋跌停是不是跟日本要把核废水排到太平洋有关呢？一切皆可逻辑。数字能不能继续，新炬网络能打出高度吗。核废水，环保能不能继续走强。超短的眼光就要盯在这些活跃的地方。#日本政府正式决定福岛核废水排海##资本市场# #上证指数#

It is evident that the model exhibits high sensitivity to texts containing keywords. However, it fails to capture the semantic information conveyed by the text. When building a corpus, texts are often obtained through keyword searches.

Nevertheless, the presence of keywords does not necessarily indicate that the semantic focus aligns with those keywords. Additionally, the TextRank and TF-IDF algorithms filter out irrelevant high-frequency words, but they do not fully capture the semantic context of the text.

Also, when adjusting the model hyperparameters, long texts are more likely to have their topic words highlighted after being processed by the TextRank algorithm due to their length. Therefore, only a small TextRank ranking value is needed to achieve a high accuracy rate for long texts. On the other hand, short texts often consist of only one or two sentences, and their topic words are not as prominent. Consequently, a larger TextRank ranking value is required compared to long texts in order to make a judgment. This observation also indicates the excellent compatibility of the model. By adjusting the hyperparameters, similar results can be obtained when performing the filtering task for both long and short texts.

V. Conclusion

In this paper, we propose a multi-model screening mechanism based on TextRank, TF-IDF, and the KNN algorithm to address the issue of irrelevant texts within a self-built corpus. Our approach utilizes the TextRank ranking as the foundation for calculating the TF-IDF values of each text. We select all high-weighted words, determined by the TextRank ranking, to form a set for TF-IDF calculation. Subsequently, we employ the TextRank ranking and the TF-IDF values as sample features for the KNN algorithm in the screening process. Through training and validation, our results demonstrate the effectiveness of the multi-model filtering mechanism comprising TextRank, TF-IDF, and KNN.

For scenarios where the corpus is enormous in scale and comprises disparate content but pertains to the same topic, the multi-model screening mechanism proposed in this paper effectively addresses this issue. Utilizing the multi-model mechanism can effectively alleviate the workload associated with manual judgment, enabling researchers to concentrate on corpus analysis. Furthermore, the multi-model mechanism does not require GPU resources for training and has relatively low device requirements, making it applicable in most scenarios.

However, the multi-model mechanism based on TextRank+TF-IDF+KNN is susceptible to data cleaning issues associated with the construction of TextRank and TF-IDF algorithms. These algorithms do not consider semantic information, and the presence of high-frequency and low-weight words can significantly impact the screening results. Therefore, further research can be conducted to enhance the selection of sample features and improve the model's prediction accuracy. This will aid in refining the text pre-screening mechanism of the self-built corpus.

References

[1] Yifan Zhu, Kaibao Hu. "The Semantic Preferences and Semantic Prosody of *Bei* Passives − A corpus-based contrastive study". Journal of Foreign Languages, vol. 37(01), pp.53-64, 2014.

[2] Duanyang Li, Zhijun Wang. "A Corpus-Based Multidimensional Analysis of English Register Features in Customs News". Journal of Xi'an International Studies University, vol. 27(01), pp.27-32, 2019.

[3] GULZIYRE·Aniwar, Zhihuan Kuang. "Study on the Construction of the Chinese-Kazakh Parallel Corpus". Journal of Shanxi University(Natural Science Edition), vol. 46(3), pp.537-545, 2023.

[4] Xiaobo Tang, Juan Zhu, and Fenghua Yang. "A Study on Sentiment Classification of Online Reviews Based on Sentiment Ontology and kNN Algorithm". Information Studies: Theory & Application,vol. 39(06), pp.110-114, 2016

[5] Liyuan Xiong, Yingtao Liu, Qindong Li. "An improved KNN short text classification algorithm based on category feature words". Computer Engineering & Science, vol. 40(01), pp.148-154, 2018

[6] Zhihua Wang, Shaoting Liu, Qi Luo. "KNN classification algorithm based on improved K-modes clustering". Computer Engineering and Design, vol. 40(08), pp.2228-2234, 2019

[7] Yang Song, Hailong Wang, Lin Liu, and Dongmei Pei. "Mongolian Music Classification Method Combining KPCA and Improved KNN". Journal of Fudan University(Natural Science), vol. 61(05), pp.573-580+588, 2022

[8] Mihalcea, Rada and Paul Tarau, "TextRank: Bringing Order into Text", EMNLP, 2014.

[9] Kuncoro, Bernardus Ari and Bambang Heru Iswanto, "TF-IDF method in ranking keywords of Instagram users' image captions", 2015 International Conference on Information Technology Systems and Innovation (ICITSI), pp.1-5, 2015.

[10] Wallach Hanna, Iain Murray, Ruslan Salakhutdinov, David Mimno, "Evaluation Methods for Topic Models". Proceedings of the 26th Annual International Conference on Machine Learning, pp. 105-112, 2009

[11] Qingtian Zeng, Xiaohui Hu, Chao Li, "Extracting Keywords with Topic Embedding and Network Structure Analysis". Data Analysis and Knowledge Discovery, vol. 3(07), pp.52-60, 2019