[1]Denghui Yang
[2]Dengyun Zhu
[3]Hailong Gai
[4,*]Fucheng Wan

Semantic Similarity Caculating based on BERT

Abstract: - The exploration of semantic similarity is a fundamental aspect of natural language processing, as it aids in comprehending the significance and usage of vocabulary present in a language. The advent of pre-training language models has significantly simplified the process of research in this field. This article delves into the methodology of utilizing the pre-trained language model, BERT, to calculate the semantic similarity among Chinese words. In order to conduct this study, we first trained our own model using the bert-base-chinese pre-trained model. This allowed us to acquire the word embeddings for every single word, which served as the basis for calculating semantic similarity. Essentially, word embeddings are vector-based depictions of words that encapsulate word's significance and surroundings, allowing for the measurement of the semantic similarity between words. Next, we executed a sequence of experiments to assess the efficiency of the BERT model in managing semantic similarity tasks within the Chinese language. The results were encouraging, as the BERT model demonstrated remarkable performance in these tasks. Furthermore, it was observed that the BERT model outperformed traditional methods in terms of performance and generalization capabilities. This study, therefore, underscores the potential of the BERT model in natural language processing, particularly in the Chinese language. This emphasizes the model's capacity to accurately calculate semantic similarity, paving the way for its widespread adoption in related fields.

Keywords: Jieba, BERT, Semantic Similarity, Pearson, Pre-Trained.

I. INTRODUCTION

The issue of semantic similarity in written works is a crucial component of natural language processing, and it serves a vital function in text classification, topic retrieval, programmed question and response systems, and textual summary research. The present state of research on textual semantic similarity has a profound impact on the progress of studies in allied fields.

In the context of text classification, the incorporation of semantic similarity can significantly enhance categorization precision while delving deeper into the semantic level [1]. This is achieved by identifying patterns and relationships between words and phrases that share similar meanings, allowing for more accurate classification of textual materials. In the realm of topic retrieval, semantic similarity serves as an efficient auxiliary information-based deep semantic analysis and text retrieval method [2]. By leveraging semantic similarity, retrieval systems can uncover hidden patterns and connections within the text, enabling the identification and extraction of relevant information more efficiently. In the field of automated Q&A systems, semantic similarity functions as a fundamental attribute of Q&A text, enabling the system to filter out more precise responses [3]. By analyzing the semantic similarity between questions and answers, these systems can gain a deeper comprehension of the context and significance beneath the inquiries, ultimately offering more precise and pertinent replies. In the domain of research text summarization, semantic similarity is vital for contextual comprehension and information filtering [4]. By considering the semantic similarity between different text passages, summarization algorithms can identify the most salient information and generate more coherent and accurate summaries.

Therefore, the semantic similarity of text itself represents a research direction of significant importance. By examining words across multiple texts, the relationships between them are abstracted, parsed, and processed, allowing for the mining of deeper connections from a data perspective. This approach can facilitate more accurate semantic representation of natural language, contributing to the development of a more refined understanding of the interconnectedness of textual data in various contexts. Furthermore, as the volume and diversity of digital content continue to grow, the ability to identify and leverage semantic similarity becomes increasingly critical.

[1] Key Laboratory of Linguistic and Cultural Computing Ministry of Education, Northwest Minzu University, Lanzhou, Gansu 730030, China
[2] Key Laboratory of Linguistic and Cultural Computing Ministry of Education, Northwest Minzu University, Lanzhou, Gansu 730030, China; Key Laboratory of China's Ethnic Languages and Intelligent Processing of Gansu Province, Northwest Minzu University, Lanzhou, Gansu, China
[3] Key Laboratory of Linguistic and Cultural Computing Ministry of Education, Northwest Minzu University, Lanzhou, Gansu 730030, China;
[4] Key Laboratory of China's Ethnic Languages and Intelligent Processing of Gansu Province, Northwest Minzu University, Lanzhou, Gansu, China
*Corresponding author: Fucheng Wan

This is particularly relevant in the realm of big data analytics, where the proficient handling and examination of extensive textual data can offer valuable perspectives and aid decision-making across diverse disciplines. In conclusion, the study and application of semantic similarity in written works hold great potential for advancing natural language processing research and improving the performance of various text-based applications [5]. As the domain of natural language processing keeps advancing, the creation of more complex techniques for examining and employing semantic resemblance will unquestionably contribute to a deeper comprehension and productive application of textual information across various scenarios.

II. DEVELOPMENT STATUS

The exploration of semantic similarity was a thriving area of research prior to the introduction of pre-trained models such as BERT [6], GPT-3 [7], and RoBERTa [8]. In the absence of these advanced models, the most prevalent techniques for semantic similarity tasks revolved around word embedding-based methods, with the Word2vec model [9][10] being a prime example. This model has the capability to project words into a low-dimensional space, where their semantic similarity can be determined by analyzing the resemblance of the vectors associated with these words.

Parallel to this approach, traditional machine learning techniques [11][12] and feature engineering methods [13] were extensively used to train models that measured textual similarity. These models often relied on methods such as lexical tagging and analysis of syntactic structure. Furthermore, comprehensive knowledge bases like WordNet [14], which covers a vast range of English vocabulary, were also employed. In WordNet, nouns, verbs, adjectives, and adverbs are organized into separate synonym networks. Each set of synonyms represents a fundamental semantic concept, and these sets are interconnected through various relationships. A polyglot word appears in a group of synonyms for each of its meanings, which can be used to compute semantic similarity.

Additionally, corpus-based language models that extract semantic information by analyzing co-occurrence data in text were also utilized. LSA [15] and LDA [16] are prominent examples of such models. In general, before the emergence of pre-trained language models like BERT, research on semantic similarity primarily relied on statistical methods. However, the rise of pre-trained language models has resulted in a notable improvement in the efficiency of semantic similarity tasks due to their exceptional performance and the ability to fine-tune these models for specific tasks. This allows for a better comprehension and capture of the complex semantic information within the text, leading to improved task performance. The future of semantic similarity tasks is likely to focus on the fine-tuning and advancement of pre-trained models.

III. USAGE

A. Processing of Data

As it is the processing of Chinese text, the following three main problems will be encountered:

First, Chinese participles can not be like English, there is a space as a separator, and the Chinese text contains the phenomenon of multiple meanings of the word inside, which may lead to ambiguity.

Second, the problem of granularity needs to be considered. The size of the particle size on the results of the word has a great impact. For example, the “Northwest Minzu University” for word division, we hope to get the result of word division is the “Northwest Minzu University”, but in fact, there will be “Northwest/ Minzu /University” such a word division, the meaning of the expression is not very accurate. In the division, the larger the granularity, the more accurate the meaning, and the corresponding recall is less.

Third, the recognition of new words. Nowadays is the era of rapid development of information, new words appear every day, and being able to quickly recognize new words is a special point to consider.

As the data selection is the data crawled on Weibo, the data volume is not very large, and BERT participle necessitates a significant amount of computational resources and memory, the participle speed is slow, and for Chinese participle and English participle as well, it is divided by a single word, which is not applicable to this task. So the stuttering participle tool is chosen to process the data.

B. Introduction to Bert

BERT is a type of pre-trained language model. What sets BERT apart from earlier models is that it does not employ standard single-direction language models or merely merge two single-direction language models for pre-training. Instead, BERT introduces a unique method, the masked language model (Mask LM) [17], and incorporates a deep bidirectional Transformer component to build the overall model. Consequently, the ultimate outcome is a

deep bidirectional language representation model that is capable of grasping and merging both left and right contextual data.

The essence of the BERT model lies in the encoder component of the Transformer structure [18], as illustrated in Figure 1. Within this structure, "Add & Norm" denotes the usage of residual connections and layer normalization operations; "Feed Forward" signifies the process of linear transformation. Through this linear transformation, the data is initially mapped to a high-dimensional space and subsequently mapped to a low-dimensional space. By employing this method, deeper features can be extracted. "Multi-Head Attention" breaks down the hidden state vector into several sections, forming multiple sub-semantic spaces, enabling the model to attend to information in various dimensional semantic spaces.

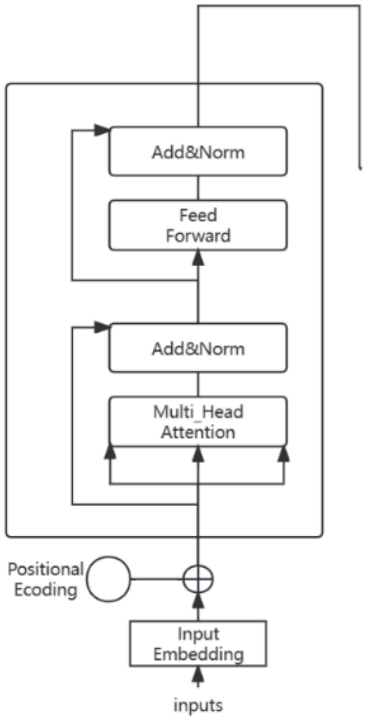

Figure 1: The Encode Part of the Transform Model

The BERT model employs a multiple stacking of the encoder section within the Transformer architecture, thereby creating a more profound neural network configuration. After multiple layers of stacking, the core part of the BERT model is formed, as shown in Figure 2.

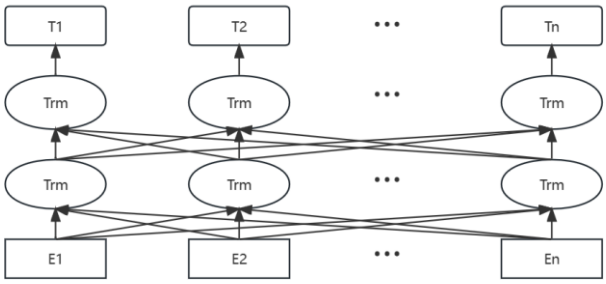

Figure 2: BERT Model's Structure

Such encoding units make up each layer in the BERT model. In the enhanced version of BERT, there are 24 levels of encoders, each level equipped with 16 attention heads, and the size of the word vector is 1024. In the smaller BERT model, there are 12 levels of encoders, each level equipped with 12 attention heads, and the size of the word vector is 768. Regardless of the model size, the feedforward layer's size is set to 4H (where H is the size of the word vector), which is 3072 when H = 768, and 4096 when H = 1024. These parameters create the hierarchical structure of the BERT model.

The BERT model can be simply summarized into three parts, which are input layer, intermediate layer and output layer.

Input Layer: In order to adapt the BERT model for downstream tasks, the input layer's statements are typically rewritten in the format of [CLS]+A+[SEP], where CLS represents a special token, denoting the classification task, and SEP acts as a separator. The input layer's embedding consists of three components: position embedding, segment embedding, and Word piece embedding. Word piece embeddings symbolize the vector representations of the words themselves. Word Piece represents the process of decomposing words into a finite collection of interchangeable subword components, intending to accomplish a harmony between preserving word viability and allowing character adaptability. Position embedding converts the position details of a word into a feature vector. As the BERT model's network structure is identical to the transformer model, there are no RNN or LSTM, making it necessary to create a position embedding. Position embeddings can be generated in two ways: the BERT model starts with a position embedding and refines it through training; in contrast, the Transformer model creates position embedding based on predefined rules. Segment embeddings serve as vector representations for distinguishing two sentences, proving particularly useful in situations with asymmetric sentences, such as in question and answer tasks. The BERT model's input components include the following: word piece token embedding, segment embedding, and position embedding.

Middle layer: the middle layer of BERT model is the same as the encoder of transformer, it is composed of self-attention layer plus ADD & BatchNorm layer plus FNN.

Output layer: each input of BERT model corresponds to one output:

C. Semantic Similarity Calculation Methods

The approach to calculating semantic similarity employs the cosine similarity algorithm [19]. This approach computes the extent of difference between two entities by ascertaining the cosine of the angle separating two vectors within a vectorial domain. After undergoing the jieba word segmentation procedure and BERT model training, each term in the text is transformed into a corresponding vector depiction. Once the word vectors of each word are obtained, the cosine similarity between every pair of words can be calculated. Assuming the word vectors of these two words are denoted as X and Y respectively. The equation for cosine similarity is as follows.

$$\cos(\vec{X},\vec{Y}) = \frac{\vec{X}\cdot\vec{Y}}{|\vec{X}||\vec{Y}|} \quad (1)$$

Vectors X and Y are associated with two distinct points in the coordinate system. Utilize the provided equation to determine the cosine value for the two vectors. As the cosine value gets closer to 1, angle comes close to 0, indicating a higher level of similarity between the two vectors. In contrast, if the cosine value is near 0, the angle leans towards 90 degrees, indicating a lower level of similarity between the two vectors.

D. Experimental Design

Step 1: Use Jieba participle to process the text. In the process of text processing, first of all, the text needs to be regularized, using regular expressions to remove non-Chinese characters. Second, for some common words such as "the", "is", "in" and so on does not carry important information, but they will increase the complexity of text processing and computational costs. Use the stop word list to filter out these irrelevant words and improve processing efficiency. In this paper, the creation of the stop word list takes into account the Chinese stop word list, the Harbin Institute of Technology stop word list, the Baidu stop word list, and the stop word library of the Machine Intelligence Laboratory of Sichuan University, and combines them into a stop word list totaling 3884 words. Stop word list [20]. Since the data comes from Weibo, some words that have no statistical significance are also added, such as "wei bo" and "zhuan fa". In addition, since new words and names often appear on the Internet, the Jieba library cannot recognize these words. For example, "Yi Yang Qian Qi" may be split into "Yi / Yang / Qian Qi" during the Jieba word segmentation process. Therefore, adding these words to jieba's vocabulary can obtain more accurate word segmentation results.

Step 2: Use BERT to train the results. In this experiment, we use "AutoTokenizer" to transform the text into model input, convert the text into pytorch token type data, and then use "AutoModel" model for training. Considering the input length in the BERT model is limited to a maximum of 512 tokens, in the experiment, because the input text is too long, it is also necessary to truncate the text, the approach employed involves incorporating a sliding window, which serves to partition the text into numerous intersecting segments, and feed each segment into the model, and then finally summarize the results of the training of all models. In this experiment, the size of the sliding window is set at 512 tokens, with a 128-token overlap.

Step 3: Find, for each word, the five most acrostic words. Firstly, we need to pass the encoded text to the BERT model, and then extract the word embedding of the model's output using the last_hidden_state () function. Next,

for every word, compute its cosine similarity against all other words and ultimately identify the top 5 words with the greatest similarity to it.

Step 4: In an effort to offer a more extensive and profound illustration of the BERT model's performance potential, a comparative study is organized and executed. The process involves the training of our in-house model through the word2vec methodology, which has been previously utilized. To maintain a balanced and precise comparison, the research is centered on finding and retaining the results for the top five most analogous words corresponding to each given word.

Step 5: Upon the completion of the training phase, the outcomes derived from both the BERT model and the word2vec model are meticulously examined and juxtaposed. In order to gauge the precision and dependability of each model, the average correlation coefficient of the associated words for each word is calculated. This measurement serves as a marker of the model's ability to identify semantic connections between words. In conclusion, the model yielding the higher average correlation coefficient is determined, signifying the model that generates more precise outcomes. This comparison enables us to evaluate the efficacy of the BERT model in comparison to the word2vec model, offering important insights into the operational efficiency of these cutting-edge natural language processing techniques.

IV. Results of the Experiment

A. *Assessment Methodology*

In this experiment, the Pearson correlation coefficient was selected as the evaluation method. This coefficient is used to determine the intensity of the correlation between two vectors. Its magnitude can span from -1 to 1. The mathematical expression of the Pearson correlation coefficient can be derived from its calculation equation:

$$r = \frac{\sum(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum(X_i-\bar{X})^2\sum(Y_i-\bar{Y})^2}} \quad (2)$$

Following the computation of the Pearson correlation coefficient, we can evaluate the correlation intensity between two words based on the range in Table 1.

Table 1: Pearson's Correlation Coefficient Strength Comparison Table

Absolute value of correlation coefficient	Correlation Intensity
0.8 to 1.0	Very robust correlation
0.6 to 0.8	robust correlation
0.4 to 0.6	Moderately robust correlation
0.2 to 0.4	Mildly correlation
0.0 to 0.2	Very mildly correlation
0.0	No correlation

B. *Results of the Experiment*

In this study, the model's performance is assessed utilizing the Pearson correlation coefficient. By finding the 5 most relevant words for each word and by calculating the correlation coefficient between them, we can get so the average of Pearson correlation coefficient of all words is 0.7578, which shows that the correlation between each word and their 5 most similar words is very strong. On this basis, 4 sets of comparison experiments were done to investigate the change of their average Pearson correlation coefficient by decreasing the number of words between each word and their most similar words in turn. The findings from the experiments are shown in Table 2.

Table 2: The Result of Bert Model Training

Number of similar words found by BERT model	Average correlation coefficient
1	0.8367
2	0.8073
3	0.7870
4	0.7708
5	0.7578

Second, with the aim to further confirm the efficiency of the BERT model, this experiment was also compared with previous models that can calculate semantic correlation, such as Word2Vec. The average Pearson correlation coefficient between each word and the five words they are most similar to was calculated using the Word2vec model, and the calculation was performed again after reducing the number of similar words, and the outcomes achieved are shown in Table 3.

Table 3: The Result of Word2vec Model Training

Number of similar words found by Word2Vec model	Average correlation coefficient
1	0.3615
2	0.3661
3	0.3763
4	0.3837
5	0.3903

C. Analysis of the Results of the Experiment

In this experiment, the Pearson correlation coefficient between each word and its five most similar words can be obtained as 0.7578, which demonstrates a notable association between the two. When the number of similarity words to be found decreases, the average Pearson's correlation coefficient between each word and its most similar words increases, indicating that they are also more similar to each other. When looking for the most similar one or two, the Pearson correlation coefficient between them reaches more than 0.8, and there is an extremely strong correlation between them. There are two possible reasons for this situation.

First: the dataset used for the experiment is not large enough, and when searching for similar words there may be some words whose similarity is not too high will be added, and thereby influencing the magnitude of the Pearson correlation coefficient.

Second: Semantic correlation is usually affected by the local context and polysemy of words. A single word may exhibit various meanings in distinct contexts, and this polysemy makes the semantic correlation between words more complicated. Pearson's correlation coefficient usually fails to capture such subtle semantic relationships.

By comparing the results of the Pearson's correlation coefficient obtained from the BERT model and the Word2vec model, it is evident that the BERT model exhibits exceptional performance in analyzing semantic correlation, which significantly outperforms the traditional methods. It can be found that the complexity and polysemy of semantic similarity can be better captured by the BERT model. The factors leading to this situation can be clarified as follows:

Initially, word2vec encoding is a fixed depiction of a word, indicating that numerous synonyms possess identical vector representations. For instance, considering the word "apple," in the context of fruit, it denotes a specific type of fruit, while in the context of the technology corporation, "Apple," it signifies a completely separate entity. In spite of their discrepancies in meaning, word2vec encoding would generate the same outcome for both. Conversely, BERT encoding is mutable and considers contextual information to extract features. This mutable method addresses the issue of polysemy that emerges in word2vec. Polysemy refers to the occurrence where a single word has multiple meanings, which can result in misunderstandings in static depictions like word2vec.

Additionally, when juxtaposed with the BERT model, the word2vec model provides a single vector representation for each word. The BERT model, nevertheless, models tokens and sentences simultaneously and includes position encodings. This discrepancy in method enables BERT to comprehend the context in which words are employed, which ultimately allows for more precise representations and interpretations of language.

V. CONCLUSION

Compared with word2vec, we can get a conclusion that the BERT model has demonstrated its exceptional capabilities in examining semantic correlation, surpassing conventional techniques and furnishing researchers and developers with a robust natural language processing instrument.

The development prospects of semantic similarity based on BERT in natural language processing are promising. This is demonstrated in several aspects as follows:

Enhanced Semantic Understanding: As a pre-trained language model, BERT possesses strong semantic understanding capabilities and can capture contextual and semantic relationships in text. By utilizing BERT to calculate semantic similarity, it can more accurately reflect the meaning similarity between texts, thus improving the performance of semantic similarity tasks.

Wide Applications: BERT-based semantic similarity research can be extensively utilized in numerous natural language processing tasks, for example it can be used in text categorization, sentiment evaluation, information extraction, and machine-assisted translation. In such tasks, the computation of semantic similarity is instrumental in improving the precision and resilience of the models.

Cross-domain Applications: In addition to natural language processing, semantic similarity research based on BERT can also be applied to other fields, such as recommendation systems, knowledge graphs, and bioinformatics.

In these domains, semantic similarity calculation also holds significant importance and can enhance the efficacy and applicability of the models.

Model Improvement and Novelty: As research deepens, optimizations and improvements can be made to the BERT model to adapt to the needs of different scenarios and tasks. For example, efforts can be made to compress and prune BERT, reducing model complexity and improving computational efficiency. Alternatively, BERT can be combined with other models to fully leverage their respective strengths and improve the performance of semantic similarity tasks.

Cross-lingual Research: Semantic similarity research based on BERT can also be extended to cross-lingual domains, investigating semantic similarity between different languages. This can promote the development of cross-lingual information retrieval, machine translation, and other tasks, enhancing the cross-lingual transfer capabilities of models.

In conclusion, semantic similarity research based on BERT holds great development prospects and is expected to achieve significant breakthroughs within the domain of natural language processing.

ACKNOWLEDGMENT

This work is supported by the Fundamental Research Funds for the Central Universities (NO. 31920230004).

REFERENCES

[1] Kenter T, Borisov A, De Rijke M. Siamese cbow: Optimizing word embeddings for sentence representations. arXiv preprint arXiv:1606.04640, 2016.

[2] Fucheng Wan. Medical Information Extraction Technology Based on Association Rules. Indian Journal of Pharmaceutical Sciences,2018[3].

[3] Fucheng Wan, Dongjiao Zhang, Lei Zhang, Ao Zhu. Question Similarity calculating method towards medical question answering system, basic clin pharmacol,2021,127(3):278-293

[4] Ribeiro M T, Singh S, Guestrin C. Semantically equivalent adversarial rules for debugging NLP models. Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: long papers). 2018: 856-865.

[5] Wan Fucheng,Yang Yimin, Zhu Dengyun, et al. Semantic Role Labeling Integrated with Multilevel Linguistic Cues and Bi-LSTM-CRF. Mathematical Problems in Engineering, 2022

[6] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[7] Floridi L, Chiriatti M. GPT-3: Its nature, scope, limits, and consequences. Minds and Machines, 2020, 30: 681-694.

[8] Liu Y, Ott M, Goyal N, et al. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

[9] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[10] Church K W. Word2Vec. Natural Language Engineering, 2017, 23(1): 155-162.

[11] Kusner M, Sun Y, Kolkin N, et al. From word embeddings to document distances. International conference on machine learning. PMLR, 2015: 957-966.

[12] Turney P D, Pantel P. From frequency to meaning: Vector space models of semantics. Journal of artificial intelligence research, 2010, 37: 141-188.

[13] Manning C, Schutze H. Foundations of statistical natural language processing. MIT press, 1999.

[14] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems, 2013, 26.

[15] Bond F, Foster R. Linking and extending an open multilingual wordnet. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2013: 1352-1362.

[16] Guo C, Lu M, Wei W. An improved LDA topic modeling method based on partition for medium and long texts. Annals of Data Science, 2021, 8: 331-344.

[17] Salazar J, Liang D, Nguyen T Q, et al. Masked language model scoring. arXiv preprint arXiv:1910.14659, 2019.

[18] Studebaker G A. A" rationalized" arcsine transform. Journal of Speech, Language, and Hearing Research, 1985, 28(3): 455-462.

[19] Schütze H, Manning C D, Raghavan P. Introduction to information retrieval. Cambridge: Cambridge University Press, 2008.

[20] Wan Fucheng , Yang Fangtao , Wu Titantian ,et al. Chinese shallow semantic parsing based on multilevel linguistic clues. Journal of Computational Methods in Sciences and Engineering, 2020(2):1-10.DOI:10.3233/JCM-194111.