ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MEASUREMENT OF SEMANTIC TEXT SIMILARITY

ZHENGFANG HE[1], CRISTINA E. DUMDUMAYA[2], VAL A. QUIMNO[3]
[1,2,3]University of Southeastern Philippines, College of Information and Computing, Davao City, Philippines
E-mail: [1]zhengfang_he@usep.edu.ph

ABSTRACT

Semantic text similarity measurement is fundamental in natural language processing (NLP). With the advancement of NLP technology, the research and application values of similarity measurement have become prominent. This paper utilizes Google Scholar as the primary search tool and collects 179 documents. Then, using filtering technology, 50 key documents are ultimately obtained. Furthermore, this paper summarizes the research progress of semantic text similarity measurement and develops a more comprehensive classification description system for text similarity measurement algorithms. The classification includes string-based, corpus-based, knowledge-based, deep learning-based, traditional pretraining-based, and state-of-the-art pretraining-based methods. For each method, this paper introduces typical models and methods and discusses the advantages and disadvantages of these approaches. The systematic research on text similarity measurement methods enables a quick grasp of these methods, summarizing and analyzing classic and the latest research in text similarity measurement. The paper also lists evaluation indicators in this field and concludes by discussing potential future research directions. The aim is to provide a reference for related research and applications.

Keywords: *NLP, Text Similarity, Semantic Similarity, Similarity Measurement, Deep Learning, Pretraining*

1. INTRODUCTION

Natural language processing (NLP) is a discipline of study that investigates how humans and computers interact. In recent years, a growing number of researchers have heavily invested in NLP to make more efficient use of information resources and improve the quality of information [1]. NLP can be divided into two tasks: natural language generation and understanding [2]. The measurement of text similarity belongs to the latter, which aims to understand the similarity of two texts [3]. Text similarity is comparing a text with another and finding their similarities. It is basically about determining the degree of closeness of the text. In NLP, determining whether the semantics of two documents are identical is a fundamental and extensive task that enables computers to comprehend human language [4].

In numerous contexts, text similarity processing technology is widely employed [5, 6]. In information retrieval [7-9], for instance, text similarity technology can organize user's search results in real-time to obtain more accurate results. When applied to an automatic question-answering system [10-12], this technology can automatically identify the user-searched queries and match them with the system database to produce the most relevant answer. Using this technology for translation [13, 14], the veracity of the translation between the source and the target sentences can be determined. Also, this technology can be used in the process of automatically generating abstracts [15, 16], to compare the generated abstract and the original abstract.

This paper employs Google Scholar as the primary search tool and gathers 179 documents. Through the application of filtering technology, 50 key documents are ultimately selected. This paper draws on the classification framework of Gomaa et al. [17] and expands the classification system. The system primarily includes string-based, corpus-based, knowledge-based, deep learning-based, traditional pretraining-based, and state-of-the-art pretraining-based approaches. Unlike the existing review literature, this paper summarizes traditional methods and specifically focuses on the latest progress in text similarity calculation based on deep learning and pre-training models.

2. METHODOLOGY

2.1 Searching

The search procedure is implemented to extract and aggregate pertinent scholarly works and literature. In this research, Google Scholar serves as the primary search tool. This phase encompasses the following steps:

1st. The identification of terminology corresponding to the research inquiries.

2nd. The compilation of all pertinent key-term alternatives.

3rd. The conjunction of these search terms using Boolean AND/OR operators.

Consequently, a search term is (Semantic Text Similarity OR Text Similarity) AND (Levenshtein Distance). Following the execution of the search process, a total of 179 research papers is amassed.

2.2 Filtering

The corpus undergoes a meticulous two-stage filtration process to ascertain the final selection of papers eligible for comprehensive analysis. A visual representation of this process is elucidated in **Figure 1**, delineating the quantities of papers retained and excluded at each stage.

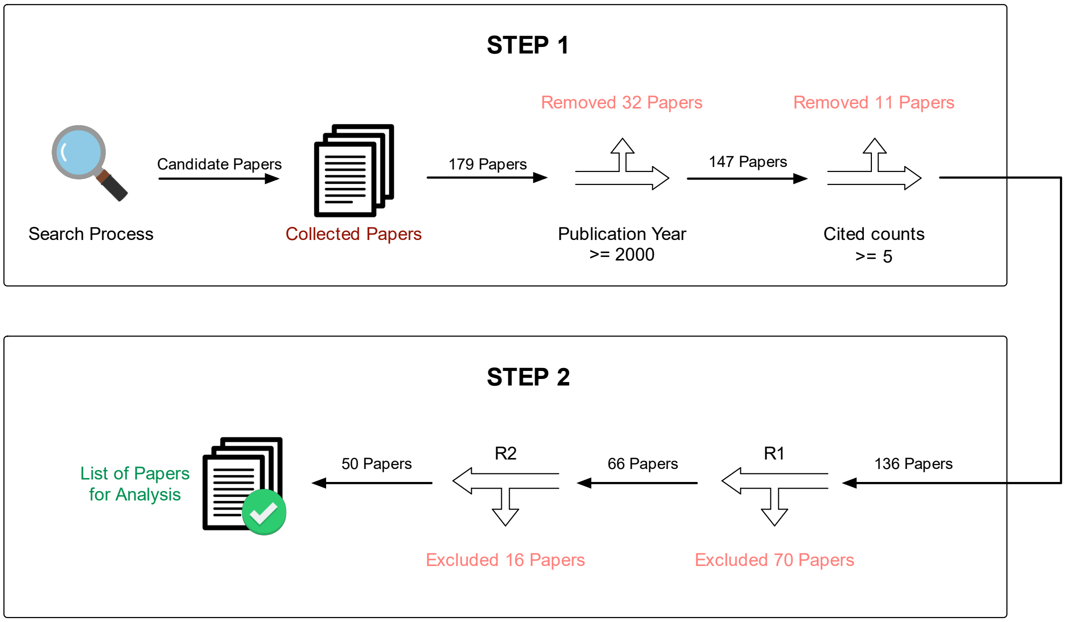

Figure 1: Two-Step Filtering Process.

In the initial step, 32 papers published before 2000 are eliminated, and 11 papers with less than 5 citations are eliminated, thus diminishing the compilation to 136. Subsequently, the second phase entails the application of inclusion and exclusion criteria, as documented in **Table 1**. Notably, the application of criterion R1 leads to the removal of 70 papers concentrating on text similarity, while the application of R2 facilitates the exclusion of 16 papers characterized as non-algorithmic. Consequently, a final selection of **50** papers is retained for comprehensive analysis and scrutiny.

Table 1: The rules of inclusion/exclusion criteria.

Rules ID	Inclusion Criteria	Exclusion Criteria
R1	Papers are deemed pertinent to the research inquiries by examining their titles, keywords, and abstracts.	Papers lacking relevance to any of the predefined research questions.
R2	Papers that have introduced an algorithm innovation, system, application, or prototype of an information technology product.	Papers that lack an algorithm aspect and do not introduce any information technology system, application, or product.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.3 Visualization

This paper uses wordcloud to visualize the titles of the 50 filtered papers, as shown in **Figure 2**.

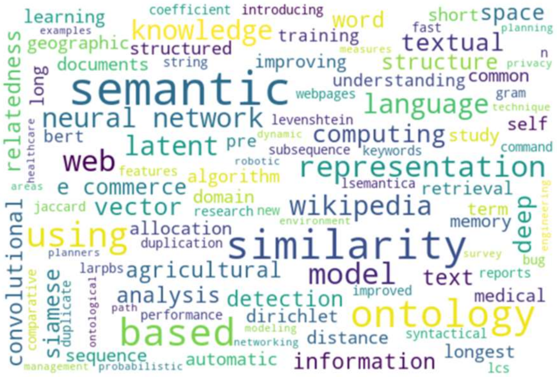

Figure 2: Visualization.

As depicted in **Figure 2**, the prominent keywords in the 50 filtered papers are 'similarity' and 'semantic,' signifying that the themes of the filtered papers align with this paper.

2.4 Classification

This paper builds upon the classification framework proposed by Gomaa et al. [17] and extends the classification system. It offers a thorough examination of semantic text similarity, systematically organized based on methods and techniques, including string-based, corpus-based, knowledge-based, deep learning-based, traditional pretraining-based, and state-of-the-art pretraining-based approaches, as illustrated in **Figure 3**.

Many specific references are thoughtfully provided for each of these technologies, offering readers a comprehensive resource for further exploration. Furthermore, the merits and drawbacks of each technology are succinctly summarized, affording a clear understanding of the strengths and limitations inherent to each approach.

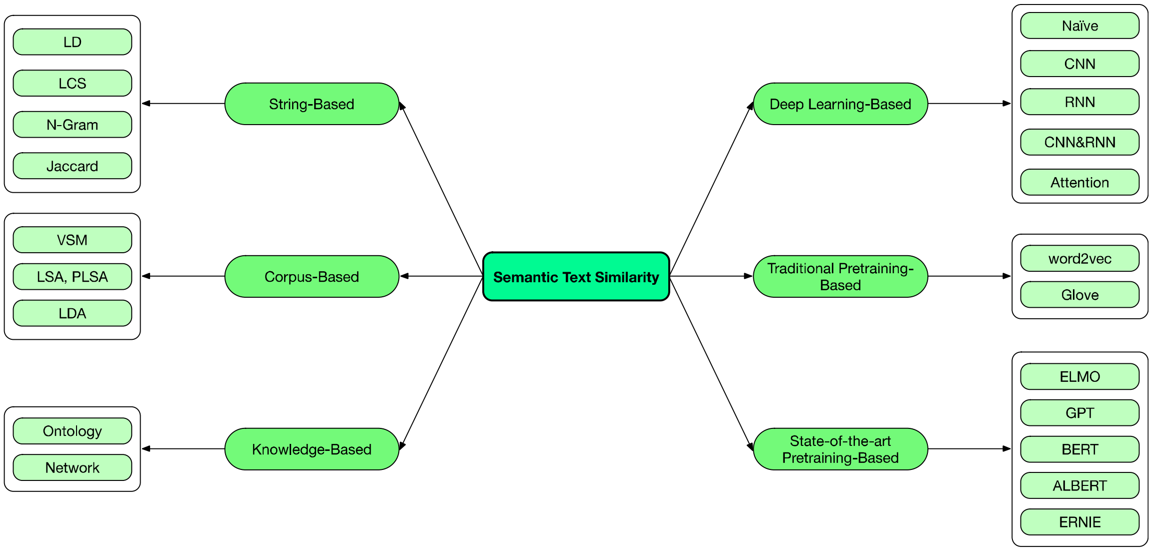

Figure 3: The Classification of semantic text similarity.

3. STRING-BASED

String-based methods, such as Levenshtein Distance (LD) [18], Longest Common Sequence (LCS) [19-21], N-Gram [22], and Jaccard [23], to compare the original texts directly.

The LD, also known as the Edit Distance (ED), is a string metric for measuring the difference between two sequences. The LD between two words is the minimum number of single-character edits (i.e., substitutions, insertions, or deletions) required to change one word into the other. A smaller LD indicates a higher similarity between the strings.

The LCS refers to the longest subsequence shared by two sequences. It is important to note that a subsequence doesn't need to occupy consecutive positions in the original sequences. For instance, consider the sequences [*wxyz*] and [*wyxwz*]. They have five common subsequences with two characters: [*wx*], [*wy*], [*wz*], [*xz*], and [*yz*]; two common subsequences with three characters: [*wxz*] and [*wyz*]; and nothing else. Therefore, [*wxz*] and [*wyz*] are their LCSs.

The basic idea of the *N-Gram* algorithm is to set a sliding window of size *N*. The text content is processed using the sliding window approach, either based on the character or word streams. After sliding, multiple text fragments of length *N* are generated. Each fragment is called an *N-tuple*. The algorithm calculates the ratio of the number of *N-tuples* in the given two texts to the total number of *N-tuples* to characterize the similarity of the two texts.

The Jaccard coefficient is the ratio of elements in the intersection set to the number of elements in the union set. It only pays attention to the same elements in the two sets and does not pay attention to the differences between the two sets. The text may be comprehended as a collection of words or as an assemblage of *N-tuples*. The Jaccard coefficient between the two sets is calculated to characterize the similarity of the two texts.

The string-based methods are simple in principle and easy to implement; they directly compare the original texts, commonly used for fast fuzzy text matching. The main disadvantage is that the meaning of words and their relationships are not considered, and issues such as synonyms and polysemous words cannot be addressed. Currently, string-based methods are rarely used alone to calculate text similarity. However, their calculation results are employed as features to characterize text within more complex methods.

4. CORPUS-BASED

Calculating similarity based on a corpus involves examining text similarities via comparing representations with a collection of text. This technique was derived from Harris's distribution hypothesis in 1954. This hypothesis posited that words with similar contexts should have similar semantics [24]. The semantic similarity between words is determined by analyzing the frequency of their co-occurrence within a given text. Currently, the primary idea of representing the text as a computer-operable vector based on the corpus uses statistical approaches to determine text similarity. Various methods for creating vectors include the Vector Space Model (VSM) [25, 26], Latent Semantic Analysis (LSA) [27], Probabilistic Latent Semantic Analysis (PLSA) [28], and Latent Dirichlet Allocation (LDA) [29, 30].

4.1 Vector Space Model (VSM)

VSM treats documents as a collection of independent feature items $(d_1, d_2, \ldots, d_n)$, and assigns weights $(w_1, w_2, \ldots, w_n)$ to each feature item according to its importance in the documents. Using $(d_1, d_2, \ldots, d_n)$ as the axes and $(w_1, w_2, \ldots, w_n)$ as the corresponding values in an n-dimensional coordinate system, cosine similarity is then employed to calculate text similarity. For large-scale corpora, VSM generates high-dimensional sparse matrices, increasing computational complexity. Additionally, VSM assumes that each feature word in the text exists independently, which separates the relationship between words and paragraphs.

4.2 Latent Semantic Analysis (LSA) and Probabilistic Latent Semantic Analysis (PLSA)

LSA is a computational model that is derived from the VSM. Both approaches use spatial vectors for text representation. However, LSA utilizes a latent semantic space and applies the Singular Value Decomposition (SVD) technique to handle high-dimensional matrices and eliminate noise in the initial vector space. Nonetheless, due to the use of SVD, the computational complexity increases. Hofmann [28] introduced the topic layer based on LSA, applied Expectation Maximization (EM) to train the topics, and developed an improved PLSA algorithm. In PLSA, polysemous words are trained on different themes, while synonyms are trained on the same themes. This approach helps to mitigate the influence of polysemous words and synonyms. However, the model parameters of PLSA will grow as documents increase.

4.3 Latent Dirichlet Allocation (LDA)

LDA is a probabilistic model that falls under generative statistical models based on Bayesian network principles. The described technique is a modeling approach for discrete data extracting topic information from a substantial corpus. Text similarity is achieved by computing the probability distribution related to the relevant subject. The limited number of representative words in short texts may hinder the ability of LDA to provide the desired outcomes in topic mining. Consequently, LDA is better suited for analyzing longer texts.

There are numerous benefits associated with corpus-based models. Firstly, semantic context: These models effectively capture the semantic context of words and texts, enabling a more comprehensive comprehension of similarity beyond mere word matching. Additionally, Unsupervised learning models are often used in unsupervised learning, whereby the utilization of labeled training data is not required. However, there are drawbacks to corpus-based models as well. Firstly, the performance of these models is contingent upon the quality and representativeness of the corpus. The presence of biases and constraints within the corpus can impact the outcome of similarity assessments. Secondly, the Bag-of-Words assumption is often used in these models, where texts are seen as collections of words without considering the sequential arrangement and syntactic structure of sentences. This may restrict their capacity to grasp intricate semantic links.

5. KNOWLEDGE-BASED

The knowledge-based methods utilize a knowledge base with a standardized organization system to calculate text similarity, generally divided into ontology-based knowledge [31] and network-based knowledge [32].

5.1 Ontology-Based

Ontology-based knowledge generally uses the relationship between concepts in the ontology structure system. If the concepts are semantically similar, there is only one path between the two concepts. The ontology used in the text similarity calculation method is not a strict concept but refers to a wide range of dictionaries, thesaurus, vocabulary, and narrow ontology. Since Berners-Lee et al. [33] introduced the concept of the Semantic Web, ontology has emerged as the primary method for knowledge modeling in this domain. The most commonly used ontology is a general-purpose dictionary, such as WordNet. In addition to dictionaries, there are also domain ontologies, such as medical ontology [34, 35], e-commerce ontology [36-38], geographic ontology [39, 40], and agricultural ontology [41-44].

The ontology-based method can reflect the internal semantic relations of concepts. Ontology generally requires experts to participate in the construction, which consumes much time. Using ontology to calculate text similarity, start by calculating at the word level and then accumulate the word similarities to obtain the similarity of the long text. Compared with corpus-based methods, the calculation efficiency of long texts is low. Whether a general ontology or a domain ontology, the ontologies are independent of each other, which is not conducive to calculating text similarity across domains.

5.2 Network-Based Calculations

In the algorithm based on network knowledge, the entries are structured, and hyperlinks connect the entries. Computers can better understand this way of information organization. Paths between concepts or links between terms become the basis for text similarity calculation. According to scholarly research, Wikipedia has emerged as a widely used platform for obtaining network-related knowledge [45]. Wikipedia is widely recognized as the biggest multilingual and openly accessible online encyclopedia. Wikipedia is recognized for its well-organized content, also referred to as semi-structured knowledge [46]. Generally, there are three categories of representative algorithms. They are WikiRelate! [47] (which is based on WordNet [48]), explicit semantic analysis (ESA) [49], and Wikipedia Link-based Measure (WLM) [50].

Network-based methods, exemplified by algorithms utilizing network knowledge from platforms like Wikipedia, offer advantages in structured information, enhancing computer comprehension and enabling accurate text similarity calculations. However, they depend on external sources like Wikipedia, which may not equally cover all domains or contain biased information.

6. DEEP LEARNING-BASED

Since the introduction of distributed word vectors in 2013, methods based on deep learning have generated numerous outstanding works in the field of semantic text similarity. Deep learning-based models are currently the most efficient. The primary algorithm used for semantic text similarity calculation is the Siamese Neural Network [51, 52], and its typical architecture is shown in **Figure 4**.

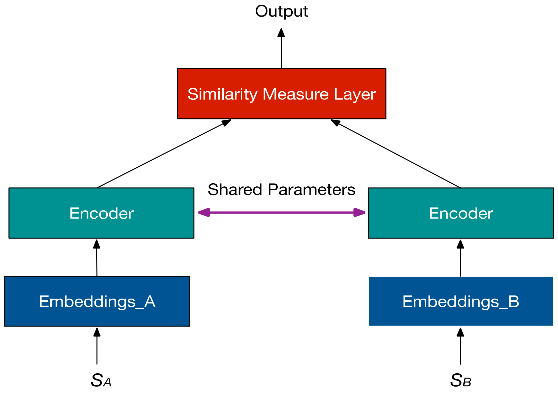

Figure 4: The typical architecture of Siamese neural network

The three layers of the Siamese Network architecture are the input, encoding, and similarity measurement layers. The input layer transforms words into word vectors before feeding them to the encoding layer. The encoding layer encodes the word vectors to yield the sentence vectors. The similarity measurement layer calculates the degree of similarity between two sentence vectors. The similarity can be calculated directly using Euclidean Distance or Cosine Similarity. Alternatively, similarity can be measured indirectly by concatenating two sentence vectors and forwarding them to other classifiers. The "Siamese" consists primarily of the sentence pair S_A and S_B being simultaneously input into the left and right networks. Both networks share the same architecture and parameters. The Encoder can be classified into different types, including Naïve-Based, Convolutional Neural Network-Based (CNN-Based), Recurrent Neural Network-Based (RNN-Based), Convolutional Neural Network and Recurrent Neural Network-Based (CNN&RNN-Based), and Attention-Based.

6.1 Naïve-Based

The encoder of the Naïve-based deep learning algorithm uses a fully connected neural network, which is a fundamental artificial neural network model. This type of neural network is a feedforward neural network consisting of multiple layers of neurons, where each neuron in one layer is connected to every neuron in the subsequent layer, hence the name "fully connected." as shown in **Figure 5**.

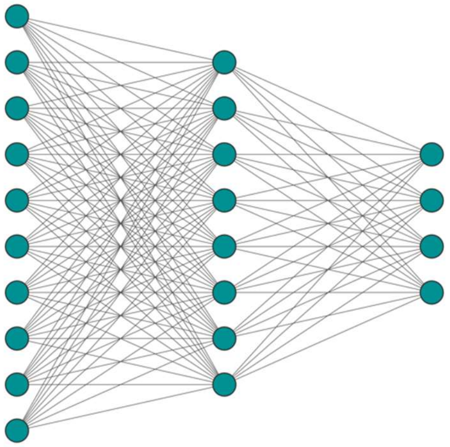

Figure 5: Fully connected neural network

The Deep Structured Semantic Models (DSSM) model proposed by Huang et al. [53] was one of the Naïve-based models. It represents one of the earliest algorithms to utilize the Siamese Network architecture for semantic text similarity calculation. The DSSM architecture is mainly composed of the input layer, presentation layer, and matching layer. This architecture is also the most commonly used architecture based on the Siamese network. The input layer maps the original text to a vector. The presentation layer maps a high-dimensional sentence vector to a low-dimensional vector, the DSSM uses 4-layer fully connected neural networks, and finally, the sentence is mapped to a 128-dimensional vector. The matching layer calculates the cosine similarity of two low-dimensional sentence vectors to characterize the semantic similarity of the two sentences.

The DSSM model has achieved outstanding results in text-matching tasks than the previous latent semantic models (e.g., LSA). However, the DSSM architecture consists of only three main layers - input, presentation, and matching layers. While simplicity can be beneficial, capturing intricate semantic relationships in text data needs more complexity. Moreover, the model's architecture may struggle to capture long-range dependencies or intricate contextual relationships in longer texts.

6.2 Convolutional Neural Network-Based (CNN-Based)

The Convolutional Neural Network (CNN) is one of the representative algorithms of deep learning and exhibits outstanding performance in massive image processing. A typical CNN architecture is shown in **Figure 6**, which includes several convolutional and pooling layers. This specific structure is well-suited for two-dimensional data. The convolutional neural network requires fewer parameters than a feed-forward neural network, so the neural network can be defined very deeply, improving the model's generalization ability [54].

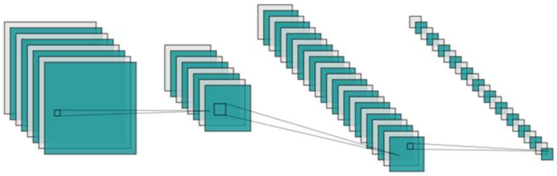

Figure 6: Convolutional neural network

Shen et al. [55] added CNN to the DSSM model to obtain more contextual information. This method mainly improves the presentation layer of DSSM, which adds a convolutional layer and a pooling layer to the presentation layer. This makes the context information more preserved, but the long-distance context information will still be lost due to the limitation of the convolution kernel.

Shao et al. [56] proposed a simple model-based convolutional neural network system. They use CNN to transfer GloVe [57] word vectors. Then, it calculates a semantic vector representation of each sentence by max-pooling. After that, it generates a semantic difference vector by absolute difference and multiplication of their semantic vectors. Next, it uses a fully connected neural network to calculate similarity scores. This model has achieved high accuracy in the English data set. However, this model is marked by simplicity, dependency on specific architectural choices, and pre-trained embeddings. These limitations suggest potential challenges in handling complex text relationships.

6.3 Recurrent Neural Network-Based (RNN-Based)

The Recurrent Neural Network (RNN) is a special network model. For a neuron node, its internal calculation data includes the previous layer's output and the same layer's output at the previous moment, as shown in **Figure 7**. Based on this unique structure, the RNN has short-term memory capabilities and preserves the relationship between data through "memory." RNN finds applications in NLP, including speech recognition, language modeling, and machine translation; it is also employed in various time-series predictions.

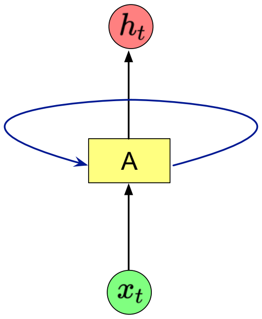

Figure 7: Recurrent neural network

To obtain more context information, Palangi et al. [58] proposed Long short-term memory DSSM (LSTM-DSSM), which added the Long short-term memory (LSTM) [59] network. This model takes into the long-distance context and word order information, improving the algorithm. The experimental result on an information retrieval (IR) task derived from the Bing web search indicates the proposed method's ability to address lexical mismatch and long-term context modeling issues, exceeding existing methods for IR tasks.

The LSTM-DSSM model addresses IR's lexical mismatch and long-term context issues by incorporating LSTM networks. However, its complexity, potential overfitting, limited generalization, and lack of contextual adaptability in shorter texts raise concerns.

6.4 Convolutional Neural Network and Recurrent Neural Network-Based (CNN&RNN-Based)

Pontes et al. [60] used both CNN and LSTM models in the Siamese architecture to calculate semantic text similarity. First, the sentence is divided into partial fragments, and then each fragment is passed through the CNN network to obtain the fragment vector. The original word vector and its corresponding context vector are then spliced together and input into the LSTM network. After obtaining the sentence vectors, the Manhattan distance between the sentence vectors represents the semantic text's similarity. This model attempts to combine CNN and LSTM for semantic similarity calculation, but it has limitations related to fragmented representation, complexity, architecture coherence, and suitability for capturing nuanced semantic relationships. Thorough evaluation and potential adjustments are needed to overcome these challenges.

6.5 Attention-Based

The encoder of the attention-based deep learning algorithm uses a self-attention [61] model. Self-Attention represents the words of the input sentence as a pair of <Key, Value>, and each word in the target sentence is called Query. Then **K** (Key), **V** (Value), and **Q** (Query) can be used to describe how to calculate *c*. The weight of each **K** corresponding to **V** can be obtained by calculating the correlation of **Q** to each **K**, which is the attention score. Then, **V** is weighted and summed to obtain the final attention vector. The calculation process is shown in **Figure 8**.

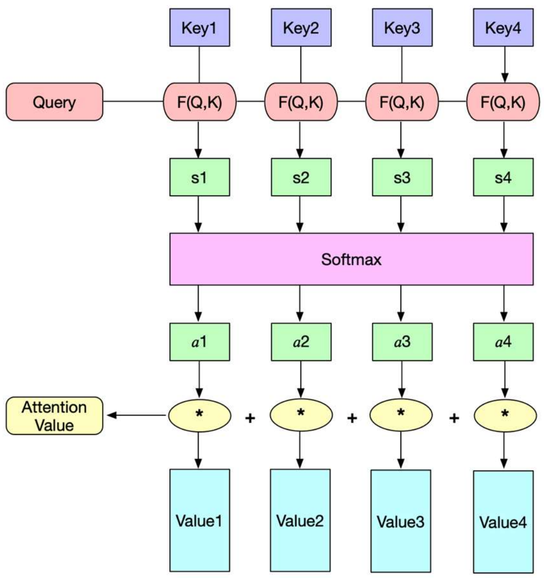

Figure 8: Self-attention

Lin et al. [62] combined bidirectional LSTM (BiLSTM) and Self-Attention technology to obtain sentence vector representation. Specifically, the sentence is passed through the BiLSTM model, and the obtained vectors in the two directions at each time step are merged into a two-dimensional matrix. The weight of each word vector in the sentence is then computed through Self-Attention. Finally, the sentence vector is obtained by the weighted summation of the word vectors.

This model also uses the Siamese architecture. After obtaining the sentence vector, simple feature extraction, such as dot product, is performed and fed into a multi-layer perceptron to obtain the final semantic text similarity. However, its shortcomings include the complexity introduced by advanced techniques, potential overfitting due to increased complexity, and limited interpretability in decision-making. These factors raise concerns about the model's efficiency, generalizability, and practicality.

The advantages and disadvantages of deep learning-based methods are presented in **Table 2**.

Table 2: The advantages and disadvantages of deep learning-based methods.

Advantages	Disadvantages
Deep learning-based approach	Data imbalance
Parameter-sharing	Difficulty in hyperparameter tuning
Flexibility in encoder choice	Interpretability challenges

As shown in **Table 2**, the advantages are (1) Deep learning-based approach: These models leverage the ability of deep learning methods, which have shown outstanding performance in semantic text similarity tasks. They can learn complex patterns and representations from text data, allowing them to capture nuanced semantic similarities between sentences. (2) Parameter-sharing: Siamese networks utilize parameter sharing between the two identical network branches, which can lead to improved generalization and reduced model complexity. (3) Flexibility in encoder choice: Siamese networks support different encoder architectures, such as Naïve-Based, CNN-based, RNN-based, CNN&RNN-based, and Attention-based. This flexibility allows researchers to explore various architectural choices and select the one that best suits the specific requirements of their task.

The disadvantages are (1) Data imbalance: When training Siamese networks for semantic text similarity, it is essential to have a balanced dataset. Imbalanced datasets with skewed similarity distributions can lead to biased model performance, particularly for less-represented similarity classes. (2) Difficulty in Hyperparameter Tuning: Challenges in Siamese neural networks for text similarity arise from the intricate architecture with complex layers, including input, encoding, and similarity measurement. The challenge lies in optimizing hyperparameters, such as the learning rate and batch size, as their interdependencies make the process intricate, demanding significant computational resources and time. (3) Interpretability challenges: The particular structure of Siamese networks makes interpreting the model's inner workings challenging. Understanding why the model assigns a particular similarity score or how it arrived at a decision can be difficult, limiting its interpretability and making it harder to diagnose errors or biases.

7. TRADITIONAL PRETRAINING-BASED

Word2vec is the earliest distributed word vector method, containing two models, Continuous Bag-of-Words (CBOW) and Skip-gram [63]. The basic idea is to determine the central word and the size of the context window. CBOW predicts the central word by the context, and Skip-gram predicts the context by the central word. Generally, word vectors are generated by self-supervised training models. The main problem with word2vec is that it can only consider local information, and the local information depends on the size of the context window.

On the other hand, the Global Vectors (GloVe) model [57], constructs a co-occurrence matrix of words through a corpus. Then, it uses the probability method to obtain the final word vector through the co-occurrence matrix. The model synthesizes the global corpus and contains part of the global information.

These models reduce the high-dimensional space of words to a lower-dimensional space (typically a few hundred dimensions). These models, like word2vec and GloVe, are trained on large corpora, which makes them general-purpose word embeddings. They can be used as the initial weights of the words for various downstream tasks. However, once the model is trained, the word embeddings are fixed. These models struggle with out-of-vocabulary (OOV) words, which are words absent in the training corpus.

CBOW's primary limitation is considering only local information, influenced by the context window size. GloVe, in contrast, constructs word vectors through co-occurrence matrices, utilizing probability methods on a corpus for a broader context. While GloVe offers more global information synthesis, it still captures only part of the global context. Both models lack a comprehensive understanding of long-range dependencies and complex semantic relationships within language, limiting their ability to represent subtle nuances in texts.

8. STATE-OF-THE-ART PRETRAINING-BASED

Peters et al. proposed the ELMO model [64], which uses a bidirectional language model and two double-layer LSTMs as encoders. The dynamic word vector is obtained by pre-training on a large corpus. Radford et al. proposed the GPT model [65], which generates word vectors by combining a one-direction language model with a Transformer with more powerful coding capabilities. Devlin et al. proposed the BERT model, which uses the Transformer and the mask mechanism [66]. At the same time, the prediction "next sentence prediction" task is added to the model, thereby generating more high-quality word vectors. This model is also one of the most commonly used word vectors currently. Lan et al. proposed the ALBERT model [67]. They present two parameter reduction techniques to reduce memory consumption and increase the training speed of BERT. Experiments show that this model has better scalability than the original BERT. Zhang et al. proposed the ERNIE model [68]. They use a large-scale text corpus and knowledge graphs to train the model, enabling it to utilize vocabulary, syntax, and knowledge information fully. Experimental results show that ERNIE has significantly progressed in various natural language processing tasks.

The strengths of these models lie in their ability to capture contextual information from large amounts of text data. They learn to represent words and sentences based on their surrounding context, facilitating understanding of complex language nuances. Additionally, these models offer transfer learning capabilities, allowing for fine-tuning specific downstream tasks with smaller amounts of task-specific data. However, these merits come with certain drawbacks. First, these models have high computational demands; training and fine-tuning large-scale pretraining models can be computationally intensive and necessitate specialized hardware, such as GPUs or TPUs. Second, interpreting these models poses a challenge; their attention mechanisms and complex architectures make it difficult to trace the exact sources of information influencing their decisions.

9. EVALUATION INDICATORS

In the semantic text similarity task, since the prediction value and label value are mostly continuous values between 0 and 5. Commonly used evaluation indicators for this task include the Pearson correlation coefficient and Spearman's rank correlation coefficient.

Pearson correlation coefficient is used to measure the correlation between two variables, X and Y. Calculate the Pearson coefficient r of the vectors X and Y according to **Equation (1)**. The value of r is the Pearson correlation coefficient. It represents the correlation between two vectors, and the value range is $(-1,1)$. The closer the value is to 1, the closer the predicted value is to the true value.

$$r = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n} (X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n} (Y_i - \bar{Y})^2}} \quad (1)$$

Spearman's rank correlation coefficient can be regarded as a kind of order or ranking, and the coefficient is solved according to the ranking position of the original data. The calculation formula is shown in **Equation (2)**. The value range of ρ is between $(-1,1)$. When evaluating the model, the ρ is closer to 1, the better the model's performance.

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)} \qquad (2)$$

10. CHALLENGES AND FUTURE RESEARCH DIRECTIONS

Challenge: Deep learning-based models have gained prominence in text similarity measurement but often need more interpretability. The recommendation for enhancing interpretability is rooted in the need for transparency and understanding in similarity measurement algorithms. Enhancing interpretability can build trust and encourage wider adoption of text similarity techniques. **Directions**: Researchers can work towards developing new interpretability methods to make the decision-making processes of deep learning-based models more transparent and understandable. This may include interpretable model architecture design, visualization techniques, and methods for interpreting internal model weights and features.

Challenge: While current methods emphasize overall similarity scores, a pressing demand exists for finer granularity in text similarity measurement. The recommendation stems from the increasing need for more detailed information in these assessments, as users and applications seek a profound comprehension of text relationships. Developing techniques for fine-grained measurement allows for a more nuanced capture and utilization of similarities. **Directions**: Researchers can explore techniques that enable the decomposition of overall similarity scores into sub-aspects or components. This involves breaking down the similarity assessment into finer categories, allowing for a more detailed understanding of specific aspects of similarity between texts. Establishing benchmark datasets and evaluation metrics designed for fine-grained similarity measurement is essential. This ensures standardized testing and facilitates the comparison of different models, fostering advancements in the field.

Challenge: The emergence of multimodal data, such as text, images, audio, and video, presents a new frontier in text similarity measurement. The basis for this recommended research is recognizing that combining different media types can unlock unique semantic similarity assessment possibilities. **Directions**: Researchers can integrate multimodal data to explore interactions and connections between textual and non-textual information, developing models that can effectively leverage multiple modalities and capture their interactions. Investigate fusion techniques to combine information from different modalities effectively. This includes exploring strategies for feature fusion, decision fusion, and attention mechanisms to optimize multimodal data integration for similarity measurement.

11. CONCLUSIONS

This paper presents a comprehensive analysis of semantic text similarity measurement methods in NLP. Classic and recent approaches are systematically examined, and a classification system is developed encompassing string-based, corpus-based, knowledge-based, deep learning-based, traditional pretraining-based, and state-of-the-art pretraining-based methods.

By researching each method, their strengths and weaknesses are emphasized. String-based approaches present simplicity and efficiency, but the text's semantics may need to be recovered. Corpus-based methods depend on large-scale text resources to extract statistical patterns but may be limited by the quality of the corpus. Knowledge-based techniques utilize knowledge base to enhance semantic understanding, but there are challenges in knowledge acquisition and expression. Deep learning-based methods are outstanding at capturing complex semantic relationships but require large amounts of annotated data and computational resources. Traditional pretraining-based approaches, such as Word2Vec and GloVe, have paved the way for the following improvements but may need to capture contextual information fully. State-of-the-art pretraining-based models like BERT and GPT have shown superior performance by leveraging transformer architectures and large-scale datasets.

Upon critical self-evaluation, several areas of improvement and unresolved questions emerge. Firstly, while Google Scholar is a valuable resource, exploring alternative academic databases could offer a more comprehensive literature review. Secondly, although our discussions on methodological strengths and weaknesses are thorough, additional in-depth analyses could contribute to a more comprehensive understanding for readers. Lastly, while our discussion on future research directions is comprehensive, a more nuanced consideration of the latest trends and challenges in the field would further strengthen the conclusions.

Acknowledging these areas for improvement and remaining open to further exploration, the aim is to enhance this study's academic and practical impact.

This research contributes to advancing the semantic text similarity measurement field and seeks to serve as a foundation for future research and applications.

REFERENCES:

[1] R. Varaprasad and G. Mahalaxmi, "Applications and Techniques of Natural Language Processing: An Overview," *IUP Journal of Computer Sciences,* vol. 16, no. 3, pp. 7-21, 2022.

[2] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia tools and applications,* vol. 82, no. 3, pp. 3713-3744, 2023.

[3] X. Liu, P. He, W. Chen, and J. Gao, "Multi-task deep neural networks for natural language understanding," *arXiv preprint arXiv:1901.11504,* 2019.

[4] F. Ahmad and M. Faisal, "A novel hybrid methodology for computing semantic similarity between sentences through various word senses," *International Journal of Cognitive Computing in Engineering,* vol. 3, pp. 58-77, 2022.

[5] M. Farouk, "Measuring sentences similarity: a survey," *arXiv preprint arXiv:1910.03940,* 2019.

[6] S. Zhou, X. Xu, Y. Liu, R. Chang, and Y. Xiao, "Text similarity measurement of semantic cognition based on word vector distance decentralization with clustering analysis," *IEEE Access,* vol. 7, pp. 107247-107258, 2019.

[7] L. Helmers, F. Horn, F. Biegler, T. Oppermann, and K.-R. Müller, "Automating the search for a patent's prior art with a full text similarity search," *PloS one,* vol. 14, no. 3, p. e0212103, 2019.

[8] Z. Yang, Z. Chen, P. Zhang, M. Liu, and Q. Li, "An information intelligent search method for computer forensics based on text similarity," in *Proceedings of the 2020 4th International Conference on Cryptography, Security and Privacy*, 2020, pp. 79-83.

[9] D. K. Po, "Similarity based information retrieval using Levenshtein distance algorithm," *Int. J. Adv. Sci. Res. Eng,* vol. 6, no. 04, pp. 06-10, 2020.

[10] Z. Jiang, C. Chi, and Y. Zhan, "Research on medical question answering system based on knowledge graph," *IEEE Access,* vol. 9, pp. 21094-21101, 2021.

[11] K. Nassiri and M. Akhloufi, "Transformer models used for text-based question answering systems," *Applied Intelligence,* pp. 1-34, 2022.

[12] Z. H. Amur, Y. Hooi, I. N. Sodhar, H. Bhanbhro, and K. Dahri, "State-of-the Art: Short Text Semantic Similarity (STSS) Techniques in Question Answering Systems (QAS)," in *International Conference on Artificial Intelligence for Smart Community: AISC 2020, 17–18 December, Universiti Teknologi Petronas, Malaysia*, 2022: Springer, pp. 1033-1044.

[13] M. Kay and M. Roscheisen, "Text-translation alignment," *Computational linguistics,* vol. 19, no. 1, pp. 121-142, 1993.

[14] H.-Q. Nguyen-Son, T. Thao, S. Hidano, I. Gupta, and S. Kiyomoto, "Machine translated text detection through text similarity with round-trip translation," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 5792-5797.

[15] M. Gambhir and V. Gupta, "Recent automatic text summarization techniques: a survey," *Artificial Intelligence Review,* vol. 47, pp. 1-66, 2017.

[16] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert systems with applications,* vol. 165, p. 113679, 2021.

[17] W. H. Gomaa and A. A. Fahmy, "A survey of text similarity approaches," *international journal of Computer Applications,* vol. 68, no. 13, pp. 13-18, 2013.

[18] S. Zhang, Y. Hu, and G. Bian, "Research on string similarity algorithm based on Levenshtein Distance," in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2017: IEEE, pp. 2247-2251.

[19] X. Xu, L. Chen, and P. He, "Fast sequence similarity computing with LCS on LARPBS," in *International Symposium on Parallel and Distributed Processing and Applications*, 2005: Springer, pp. 168-175.

[20] M. Elhadi and A. Al-Tobi, "Detection of Duplication in Documents and WebPages Based Documents Syntactical Structures through an Improved Longest Common Subsequence," *Int. J. Inf. Process. Manag.,* vol. 1, no. 1, pp. 138-147, 2010.

[21] B. S. Neysiani and S. M. Babamir, "Improving performance of automatic duplicate bug reports

detection using longest common sequence: Introducing new textual features for textual similarity detection," in *2019 5th conference on knowledge based engineering and innovation (KBEI)*, 2019: IEEE, pp. 378-383.

[22] G. Kondrak, "N-gram similarity and distance," in *International symposium on string processing and information retrieval*, 2005: Springer, pp. 115-126.

[23] S. Niwattanakul, J. Singthongchai, E. Naenudorn, and S. Wanapu, "Using of Jaccard coefficient for keywords similarity," in *Proceedings of the international multiconference of engineers and computer scientists*, 2013, vol. 1, no. 6, pp. 380-384.

[24] Z. Harris, "Distributional hypothesis," *Word World,* vol. 10, no. 23, pp. 146-162, 1954.

[25] L. Xu, S. Sun, and Q. Wang, "Text similarity algorithm based on semantic vector space model," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, 2016: IEEE, pp. 1-4.

[26] O. Shahmirzadi, A. Lugowski, and K. Younge, "Text similarity in vector space models: a comparative study," in *2019 18th IEEE international conference on machine learning and applications (ICMLA)*, 2019: IEEE, pp. 659-666.

[27] C. Schwarz, "lsemantica: A command for text similarity based on latent semantic analysis," *The Stata Journal,* vol. 19, no. 1, pp. 129-142, 2019.

[28] T. Hofmann, "Probabilistic latent semantic analysis," *arXiv preprint arXiv:1301.6705,* 2013.

[29] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research,* vol. 3, no. Jan, pp. 993-1022, 2003.

[30] V. Rus, N. Niraula, and R. Banjade, "Similarity measures based on latent dirichlet allocation," in *Computational Linguistics and Intelligent Text Processing: 14th International Conference, CICLing 2013, Samos, Greece, March 24-30, 2013, Proceedings, Part I 14*, 2013: Springer, pp. 459-470.

[31] R. Gayathri and V. Uma, "Ontology based knowledge representation technique, domain modeling languages and planners for robotic path planning: A survey," *ICT Express,* vol. 4, no. 2, pp. 69-74, 2018.

[32] D. Jones, J. Keeney, D. Lewis, and D. O'Sullivan, "Knowledge-based networking," in *Proceedings of the second international conference on Distributed event-based systems*, 2008, pp. 329-332.

[33] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific american,* vol. 284, no. 5, pp. 34-43, 2001.

[34] J. R. Simon, "Medical ontology," in *Philosophy of medicine*: Elsevier, 2011, pp. 65-114.

[35] F. Zeshan and R. Mohamad, "Medical ontology in the dynamic healthcare environment," *Procedia Computer Science,* vol. 10, pp. 340-348, 2012.

[36] A. Gómez-Pérez, M. Fernández-López, and O. Corcho, *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer Science & Business Media, 2006.

[37] M. Hecker, T. S. Dillon, and E. Chang, "Privacy ontology support for e-commerce," *IEEE Internet Computing,* vol. 12, no. 2, pp. 54-61, 2008.

[38] A. I. La Paz, A. Ramaprasad, T. Syn, and J. Vasquez, "An ontology of E-commerce-mapping a relevant corpus of knowledge," vol. 10, ed: Multidisciplinary Digital Publishing Institute, 2015, pp. 1-9.

[39] F. T. Fonseca, M. J. Egenhofer, P. Agouris, and G. Câmara, "Using ontologies for integrated geographic information systems," *Transactions in GIS,* vol. 6, no. 3, pp. 231-257, 2002.

[40] H. Bouyerbou, K. Bechkoum, and R. Lepage, "Geographic ontology for major disasters: Methodology and implementation," *International journal of disaster risk reduction,* vol. 34, pp. 232-242, 2019.

[41] M. T. Maliappis, "Applying an agricultural ontology to web-based applications," *International Journal of Metadata, Semantics and Ontologies,* vol. 4, no. 1-2, pp. 133-140, 2009.

[42] N. Bansal and S. K. Malik, "A framework for agriculture ontology development in semantic web," in *2011 International Conference on Communication Systems and Network Technologies*, 2011: IEEE, pp. 283-286.

[43] Y.-y. Wei, R.-j. Wang, Y.-m. Hu, and W. Xue, "From web resources to agricultural ontology: a method for semi-automatic construction," *Journal of Integrative Agriculture,* vol. 11, no. 5, pp. 775-783, 2012.

[44] T. Kim, N. Bae, M. Lee, C. Shin, J. Park, and Y. Cho, "A study of an agricultural ontology model for an intelligent service in a vertical farm," *International Journal of Smart Home,* vol. 7, no. 4, pp. 118-126, 2013.

[45] P. Singer *et al.*, "Why we read Wikipedia," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 1591-1600.

[46] P.-M. Ryu, M.-G. Jang, and H.-K. Kim, "Open domain question answering using Wikipedia-based knowledge model," *Information Processing & Management,* vol. 50, no. 5, pp. 683-692, 2014.

[47] M. Strube and S. P. Ponzetto, "WikiRelate! Computing semantic relatedness using Wikipedia," in *AAAI*, 2006, vol. 6, pp. 1419-1424.

[48] G. A. Miller and C. Fellbaum, "WordNet then and now," *Language Resources and Evaluation,* vol. 41, pp. 209-214, 2007.

[49] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using Wikipedia-based explicit semantic analysis," in *IJcAI*, 2007, vol. 7, pp. 1606-1611.

[50] D. Milne, "Computing semantic relatedness using wikipedia link structure," in *Proceedings of the new zealand computer science research student conference*, 2007, vol. 7, no. 8: Citeseer.

[51] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, 2015, vol. 2, no. 1: Lille.

[52] D. Chicco, "Siamese neural networks: An overview," *Artificial neural networks,* pp. 73-94, 2021.

[53] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, 2013, pp. 2333-2338.

[54] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM,* vol. 60, no. 6, pp. 84-90, 2017.

[55] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*, 2014, pp. 101-110.

[56] Y. Shao, "Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 2017, pp. 130-133.

[57] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532-1543.

[58] H. Palangi *et al.*, "Semantic modelling with long-short-term memory for information retrieval," *arXiv preprint arXiv:1412.6629,* 2014.

[59] F. Gers, "Long short-term memory in recurrent neural networks," Verlag nicht ermittelbar, 2001.

[60] E. L. Pontes, S. Huet, A. C. Linhares, and J.-M. Torres-Moreno, "Predicting the semantic textual similarity with siamese CNN and LSTM," *arXiv preprint arXiv:1810.10641,* 2018.

[61] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems,* vol. 30, 2017.

[62] Z. Lin *et al.*, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130,* 2017.

[63] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781,* 2013.

[64] M. Peters *et al.*, "Deep contextualized word representations. arXiv 2018," *arXiv preprint arXiv:1802.05365,* vol. 12, 2018.

[65] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," 2018.

[66] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805,* 2018.

[67] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942,* 2019.

[68] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," *arXiv preprint arXiv:1905.07129,* 2019.