

Received July 1, 2020, accepted July 21, 2020, date of publication August 4, 2020, date of current version August 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014185

# Entity Thematic Similarity Measurement for Personal Explainable Searching Services in the Edge Environment

**YU BAI<sup>1,2</sup>, LIANG ZHAO<sup>ID2</sup>, ZHIGUANG WANG<sup>2</sup>, JIANJUN CHEN<sup>3</sup>, AND PENG LIAN<sup>3</sup>**

<sup>1</sup>College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

<sup>2</sup>College of Computer Science and Technology, Shenyang Aerospace University, Shenyang 110136, China

<sup>3</sup>Global Envoy Software Company Ltd., Shenyang Aerospace University, Shenyang 110136, China

Corresponding author: Liang Zhao (lzhao@sau.edu.cn)

This work was supported in part by the Key Research and Development Program of Liaoning under Grant 2019JH2/10100020, in part by the National Natural Science Foundation for Young Scientists of China under Grant 61701322, in part by the Ministry of Education Project of Humanities and Social Sciences Foundation for Young Scientists of China under Grant 17YJCZH003, in part by the Young and Middle-Aged Science and Technology Innovation Talent Support Plan of Shenyang under Grant RC190026, in part by the Liaoning Natural Science Foundation under Grant 2020-MS-237, and in part by the Liaoning Provincial Department of Education Science Foundation under Grant JYT19052.

**ABSTRACT** Currently, search engines are widely used to address the information overload problem. Different from the existing client-and-server-based frameworks, edge computing (EC) technology can provide a new architecture for personalized searching services. The issue of how to measure the similarities among entities by using the context information generated by user behavior in the edge environment is vital in the task of entity-related personal searching. To analyze and measure the similarities among entities, existing methods are mainly based on either the textual content or relationships unilaterally, and the results usually have a fixed degree of similarity. However, the similarities among entities depend on the set of properties that belong to the entities. This approach should be used in determining the similarity or dissimilarity associated with the surrounding context. To address this limitation, we propose a novel semantic augmentation method with a double attention mechanism. The method refers to a dynamic representation learning process that maps an entity to a real number vector in semantic space. In this article, different from the existing similarity measurement methods, we propose a thematic similarity measure approach to analyze the connotation and denotation similarities among entities. The experimental results show that the double attention mechanism leads to a significant improvement in the entity thematic similarity measurement tasks. The model can make a separation among the entities from different domains effectively. In addition, it can take similar entities that are closer in the same domain. It also shows excellent performance on the task of entity thematic similarity, which makes the recommendation results more explainable.

**INDEX TERMS** Edge computing, entity thematic similarity, semantic augmentation, personalized searching service.

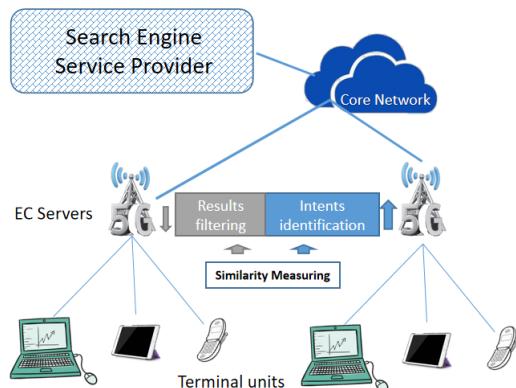
## I. INTRODUCTION

Edge computing technology gain increasingly more attention because of its special advantages, and it has been applied in various types of domains [1]–[3]. In addition, it provides a new architecture for personalized recommendation systems. Different from the existing client-and-server-based framework, personal searching services in the edge environment

The associate editor coordinating the review of this manuscript and approving it for publication was Lu Liu .

call for processing the user's intents or search results data at the edge of the network. By sinking operations such as intent identification and search results filtering into the EC servicers (or called edge cloud), the data generated by user interaction behavior is used to realize personalized computing in an edge environment. It has the potential to address the common concerns of bandwidth cost-savings as well as data safety and privacy.

The issue of how to measure the similarities among entities by using the context information generated by user behavior



**FIGURE 1.** Edge computing paradigm of personal searching services.

in the edge environment is vital in the task of entity-related personal searching.

As a branch of word semantics similarity computing, entity semantic similarity computing or similarity measuring is the essential work of entity-related personal searching in the edge environment. With the rapid growth of information retrieval [4], recommendations [5] and knowledge graph (KG) research in artificial intelligence [6], [7], estimating the similarity between entity names also plays an essential role in entity resolution [8] and entity linking [9] tasks.

As correlational research, words semantic similarity computing has long been an established research area in natural language processing and has attracted much research attention in natural language understanding, as well as in the information retrieval communities [10]. There are many ways to quantify the similarity of a pair of words. In a survey, the methods can be categorized into property similarity measures [11] and relation similarity measures [12]. For property similarity measures, the method is to measure the similarity by comparing the properties of each word. For example, the two words *missile* and *fly bomb* share many properties (e.g., both of them are weapons of destruction). In recent years, we use context words in a pretraining corpus instead of properties, and the state-of-the-art methods are to compute the similarity between two distributional word vectors [13]; this approach has been shown to perform well on semantic similarity and relatedness tasks [14], [15]. By following the distributional hypothesis, two words are assumed to be more similar if their surrounding contexts are more similar or they appear together more frequently. Thus, words with similar meanings will have vector representations that are close together in the embedding space. Accordingly, cosine similarity correlates with the cosine of the angle between vectors, and it is a popular measure approach to assess the similarity between words.

Since the distributional word vectors always are pretraining uniquely [16], this results in a fixed degree of similarity between two words. However, a similarity relation is a binary relation between objects; it is reflexive and symmetric relative. The similarity among entities depends on their

properties; these properties should be used in determining the entities' similarities or dissimilarities that are associated with the surrounding context [17]. In practical applications, moreover, we note that the entities' semantic similarity is related to the contextual scenario and depends on the properties that they have expressed in common. For example, when we talk about *propulsion systems*, a *missile* is more similar to a *rocket* than to a *bomb*. (In general, missiles have their own propulsion systems, whereas bombs still use gravity). However, if we talk about a *damage model*, the opposite is true: the *missile* is more similar to a *bomb* compared with a *rocket*. However, in the current general word embedding method, the vectors are built for individual words, and each word is represented by a single vector in the semantic space and has a single meaning regardless of its practical application context. It has a significant limitation to differentiate entities' similarities in different context scenarios and would often produce inaccurate semantic similarity computing results for the subsequent tasks.

The main contributions of this work can be summarized as below. We propose a new semantic embedding method for entities' semantic similarities; in addition, we transform a point-to-point similarity measurement problem to a sequence-to-sequence problem, and then, we propose a novel semantic augmentation method by utilizing a double attention mechanism. For measuring the semantic similarity among entities, the knowledge graph embedding and transformer networks were introduced in the model. Different from the previous state-of-the-art similarity measurement methods, in this article, we propose a thematic similarity measure approach to analyze the connotation and denotation similarities among entities.

The remainder of this article is organized as follows. Section 2 presents related work and models, and we review the conventional semantic similarity measurement methods and semantic augmentation related methods. Section 3 gives the formal definition of the entity thematic similarity problem, and our proposed method is illustrated in Section 4. Section 5 describes the evaluation matrix, corpus construction, and experiments in detail. Finally, we draw conclusions and outline aspects of future work in Section 6.

## II. RELATED WORK AND MODELS

In this section, we review previous work that is relevant to aspects of our proposed method. This review mainly involves the conventional semantic similarity measuring approaches, the state-of-the-art results for popular evaluation datasets, knowledge graph embedding, transformer networks and Siamese neural networks for sequence similarity measurement.

### A. WORD SIMILARITY MEASUREMENT

Measuring the similarity between words is the core and roughly equivalent problem of entity similarity. In the literature, there are many metrics for measuring the similarity between words [18], [19]. The approaches of measuring

words' semantic similarities can be divided into the following groups: corpus-based approaches [20]–[24] and knowledge-based approaches [25]–[27].

A large number of the proposed approaches could be categorized as corpus based. These semantic similarity approaches are based on word associations learned from large text collections following the distributional hypothesis. Two entities are assumed to be more similar if their surrounding contexts are more similar or if they appear together more frequently. Therefore, by analyzing a large corpus, the valuable information can be extracted and used to measure the similarity between words. According to different computational models, there are count-based approaches and predictive-based approaches.

Count-based methods [28] use normal statistical analysis, and the cooccurrence statistics are directly applied with probabilistic models, matrix factorization and dimension reduction. Web-based word similarity uses web content as a corpus. In a general way, the similarity between words is calculated as the ratio between the number of web pages or snippets that contain both and number of pages or snippets that contain only one word of them.

There is a predictive-based approach that uses word embeddings to find word similarity [29]–[32]: Deep learning is used to represent words semantically. The generated word representation depends on the cooccurrence of words in the corpus. Usually, it directly learns dense vectors through predicting a word from its surrounding context. It has been reported to have good performance in many applications. Since the continuous bag of words (CBOW) model is more computationally efficient and suitable for a larger corpus than the skip-gram mode, the CBOW model is used to train word vectors in a neural network architecture that consists of an input layer, a projection layer, and an output layer, to predict a word given its surrounding words with a certain context window size. Having the trained word vectors, the word similarity is computed using standard cosine similarity. However, the training of word vectors uses only word sequences, and a wide variety of word relations is considered to be equally related according to their cooccurrences, which makes the similarity between trained word vectors coarse and unable to address synonymous words and hierarchical relations accurately. In consequence, knowledge-based semantic similarity methods are considered to enrich some commonsense knowledge of words.

Knowledge-based semantic similarity methods are used to measure the semantic similarity among entities based on semantic knowledge bases (e.g., Hownet [33] and Wordnet [34]), text knowledge bases (e.g., Wikipedia) and a triple-based knowledge bases (e.g., DBpedia [35] and YAGO [36]). Two words are considered to be more similar if they are located closer in the given knowledge bases. Many knowledge-based methods have been proposed in the literature. For example, measuring similarity in WordNet exploits various information, such as the shortest path length, depth, and information content of concepts. Two concepts are

assumed to be more similar if they are closer to each other in WordNet. Another common piece of information that is used to compute the semantic similarity is the depth, which is defined as the shortest path length between the root concept and a given concept through hierarchical relations. However, the structural knowledge of the taxonomy has a common drawback of having a uniform distance between concepts. Some methods considered can overcome this drawback by computing the similarity between the information content of the concepts. As information content-based methods lack important information on the path and depth, they are not able to represent the concepts' distance and specificity accurately. Apart from semantic similarity methods for specific ontologies such as WordNet, recent works have also started to propose semantic distances and similarity methods for linked open data or knowledge graphs [7], where a wide coverage of semantic relations between semantic resources is provided. These similarity methods are proposed for more general semantic networks and are focused on entity level resources.

## B. KNOWLEDGE GRAPH EMBEDDING

Knowledge graph embedding [37] is to project the components (such as entities, relations and property concepts) into a continuous low-dimensional, real-valued vector space, to simplify the manipulation while preserving the inherent structure of the KG. It can benefit a variety of downstream tasks, such as KG completion and relation extraction, and hence, it has quickly gained massive attention. It is highly beneficial for solving various natural language tasks that involve real world knowledge to access the embeddings of entities in a large knowledge base.

Recently, knowledge graphs (KGs) such as Wikidata, DBpedia, YAGO, and Microsoft Concept graphs, have been published as noteworthy large, cross-domain, and freely available resources; they have also been broadly used in improving the transparency of learning methods, as well as making it explainable for many artificial intelligence tasks. These graph structured knowledge bases contain a large number of entities and property information. Hence, KGs could be an ideal resource for obtaining a properties set of an entity and its representation correspondingly.

Most of the open knowledge bases, such as Wikipedia and YAGO, have items that are organized in a graphical structure. Graph analysis has been attracting increased attention in recent years due the ubiquity of networks in the real world. There have been several solutions that use graph matches to compute the similarity of two entities based on their neighborhood graphs. Recently, methods based on representing networks and each individual node in a vector space, while preserving their properties, have become widely popular. A graph  $G = (V, E)$  is a collection of  $V = \{v_1, v_2, \dots, v_n\}$  vertices and  $E$  edges. A graph embedding is a mapping  $f : v_i \rightarrow y_i \in R^d, \forall i \in [1, 2, \dots, n]$  such that  $d \ll |V|$  and the function  $f$  preserves some proximity measure defined on graph  $G$ . An embedding therefore maps each node to a

low-dimensional feature vector and attempts to preserve the connection strengths between vertices. Therefore, each entity is treated as a point in a vector space, and each relation is viewed as an operation over entity embeddings.

Each entity in the Wikipedia knowledge base is associated with an undirected graph whose nodes are entities and whose edges represent links among entities. Therefore, the entity embeddings could be learned by predicting neighboring entities in this link graph. Wikipedia2Vec [38], a Python-based open-source tool for learning the embeddings of entities from Wikipedia, learns embeddings by jointly optimizing word-based skip-grams, anchor context, and linkgraph models. It implements the conventional skip-gram model to map words and entities into a vector space. The skip-gram model [31] is a neural network model with a training objective to find embeddings that are useful for predicting context items given each item.

A typical knowledge graph such as YAGO usually depicts knowledge as multirelational data, and to demonstrate the relation between two entities, the triple facts (head entity, relation and tail entity) are used in general. An adjacency matrix is commonly used to represent the topology of a network, where each column and each row represent a node, and the matrix entries indicate the relationships among the nodes. To describe the local structural characteristics of a node, the neighborhood structure is important for network embedding. Although the adjacency vector of a node encodes the first-order neighborhood structure of a node, it is usually a sparse, discrete, and high-dimensional vector due to the nature of sparseness in large-scale networks. Such a representation is not friendly to subsequent applications. In the field of natural language processing, the word representation also suffers from similar drawbacks. The development of Word2Vec significantly improves the effectiveness of the word representation by transforming sparse, discrete and high-dimensional vectors into dense, continuous and low-dimensional vectors. To make analogy with Word2Vec, random walk models are exploited to generate random paths over a network. Some representative methods include DeepWalk [39] and Node2Vec [40]. Similar to DeepWalk, Node2Vec preserves higher-order proximity between nodes by maximizing the probability of occurrence of subsequent nodes in fixed length random walks. Moreover, some methods such as SDNE [41], SDAE [42], and SiNE [43], propose deep learning models for network embedding to make deep models fit to network data and to impose network structure and property-level constraints on deep models.

An entity is embedded into a low-dimensional continuous vector space while certain properties of the graph are preserved. The embedding vectors are usually obtained by minimizing a global loss function with regard to all entities and relations in such a way that each entity vector captures both global and local structural patterns of the original knowledge graph. Thus, we can utilize entity embeddings to encode prior knowledge for measuring the entities' similarities.

For obtaining the degree of attention for each property in a given thematic sentence, it is necessary to identify a semantic space and the corresponding representations. In this article, knowledge graph embedding methods were used to obtain the representation of each individual entity, property and other concept.

### C. TRANSFORMER NETWORKS

Recurrent neural networks (RNNs) [44], such as long short-term memory networks (LSTMs) [45], have been successfully employed for many tasks that require the modeling of sequential data; such tasks include language modeling [46], speech recognition [47], and machine translation [48]. In RNNs, the output predictions were made by computing a hidden state vector  $h_t$  based on the current input token and the previous states. However, because of requiring previous hidden states to be computed before the current time step, they cannot benefit from parallelization; this property underlies their ability to map arbitrary inputoutput sequence pairs. The transformer network [49] avoids the recurrence completely and uses only self-attention. The feed-forward network in each layer of the transformer network is a two-layered network with a *ReLU* activation. The sublayer is defined as follows:

$$FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2 \quad (1)$$

where  $W_1, W_2, b_1, b_2$  are trainable weights.

In this article, we propose a modified encoder part of a transformer network wherein the first component of each layer is a Siamese architecture and all parametric variables are shared.

### D. SIAMESE NEURAL NETWORKS

Siamese nets were first introduced in the early 1990s to solve signature verification as an image matching problem [50]. A Siamese neural network consists of twin networks, and the parameters between the twin networks are tied. An energy function that computes some metric between the highest-level feature representation on each side joined the networks at the end. For entity similarity measuring, these Siamese architectures guarantee that two extremely similar property sequences could not possibly be mapped by their respective networks to very different locations in semantic space.

The skip-thoughts model [51], which extends the skip-gram approach of Word2Vec from the word to sentence level, feeds each sentence into an RNN encoder-decoder. To measure the similarities of sentences, the RNN encoder was used to obtain a skip-thought vector. Subsequently, a separate classifier is trained by using skip-thought vectors for the pair of sentences that appear in each training example.

As a benefit from the order-sensitive chain-structure, standard LSTMs have become the state-of-the-art models for a variety of machine learning problems, such as text classification and language translation. The Manhattan LSTM model [52] is a Siamese adaptation of the LSTM network, which

is proposed for labeled data comprised of pairs of variable-length sequences. It could be applied to assess the semantic similarity between sentences. This model produced a mapping from a general space of variable length sequences into an interpretable structured metric space of fixed dimensionality. By restricting the subsequent operations to rely on a simple Manhattan metric, a highly structured space whose geometry reflects complex semantic relationships was formed for representing the sentences.

BERT [53] and RoBERTa [54] have set a new state-of-the-art performance on sequence-pair regression tasks. However, the construction of BERT makes it unsuitable for semantic similarity search; it requires that both sentences are fed into the network, which causes a massive computational overhead. Sentence-BERT (SBERT) [55] is a modification of the pre-trained BERT network that uses Siamese and triplet network structures to derive semantically meaningful sentence embeddings that can be compared using cosine-similarity. This approach reduces the effort for the massive computational problem.

### III. PROBLEM FORMULATION

Existing similarity computing methods cannot be applied directly in this case because, in reality, the similarity is a dynamic phenomenon: it varies with the context scenario.

To address this limitation, a measurement for reasoning about relative similarities is presented. The relative similarity measurement of entities is defined as accessing the comparison of similarities between two pairs of entities' connotation and denotation information. Therefore, let  $C$  be a function of relative similarity measuring; it has two parts of subfunctions that represent the connotation and denotation similarity of entities, respectively. Let  $Sim\_C$  be the function of connotation similarity of entities, and let  $Sim\_D$  be the function of denotation similarity of entities; then, the semantic similarity between entity  $E_A$  and entity  $E_B$  could be defined as follows:

$$C(E_A, E_B) = \alpha \cdot Sim\_C(E_A, E_B) + (1 - \alpha) \cdot Sim\_D(E_A, E_B) \quad (2)$$

Here,  $\alpha \in [0, 1]$  is a compound factor. Usually, we consider that the set of properties always implies the connotation information of the entity, while the denotation information of the entity is contained in the context scenarios. In the field of natural language processing, the context scenarios of an entity are always implicit in the sentence associated with the entity.

In this article, we proposed a novel semantic augmentation method with a double attention mechanism. The method refers to a dynamic representation learning process that maps an entity to a real number vector in semantic space based on its properties and the context scenario.

For measuring the connotation similarity  $Sim\_C$ , we construct a property attention matrix, according to the thematic sentence and the properties of the entity. Thus, let  $P_r$  be a properties set of an entity; then, the connotation similarity

between entity  $E_A$  and entity  $E_B$  could be defined as follows:

$$Sim\_C(E_A, E_B) \cong Sim(P_r^{(A)}, P_r^{(B)}) \quad (3)$$

On the other hand, we assumed that similar entities have similar context scenarios. That means that there is a degree of interchangeability between two similar entities associated with the context scenarios. Thus, we defined the denotation similarity between entity  $E_A$  and entity  $E_B$  as the following:

$$Sim\_D(E_A, E_B) \cong Sim(S_A^{(A)}, S_A^{(B)}, S_B^{(A)}, S_B^{(B)}) \quad (4)$$

Here,  $S_x^{(y)}$  means a batch of sentences in which item  $x$  has been replaced by  $y$ .

Through the above analysis, we can transform the point-to-point similarity measurement problem to the sequence-to-sequence problems. State-of-the-art results on sequence embedding often use attentional models with some form of convolution or recursion [56], [57]. Instead, the research in [49] introduces the transformer network, which uses only self-attention and feed-forward layers to avoid the recurrence equation and maps the input sequences into hidden states. Specifically, the authors use positional encodings in conjunction with a multihead attention mechanism. This approach allows for increased parallel computation and reduces the time to convergence and achieves state-of-the-art results on several natural language processing tasks [58]–[61].

Drawing on the research approach of transformer networks, for comparing the similarity among entities, we employ an attention mechanism in this work. Our proposed semantic augmentation model represents the properties of sequences and context scenarios of a sentence of an entity by using self-attention and Siamese self-attention networks, respectively. Consequently, a semantically structured representation space can be learned.

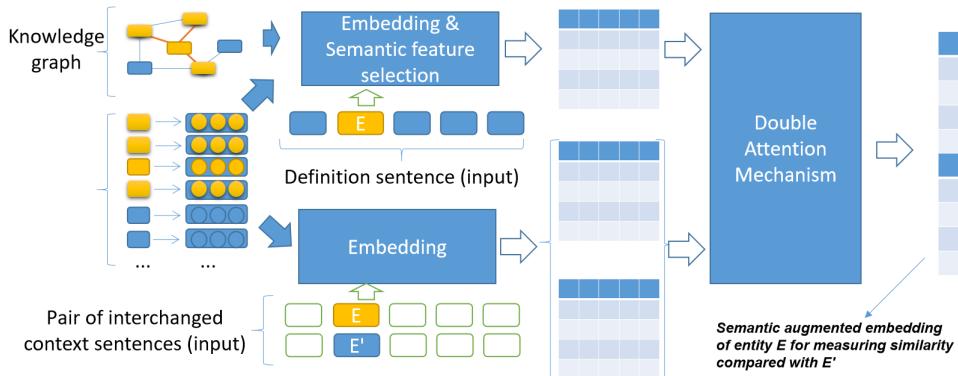
### IV. SEMANTIC AUGMENTATION

In this section, we present a semantic augmentation method for measuring the similarity of entity names. The processes consist of semantic feature selection and the double attention mechanism embedding refining phase.

#### A. SEMANTIC FEATURE SELECTION

Through the embedding process, we obtain a concept/property embedding index, denoted as  $I_p$ , which would be used as a basic look-up table for converting a property to its vector representation in subsequent processes. To identify and rank the properties of an entity according to the degree of attention in a thematic sentence, we present a semantic feature selection method that consists of properties' attention calculating phase, selecting phase and sorting phase.

The goal of feature selection is to obtain a sequence of the ideal attention ranking of the properties. Let  $f$  be a connotation attention function and let  $P_r$  be a property set of an entity, as mentioned above. Let  $pr_i^f$  be the degree of attention for a property  $p_{ri}$  ( $p_{ri} \in P_r$ ); then,  $pr_1^f, pr_2^f, \dots, pr_n^f$  can be



**FIGURE 2.** The overall working process of semantic augmentation for measuring entity similarity.

the degree of relevancy associated with the thematic sentence and ranked by the attention function  $f$ .

Let  $W = \{w_1, w_2, \dots, w_{i\_entity}, \dots, w_m\}$  be a sentence that implies the context scenario of the entity  $w_{i\_entity}$ , and then, the entity  $w_{i\_entity}$  is represented by its properties set. In this article, we use the entity's adjacency neighbors in the knowledge graph instead, denoted as  $pr = \{pr_1, pr_2, \dots, pr_j, \dots, pr_n\}$ , and obviously,  $pr_i \in I_p$ .

Here,  $\Phi$  is an indicator function, and  $\Phi(w_i) = 1$ , if the word-embedding pair  $\langle w_i, w_{i\_embedding} \rangle$  is contained in dictionary  $I_p$ ; otherwise,  $\Phi(w_i) = 0$ .

By applying max-pooling to each row of the attention matrix A, we obtained the max attention value of the property, and then, the property sequence  $pr_1^f \rightarrow pr_2^f \rightarrow \dots \rightarrow pr_i^f \rightarrow \dots \rightarrow pr_n^f$  was identified according to the descending order from the values of the attention. This property sequence could express the semantic information of an entity combined with the context scenario.

### B. ATTENTION MECHANISM

The connotation and denotation information could be utilized to enhance the degree of similarity among entities. The self-attention and a Siamese self-attention mechanism were used to compile that information in our work.

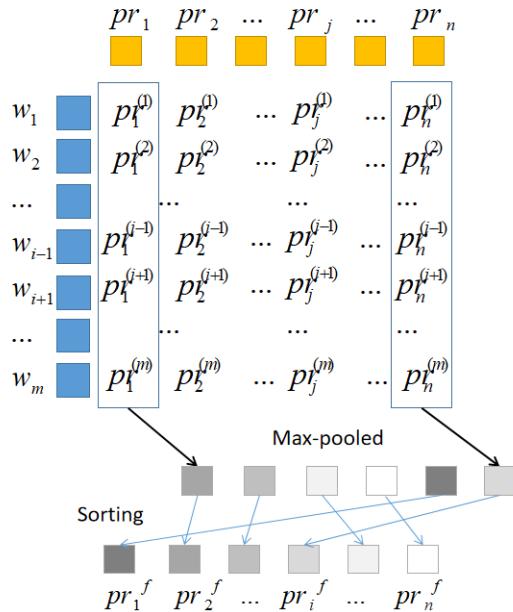
For encoding the semantic feature sequence which was selected from the knowledge graph, as mentioned in the transformer networks, our model follows the architecture that uses stacked self-attention and pointwise, fully connected layers for the encoder shown in the left half of Figure 4.

To encode the interchanged context scenarios with comparable entities, the proposed Siamese transformer networks model is outlined in the right half of Figure 4.

From the source tokens of each of the sentences pairs, we add “positional encodings” to the pre-trained input embeddings at the bottoms of the encoder. Since the positional encodings have the same dimension as the embeddings, by using an additive operation, the embeddings of dimension  $d_{model}$  are generated. The encoder consists of N layers, and each layer contains two sublayers. The proposed attention layer can be described as follows:

$$\begin{aligned} & SiamMultiHead(Q^{(a)}, K^{(a)}, V^{(a)}, Q^{(b)}, K^{(b)}, V^{(b)}) \\ &= \text{Concat}(\text{MultiHead}(Q^{(a)}, K^{(a)}, V^{(a)}), \\ & \quad \text{MultiHead}(Q^{(b)}, K^{(b)}, V^{(b)})) \end{aligned} \quad (6)$$

$$\begin{aligned} & \text{MultiHead}(Q^{(.)}, K^{(.)}, V^{(.)}) \\ &= \text{Concat}_i(\text{head}_i^{(.)})W^o \end{aligned} \quad (7)$$



**FIGURE 3.** The working process of semantic feature selection. Here,  $w_i$  is the target entity. Thus,  $\{W - w_i\}$  was used in calculating the attention matrix.

The attention calculation between  $\{W - w_{i\_entity}\}$  and  $pr$  produce an attention matrix  $A = \{pr_j^{(i)}\}$ ,  $1 \leq i \leq m$  and  $i \neq i\_entity$ ,  $m = |W|$ ,  $1 \leq j \leq n$ ,  $n = |pr|$  as follows:

$$pr_j^{(i)} = \text{cosine}(w_{i\_embedding}, pr_j\_embedding).\phi(w_i) \quad (5)$$

$$\text{head}_i^{(.)} = \text{Attention}(Q^{(.)}W_i^Q, K^{(.)}W_i^K, V^{(.)}W_i^V) \quad (8)$$

Different from the original transformer network, there are two parallel networks of multihead attention, which each process one of the sentences in a given pair ( $S_A^{(\sim)}, S_B^{(\sim)}$ ), but we solely focus on Siamese architectures with tied weights in this work.

We employ  $h = 4$  for each of the parallel multihead attention. For each of these, we use  $d_k = d_v = d_{model}/h = 50$ .

Due to the reduced dimension of each head, the total computational cost is similar to that of single-head attention with full dimensionality. More concretely, each pair of sentences (each represented as a sequence of word vectors) is passed to the Siamese multihead attention, which updates its hidden state at each sequence. The final representation of the sentences pair is encoded by a feed-forward network, and it uses a two-layered network with a *ReLU* activation.

The max and mean pooling methods are operated on the output of fully connected layers for the encoders shown in the left and right halves of Figure 4. Then, we concatenated the last hidden state of the models, and the refined embedding of the entity was normalized via a softmax layer. Since the dimension of the embedding vector is dependent on the outputs of dimension  $d_{model}$ , for similarity measuring, the entities have embeddings with the same dimension.

## V. EXPERIMENTS

This section presents the datasets, implementation, and evaluation and provides a brief discussion about the obtained experimental results. To the best of our knowledge, there is currently no standard method and data sets to evaluate the performance of the entity semantic similarity method. Therefore, we construct an entity similarity dataset that is based on popular datasets for word similarity evaluation and datasets for measuring entity relatedness.

### A. DATASETS AND IMPLEMENTATION

Attributional similarity is the degree to which two words are synonymous. We collected several publicly available standard datasets for evaluating semantic similarity models, and these data sets are conventionally the most commonly used for word similarity evaluation. Some typical evaluation data sets are described as follows.

The WordSimilarity-353 (WS353) Test Collection is perhaps the most commonly used gold labeled data set for semantic similarity, at present. This data set contains 353 pairs of English words, which is merely a concatenation of two smaller sets. The first set contains 153-word pairs along with their similarity scores assigned by 13 subjects. The second set contains 200-word pairs, with their similarity assessed by 16 subjects. Each set provides the raw scores assigned by each subject, as well as the mean score for each word pair.

The Rubenstein & Goodenough (RG-65) dataset is the first and most used data set that contains human assessments

of word similarities. The dataset resulted from experiments conducted in 1965 in which a group of 51 subjects (all native English speakers) assessed the similarity of 65 pairs of words selected from ordinary English nouns. The similarity of each pair is scored according to a scale from 0 to 4 (the higher the value is, the higher the similarity).

SimLex-999 [62] is a recently released dataset that consists of 999 word pairs for evaluating semantic similarity specifically. Pairs of words were chosen to represent different ranges of similarity, with either a high or low association. Each pair of words was rated by more than 35 subjects (native English speakers) with similarity scores on a scale from 0 to 10, and the average was assigned as the final judgment.

**TABLE 1. State-of-the-art results for typical datasets.**

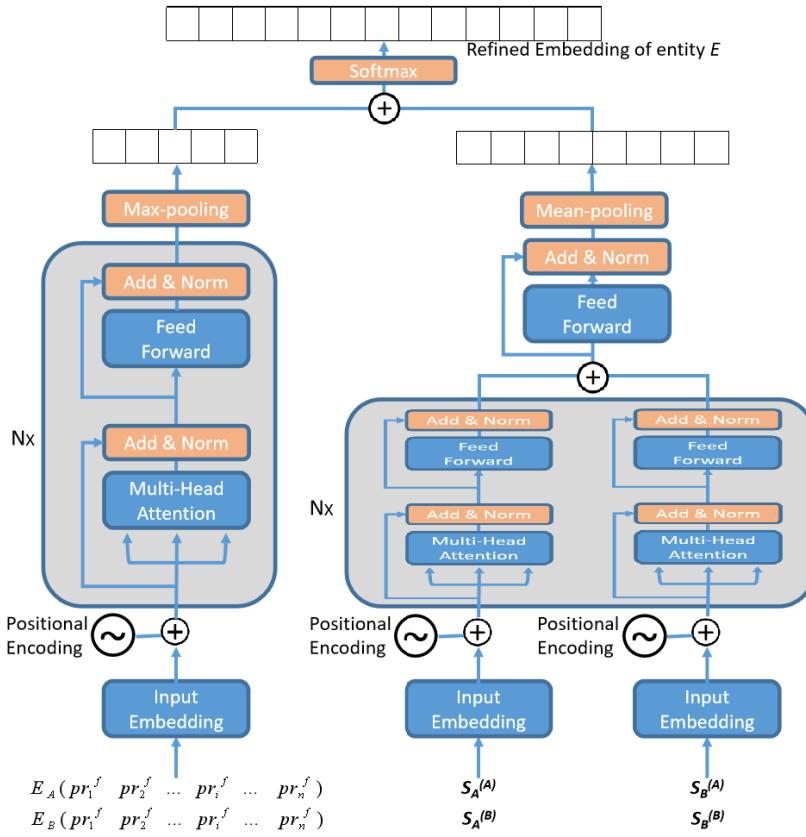
Data Sets	Type of methods	Spearman correlation ( $\rho$ )	Pearson correlation (r)
WS353	Hybrid[63]	0.828	-
	Corpus-based [22]	0.810	-
	Knowledge-based[23]	0.622	0.629
RG-65	Knowledge-based[27]	0.920	0.910
	Hybrid[25]	0.901	0.896
	Corpus-based[23]	0.833	0.861
SimLex-999	Hybrid [64]	0.760	-
	Combined[65]	0.642	0.658
	distributional [66]	0.560	-
MEN	Hybrid [25]	0.862	0.865
	Corpus-based, distributional [25]	0.809	0.803

The MEN Test Collection contains 3,000-word pairs, which were randomly selected from words that occur at least 700 times in the freely available ukWaC and Wikipedia corpora combined and at least 50 times in the open-sourced subset of the ESP game dataset. Two sets of English word pairs (one for training and one for testing) together with human-assigned similarity judgments were obtained by crowdsourcing using Amazon Mechanical Turk via the CrowdFlower interface.

All of the datasets described above contain a list of word pairs along with human-assigned similarity judgments. The semantic ratings of those word pairs have been proven to be highly correlated and could be reliably used for evaluating semantic similarity methods. The state-of-the-art results for each dataset are listed in table 1. The semantic similarity metrics presented in this article are used for entities, which are associated with the text around them. For training the Siamese network, more datasets for evaluating the methods of word similarity and entity relatedness were employed.

We used the January 2020 version of the English Wikipedia dump and an open-source tool WikiExtractor<sup>1</sup> to create the training corpus. For each word/entity in the word pairs, was removed duplicate data and overlapping definitions in Wikipedia. A sentence that contains the word/entity itself was selected as the context scenario sentence. There are a total

<sup>1</sup><https://github.com/attardi/wikiextractor>



**FIGURE 4.** The entity semantic augmentation architecture for measuring the similarity among entities.

**TABLE 2.** The count of word pairs with the removal of duplicate data and overlapping definitions in the wikipedia knowledge base.

Datasets	Original word pairs count	Reserved word pairs count
MTURK771	711	705
Agirre201	201	201
WP300	300	244
MEN	3000	2973
M&C28	28	5
PiuroSoco	65	34
RareWords2034	2034	1336
ReI22	122	121
SCWS1994	1994	1773
SimLex999	999	947
WordSim353	353	28
YP130	130	117
Total	9937	8484

of 28068 training data instances, and the format of the data is explained in Figure 5.

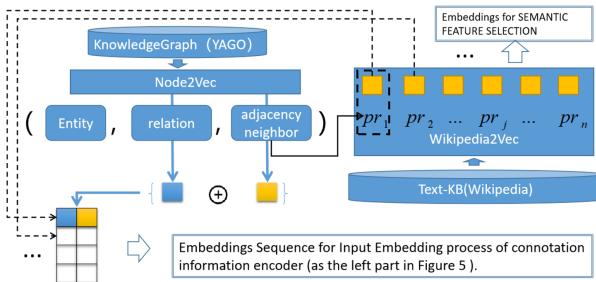
To evaluate the similarity among the entities in a certain context, the testing data set was constructed based on 300 unique pairs of entities, such as person, cities, organizations, and so on. In total, 60 pairs were allocated with different similarity values according to different context attention of their entities. Thus, we consider that these are different testing data. Finally, the testing data set was composed of 360 entity

```
<IDATA>
<PAIR_Source>WP300
<Score>0.9
<Normalized>0.9
<E_A>microsoft
    <DEF>{Microsoft} Corporation (NASDAQ:MSFT) is a company that makes computer software and video games for users around the world.
    <CONTX_1>{Microsoft} makes Windows, {Microsoft} Office (including {Microsoft} Word), Edge, MSN and Xbox, among others.
    <CONTX_2>Most {Microsoft} programs cannot be downloaded for free.
    <CONTX_3>{Microsoft} Word Viewer is a freeware program for {Microsoft} Windows.
    <CONTX_4>{Microsoft} Windows is an operating system for computers made by the United States-based company {Microsoft}.
<E_B>apple
    <DEF>{Apple} Inc. is a multinational company that makes computer hardware (the Macintoshes), software (macOS, iOS, watchOS and tvOS), and mobile devices (iPod, iPhone and iPad) like music players.
    <CONTX_1>{Apple} calls its computers Macintoshes or Macs, and it calls its laptops MacBooks.
    <CONTX_2>{Apple} made a hardware keyboard that plugs into its side.
```

**FIGURE 5.** The example of the data in the training corpus.

pairs, and the similarity of each pair is scored according to a scale from 0 to 1 (the higher the “similarity of meaning” is, the higher the number). The similarity values in the dataset are the means of judgments made by 6 subjects.

To obtain the embeddings for the semantic feature selection, we used an open-source Python library,

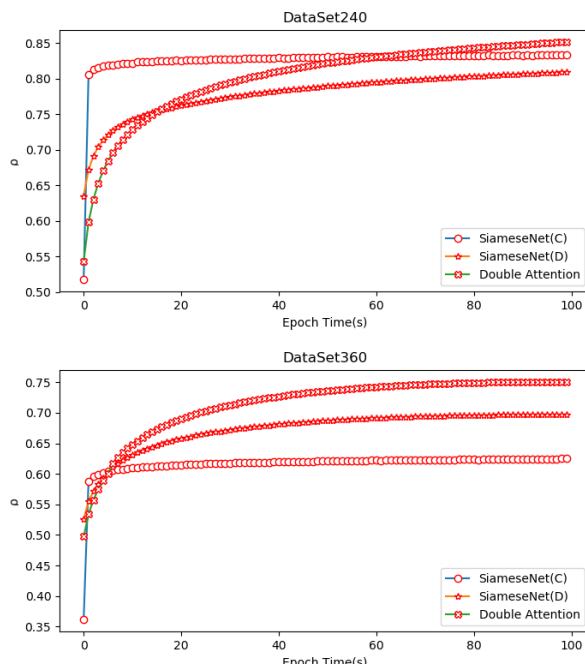


**FIGURE 6.** The input embedding approaches for the semantic feature selection and connotation information encoder.

Wikipedia2Vec,<sup>2</sup> which provides a unified interface to the implementations of entity embedding methods. We used a source Wikipedia dump file enwiki-20200101-pages-articles.xml.bz2, 15.6 GB from Wikimedia Downloads (<https://dumps.wikimedia.org/>). Then, the embeddings were trained from a Wikipedia dump using the train command of wikipedia2vec with the following options:

**TABLE 3.** The train-embedding command options.

Train-embedding	Options	Value
enwiki_20200101_win10	dim-size	200
	window	10
	iteration	10
	negative	15
	others	default



**FIGURE 7.** The Spearman rank correlation  $\rho$  between the cosine similarity of the entity representations and the gold labels for entity similarity (on dataset240) and entity thematic similarity (on dataset360) tasks.

<sup>2</sup><https://wikipedia2vec.github.io>

To obtain the embeddings sequence for the input embedding process of the connotation information encoder, we employed a large semantic knowledge base YAGO, which was derived from Wikipedia, WordNet, WikiData, GeoNames, and other data sources. Currently, YAGO knows more than 17 million entities (such as persons, organizations, and cities), and it contains more than 150 million facts about these entities. The reason is that choosing the right balance enables node2vec to preserve the community structure as well as the structural equivalence between the nodes. We employed node2vec as the method for knowledge graph embedding. We utilized an open-source Python library, GEM<sup>3</sup> (graph embedding methods), which provides a unified interface to the implementations of graph embedding methods. The library provides implementations of node2vec [41]. Through the knowledge graph embedding process, we concatenated the embeddings of the relation and tail entity associated in triple facts with the entity. These steps constructed a basic dictionary for converting a property to its vector representation for subsequent processes. For input embedding processing of the denotation information encoder, we used the pretrained BERT model with the huggingface PyTorch library<sup>4</sup> to fine-tune our model.

Finally, for a pair of entities  $E_A$  and  $E_B$ , their similarity score is defined by the cosine value of the corresponding refined embedding vectors

We used the training set to fine-tune our models using the regression objective function. At the time of prediction, we computed the cosine-similarity between the entity embeddings. All of the models were trained with 10 random seeds to counter the variances.

## B. EVALUATION MATRIX AND RESULTS

Spearman's rank correlation coefficient is a nonparametric rank statistic that was proposed by Charles Spearman as a measure of the strength of an association between two variables. In this article, we utilize Spearman's rank correlation coefficient to evaluate the strength of an association between similarities accessed by comparable models and gold labels.

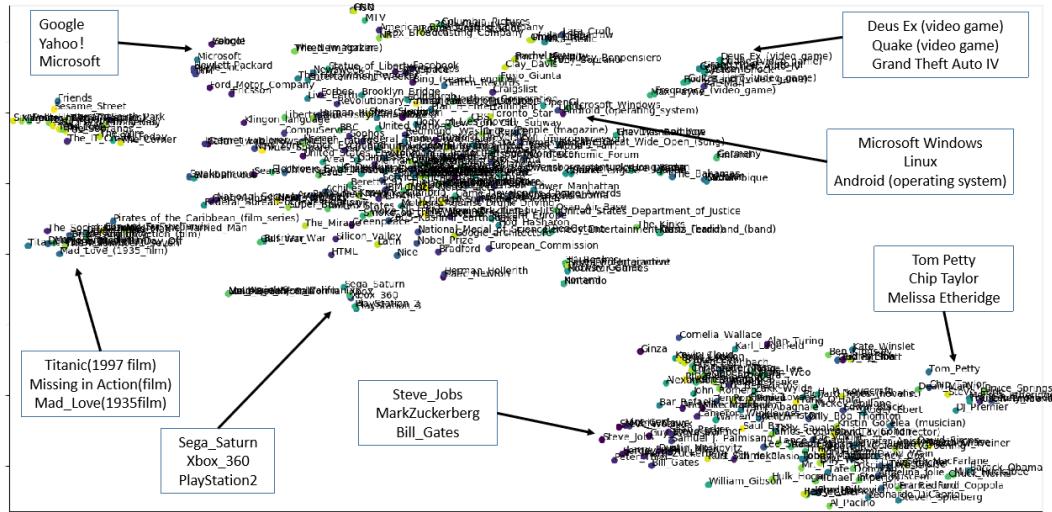
To evaluate the model's context adaptation, we use the testing data set DataSet360 and its subset DataSet240 (after removing test data that has two similarity scores for the full testing dataset). In order to be able to present a reliable comparison, we have done tests with multiple state-of-the-art models in such a way that the same input data was used those models.

The results are depicted in table 4. We experimented with three semantic augmentation models: Only using the connotation information (SiameseNet(C)) or denotation information (SiameseNet(D)) and the combined model (Double Attention) to measure the similarity among entities.

The results show that directly using the output of BERT leads to rather poor performances, which achieve an aver-

<sup>3</sup><https://github.com/palash1992/GEM>

<sup>4</sup><https://github.com/huggingface/transformers>



**FIGURE 8.** This schematic shows that our model could effectively make a separation among the entities from different domains. In addition, makes similar entities closer.

**TABLE 4.** The Test results.

Type of methods	Models	Spearman correlation ()	
		DataSet240	DataSet360
Knowledge-based	NASARI[27]	0.860	0.632
	J&C[23]	0.612	0.510
Corpus-based	HVSM[22]	0.770	0.557
	D&C[25]	0.769	0.571
Hybrid	ESA[23]	0.793	0.610
	BERT	0.633	0.581
OUR	WE4L[62]	0.710	0.585
	D&C[25]	<b>0.861</b>	0.670
OUR	SiameseNet(C)	0.834	0.625
	SiameseNet(D)	0.809	0.697
OUR	Double Attention	0.852	<b>0.751</b>

age correlation of only 0.633 and 0.581 on Dataset240 and Dataset360, respectively, for the best of three semantic augmentation models. All are worse than the existing state-of-the-art methods.

We show that using the described Siamese network structure and fine-tuning mechanism substantially improves the correlation, and the performance of the semantic augmentation models is comparable with existing state-of-the-art methods in the traditional entity similarity measurement task. Moreover, we observe that the combined strategy leads to a significant improvement in the entity thematic similarity measuring task.

### C. CASE STUDY

For the case study, we show that our method's performance on the Related Entities Dataset KORE, which contained 20 seed entities from 4 domains (IT companies, Hollywood celebrities, video games, television series). For each of these entities, 20 entities linked from their Wikipedia article were selected, and they were ranked by human annotators on Mechanical Turk

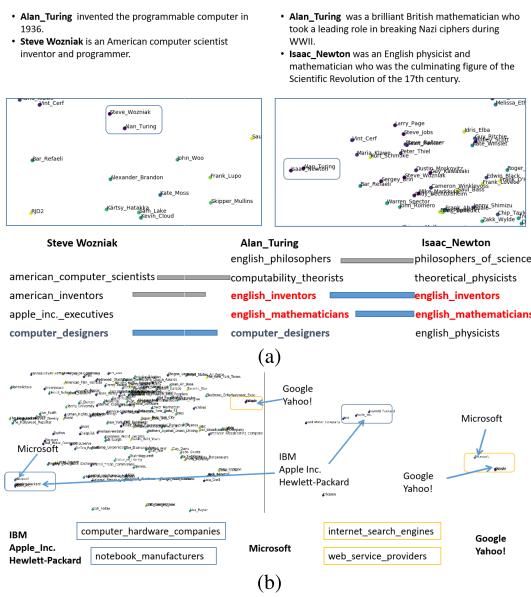
**TABLE 5.** The seed entities and the linked entities form their Wikipedia article, which were ranked by human annotators (examples).

Seed Entity	Ranked Relative Entities
Apple Inc.	Steve_Jobs; Steve_Wozniak; Mac_Pro; Silicon_Valley; Safari_(web_browser); FileMaker,_Inc.; Microsoft; Ford_Motor_Company
Google	Larry_Page; Sergey_Brin; Google_Maps; Android_(operating_system); Bing_(search_engine); Stanford_University; Yahoo!; Apple_Inc.
Brad Pitt	Angelina_Jolie; Jennifer_Aniston; Fight_Club_(film); Rusty_Ryan; Seven_(film); Tom_Cruise; People_(magazine); CNN; Pakistan
Leonardo DiCaprio	Inception(film); Titanic(1997 film); Frank_Bagnale; Golden_Globe_Award
Guitar Hero (video game)	for_Best_Actor; Los_Angeles; Steven_Spielberg; Body_of_Lies_(novel); Harmonix_Music_Systems; Red_Octane; Marcus_Henderson; Gibson_SG; Rock_Band_(video_game); Smoke_on_the_Water; Japan; Wired_(magazine); Omar_Little; Jimmy_McNulty; David_Simon; Robert_F._Colesberry; HBO; Franks_Wild_Years; Six_Feet_Under_(TV_series); Chicago_Sun-Times; Soul_Food_(TV_series); Salon.com
The Wire	

Figure 8 shows that our model could effectively make a separation among the entities from different domains. In addition, it makes similar entities closer.

Our model shows good performance on the task of entity thematic similarity, as shown in Figure 9 (a). Taking the person name entities of Alan Turing, Isaac Newton and Steve Wozniak as an example, if the context is to talk about computer science, then Alan Turing is closer to Steve Wozniak than to Isaac Newton. Our model could also give an explainable cause in that both Alan Turing and Steve Wozniak are computer designers. If the context is to talk about a professional's nationality, then Alan Turing is closer to Isaac Newton than to Steve Wozniak. An explainable cause is

that both Alan Turing and Isaac Newton are English inventors and mathematicians. In Figure 9 (b), taking company name entities Microsoft, IBM, Apple Inc., Hewlett-Packard, Google and Yahoo! as an example, if the context is to talk about hardware manufacture, then Microsoft is closer to IBM, Apple Inc., Hewlett-Packard, and an explainable cause is that those companies are all computer hardware companies and notebook manufacturers. If the context is to talk about web services, then Microsoft is closer to Google and Yahoo! because all of them are web service providers and have products that are internet search engines.



**FIGURE 9.** The entity thematic similarity examples in person name entities (a) and company name entities (b).

## VI. CONCLUSION

Estimating the similarity between entity names plays an important role in personalized searching services in the edge environment. By considering that the entities' semantic similarity is related to the context scenario and depends on properties they have expressed in common, we propose a novel semantic augmentation method with a double attention mechanism. The method refers to a dynamic representation learning process that maps an entity to a real number vector in semantic space. Different from previous state-of-the-art similarity measurement methods, in this article, we propose a thematic similarity measure approach to analyzing the connotation and denotation similarities among entities. By utilizing a double attention mechanism, we transform a point-to-point similarity measurement problem to a sequence-to-sequence problem, and the knowledge graph embedding and transformer networks were introduced in the model.

The experimental results show that directly using the output of BERT leads to rather poor performances, but the combined strategy leads to a significant improvement in the entity thematic similarity measurement task. The model could effectively make a separation among the entities from

different domains. In addition, it makes similar entities closer. It also shows good performance on the task of entity thematic similarity.

## REFERENCES

- [1] L. Zhao, G. Han, Z. Li, and L. Shu, "Intelligent digital twin-based software-defined vehicular networks," *IEEE Netw.*, early access, Jun. 5, 2020, doi: [10.1109/MNET.2011.1900587](https://doi.org/10.1109/MNET.2011.1900587).
- [2] N. Lin, L. Fu, L. Zhao, G. Min, A. Al-Dubai, and H. Gacanin, "A novel multimodal collaborative drone-assisted VANET networking model," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4919–4933, Jul. 2020.
- [3] N. Hassan, K.-L.-A. Yau, and C. Wu, "Edge computing in 5G: A review," *IEEE Access*, vol. 7, pp. 127276–127289, 2019, doi: [10.1109/ACCESS.2019.2938534](https://doi.org/10.1109/ACCESS.2019.2938534).
- [4] M. A. Rodriguez and M. J. Egenhofer, "Determining semantic similarity among entity classes from different ontologies," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 2, pp. 442–456, Mar. 2003.
- [5] Q. Zhao, C. Wang, P. Wang, M. Zhou, and C. Jiang, "A novel method on information recommendation via hybrid similarity," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 3, pp. 448–459, Mar. 2018.
- [6] S. Sang, Z. Yang, X. Liu, L. Wang, H. Lin, J. Wang, and M. Dumontier, "GrEDE: A knowledge graph embedding based method for drug discovery from biomedical literatures," *IEEE Access*, vol. 7, pp. 8404–8415, 2019.
- [7] G. Zhu and C. A. Iglesias, "Computing semantic similarity of concepts in knowledge graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 1, pp. 72–85, Jan. 2017, doi: [10.1109/TKDE.2016.2610428](https://doi.org/10.1109/TKDE.2016.2610428).
- [8] L. Li, X. Shang, J. Li, and J. Hu, "Learning distance metrics for entity resolution," *IEEE Access*, vol. 6, pp. 54900–54909, 2018, doi: [10.1109/ACCESS.2018.2871168](https://doi.org/10.1109/ACCESS.2018.2871168).
- [9] X. Yin, Y. Huang, B. Zhou, A. Li, L. Lan, and Y. Jia, "Deep entity linking via eliminating semantic ambiguity with BERT," *IEEE Access*, vol. 7, pp. 169434–169445, 2019, doi: [10.1109/ACCESS.2019.2955498](https://doi.org/10.1109/ACCESS.2019.2955498).
- [10] Y. Li, Z. A. Bandar, and D. McLean, "An approach for measuring semantic similarity between words using multiple information sources," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 4, pp. 871–882, Jul. 2003, doi: [10.1109/TKDE.2003.1209005](https://doi.org/10.1109/TKDE.2003.1209005).
- [11] K. Heylen, Y. Peirsman, D. Geeraerts, and D. Speelman, "Modelling word similarity: An evaluation of automatic synonymy extraction algorithms," in *Proc. Int. Conf. Lang. Resour. Eval. LREC*, Marrakech, Morocco: DBLP, May 2008, pp. 3243–3249.
- [12] A. Panchenko and O. Morozova, "A study of hybrid similarity measures for semantic relation extraction," in *Proc. ACL Workshop Innov. Hybrid Approaches Process. Textual Data*, 2012, pp. 10–18.
- [13] E. Chen, S. Qiu, C. Xu, F. Tian, and T. Liu, "Word embedding: Continuous space representation for natural language," *J. Data Acquisition Process.*, vol. 29, no. 1, pp. 19–29, 2014.
- [14] M. Toshevská, F. Stojanovska, and J. Kalajdžieski, "Comparative analysis of word embeddings for capturing word similarities," 2020, *arXiv:2005.03812*. [Online]. Available: <http://arxiv.org/abs/2005.03812>
- [15] Y. Shi, Y. Zheng, K. Guo, L. Zhu, and Y. Qu, "Intrinsic or extrinsic evaluation: An overview of word embedding evaluation," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2018, pp. 1255–1262.
- [16] A. Alwehaibi and K. Roy, "Comparison of pre-trained word vectors for arabic text classification using deep learning approach," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 1471–1474, doi: [10.1109/ICMLA.2018.00239](https://doi.org/10.1109/ICMLA.2018.00239).
- [17] J. Liu and L. Birnbaum, "Measuring semantic similarity between named entities by searching the Web directory," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Nov. 2007, pp. 461–465.
- [18] A. M. Jacobs and A. Kinder, "Features of word similarity," 2018, *arXiv:1808.07999*. [Online]. Available: <http://arxiv.org/abs/1808.07999>
- [19] L. Jin and W. Schuler, "A comparison of word similarity performance using explanatory and non-explanatory texts," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2015, pp. 990–994.
- [20] A. Islam and D. Inkpen, "Semantic text similarity using corpus-based word similarity and string similarity," *ACM Trans. Knowl. Discovery From Data*, vol. 2, no. 2, pp. 1–25, Jul. 2008, doi: [10.1145/1376815.1376819](https://doi.org/10.1145/1376815.1376819).
- [21] Y. E. Esin, O. Alan, and F. N. Alpaslan, "Improvement on corpus-based word similarity using vector space models," in *Proc. 24th Int. Symp. Comput. Inf. Sci.*, Sep. 2009, pp. 280–285.

- [22] W. Yih and V. Qazvinian, "Measuring word relatedness using heterogeneous vector space models," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL-HLT)*, 2012, pp. 616–620, doi: [10.1007/s004930070014](https://doi.org/10.1007/s004930070014).
- [23] H. Samer and R. Mihalcea, "Semantic relatedness using salient semantic analysis," in *Proc. AAAI*, 2011, pp. 1–6.
- [24] A. Pawar and V. Mago, "Calculating the similarity between words and sentences using a lexical database and corpus statistics," *Comput. Lang.*, 2018. [Online]. Available: <https://arxiv.org/abs/1802.05667>
- [25] A. Dobó and J. Csirik, "A comprehensive study of the parameters in the creation and comparison of feature vectors in distributional semantic models," *J. Quant. Linguistics*, vol. 27, no. 3, pp. 244–271, 2019, doi: [10.1080/09296174.2019.1570897](https://doi.org/10.1080/09296174.2019.1570897).
- [26] J. J. Lastra-Díaz, J. Goikoetxea, M. A. H. Taieb, A. García-Serrano, M. B. Aouicha, and E. Agirre, "A reproducible survey on word embeddings and ontology-based methods for word similarity: Linear combinations outperform the state of the art," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 645–665, Oct. 2019, doi: [10.1016/j.engappai.2019.07.010](https://doi.org/10.1016/j.engappai.2019.07.010).
- [27] J. Camacho-Collados, M. T. Pilehvar, and R. Navigli, "NASARI: A novel approach to a semantically-aware representation of items," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2015, pp. 567–577, doi: [10.3115/v1/N15-1059](https://doi.org/10.3115/v1/N15-1059).
- [28] L. Rémi and R. Collobert, "Rehabilitation of count-based models for word vector representations," in *Proc. Int. Conf. Intell. Text Process. Comput. Linguistics*. Cham, Switzerland: Springer, 2015, pp. 417–429, doi: [10.1007/978-3-319-18111-0\\_31](https://doi.org/10.1007/978-3-319-18111-0_31).
- [29] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*. [Online]. Available: <https://arxiv.org/abs/1301.3781>
- [30] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 3111–3119.
- [31] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 3, 2014, pp. 2177–2185.
- [32] Y. Goldberg and O. Levy, "word2vec Explained: Deriving Mikolov et al.'s negative-sampling word-embedding method," 2014, *arXiv:1402.3722*. [Online]. Available: <https://arxiv.org/abs/1402.3722>
- [33] Z. Dong and Q. Dong, "HowNet—A hybrid language and knowledge resource," in *Proc. Int. Conf. Natural Lang. Process. Knowl. Eng.*, Oct. 2003, pp. 820–824.
- [34] G. A. Miller, "WordNet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995, doi: [10.1145/219717.219748](https://doi.org/10.1145/219717.219748).
- [35] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A nucleus for a Web of open data," in *Proc. Semantic Web, Int. Semantic Web Conf., Asian Semantic Web Conf., Iswc + Aswc*, Busan, KoreaSouth Korea Nov. 2007, pp. 722–735.
- [36] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: A core of semantic knowledge," in *Proc. 16th Int. Conf. World Wide Web WWW*, 2007, pp. 820–824, doi: [10.1145/1242572.1242667](https://doi.org/10.1145/1242572.1242667).
- [37] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 12, pp. 2724–2743, Dec. 2017, doi: [10.1109/TKDE.2017.2754499](https://doi.org/10.1109/TKDE.2017.2754499).
- [38] I. Yamada et al., "Wikipedia2Vec: An optimized tool for learning embeddings of words and entities from Wikipedia," *Comput. Lang.*, 2018. [Online]. Available: <https://arxiv.org/abs/1812.06280>
- [39] C. Tu, W. Zhang, Z. Liu, and M. Sun, "Max-margin DeepWalk: Discriminative learning of network representation," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1–7.
- [40] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.
- [41] D. Wang, P. Cui, and W. Zhu, "Structural deep network embedding," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1225–1234, doi: [10.1145/2939672.2939753](https://doi.org/10.1145/2939672.2939753).
- [42] S. Cao, "Deep neural network for learning graph representations," in *Proc. AAAI*, 2016, pp. 1–8.
- [43] F. S. Rizi and M. Granitzer, "Signed heterogeneous network embedding in social media," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, Mar. 2020, pp. 1877–1880.
- [44] P. Frasconi, M. Gori, and G. Soda, "RNNs and prior knowledge for sequence processing: A constrained nondeterministic approach," *Knowl. Based Syst.*, vol. 8, no. 96, pp. 313–332, 1995, doi: [10.1016/0950-7051\(96\)81916-2](https://doi.org/10.1016/0950-7051(96)81916-2).
- [45] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000, doi: [10.1162/089976600300015015](https://doi.org/10.1162/089976600300015015).
- [46] D. Soutner and L. Müller, "Application of LSTM neural networks in language modelling," in *Proc. Int. Conf. Text.* Berlin, Germany: Springer, 2013, pp. 105–112, doi: [10.1007/978-3-642-40585-3\\_14](https://doi.org/10.1007/978-3-642-40585-3_14).
- [47] K. Irie, Z. Tüiske, T. Alkhouli, R. Schlüter, and H. Ney, "LSTM, GRU, highway and a bit of attention: An empirical overview for language modeling in speech recognition," in *Proc. Conf. Int. Speech Commun. Assoc.*, 2016, pp. 3519–3523, doi: [10.21437/Interspeech.2016-491](https://doi.org/10.21437/Interspeech.2016-491).
- [48] Y. Cui, S. Wang, and J. Li, "LSTM neural reordering feature for statistical machine translation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.* San Diego, CA, USA: Association for Computational Linguistics, 2016, pp. 977–982, doi: [10.18653/v1/N16-1112](https://doi.org/10.18653/v1/N16-1112).
- [49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*. [Online]. Available: [http://arxiv.org/abs/1706.03762](https://arxiv.org/abs/1706.03762)
- [50] J. Bromley, J. W. Bentz, and L. Bottou, "Signature verification using a siamese time delay neural network," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, no. 4, pp. 669–688, 1993.
- [51] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Cambridge, MA, USA: MIT Press, 2015, pp. 3294–3302.
- [52] N. Othman, R. Faiz, and K. Smaili, "Manhattan siamese LSTM for question retrieval in community question answering," in *Proc. Int. Conf. Ontologies*, 2019, pp. 661–667.
- [53] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: [http://arxiv.org/abs/1810.04805](https://arxiv.org/abs/1810.04805)
- [54] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*. [Online]. Available: [http://arxiv.org/abs/1907.11692](https://arxiv.org/abs/1907.11692)
- [55] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," 2019, *arXiv:1908.10084*. [Online]. Available: [http://arxiv.org/abs/1908.10084](https://arxiv.org/abs/1908.10084)
- [56] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: [http://arxiv.org/abs/1412.3555](https://arxiv.org/abs/1412.3555)
- [57] A. Graves, "Supervised sequence labelling with recurrent neural networks," in *Long Short-Term Memory*. Berlin, Germany: GER, 2012.
- [58] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in BERTology: What we know about how BERT works," 2020, *arXiv:2002.12327*. [Online]. Available: [http://arxiv.org/abs/2002.12327](https://arxiv.org/abs/2002.12327)
- [59] J. Qadruddin, A. B. Rabiu, R. Walker, R. Soni, M. Gajek, G. Pack, and A. Rangaraj, "Transformer-based language models for similar text retrieval and ranking," 2020, *arXiv:2005.04588v2*. [Online]. Available: <https://arxiv.org/abs/2005.04588v2>
- [60] W. Bao, H. Che, and J. Zhang, "An accurate model for predicting the (Graded) effect of context in word similarity based on bert," 2020, *arXiv:2005.01006*. [Online]. Available: [http://arxiv.org/abs/2005.01006](https://arxiv.org/abs/2005.01006)
- [61] R. Liu, Z. Lin, and W. Wang, "Keyphrase prediction with pre-trained language model," 2020, *arXiv:2004.10462*. [Online]. Available: [http://arxiv.org/abs/2004.10462](https://arxiv.org/abs/2004.10462)
- [62] F. Hill, R. Reichart, and A. Korhonen, "SimLex-999: Evaluating semantic models with (Genuine) similarity estimation," *Comput. Linguistics*, vol. 41, no. 4, pp. 665–695, 2016, doi: [10.1162/COLI\\_a\\_00237](https://doi.org/10.1162/COLI_a_00237).
- [63] R. Speer, J. Chin, and C. Havasi, "ConceptNet 5.5: An open multilingual graph of general knowledge," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2017, pp. 1–9.
- [64] G. Recsik, E. Iklodi, K. Pajkossy, and A. Kornai, "Measuring semantic similarity of words using concept networks," in *Proc. 1st Workshop Represent. Learn. NLP*, 2016, pp. 193–200, doi: [10.18653/v1/W16-1622](https://doi.org/10.18653/v1/W16-1622).
- [65] R. Banjade, N. Mahajan, N. Niraula, V. Rus, and D. Gautam, "Lemon and tea are not similar: Measuring word-to-word similarity by combining different methods," in *Computational Linguistics and Intelligent Text Processing*, 2015, pp. 335–346, doi: [10.1007/978-3-319-18111-0\\_25](https://doi.org/10.1007/978-3-319-18111-0_25).
- [66] R. Schwartz, R. Roi, and A. Rappoport, "Symmetric pattern based word embeddings for improved word similarity prediction," in *Proc. CoNLL*, Beijing, China, 2015, pp. 258–267, doi: [10.18653/v1/K15-1026](https://doi.org/10.18653/v1/K15-1026).



**YU BAI** is currently pursuing the Ph.D. degree with the Nanjing University of Aeronautics and Astronautics, and a Lecturer with Shenyang Aerospace University, China. He has published more than 30 articles. His research interests include AI, NLP, IR, and KG. Since 2017, he sat on the Language and Knowledge Computing Professional Committee on Chinese Information Processing Society of China. He currently serving as the CKO at Global Envoy Software Company Ltd.



**ZHIGUANG WANG** is currently a Graduate Student with the School of Computing, Shenyang Aerospace University, China. His research interests include information retrieval and knowledge graph. He is currently developing machine learning algorithms and human-machine cooperation solutions in information services in the field of IIoT.



**LIANG ZHAO** received the Ph.D. degree from the School of Computing, Edinburgh Napier University, in 2011. He is currently an Associate Professor with Shenyang Aerospace University, China. Before joining Shenyang Aerospace University, he worked as an Associate Senior Researcher with Hitachi (China) Research and Development Corporation, from 2012 to 2014. He has published more than 80 articles. His research interests include ITS, VANET, WMN, and SDN. He was a recipient of the Best Paper Awards at the IEEE ICCC 2015 and the ACM MOPM 2013. He served as the Chair of several international conferences and workshops, including the 2019 IEEE ICCC (Program Co-Chair), the 2019 IEEE Scalcom (Poster/Demo Co-Chair), the IoT-Smart-2015 (Program Co-Chair), the 2019 AI-driven Network Workshop (Program Co-Chair), and the NGDN workshop (founder). He is/has been a Guest Editor of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, the *Journal of Computing* (Springer), and the *Internet Technology Letters* (Wiley).



**JIANJUN CHEN** is currently an Assistant Professor with the Shenyang Northern Software College of Information Technology, China. He joined the Global Envoy Software Company Ltd., in 2002. His research interests include knowledge engineering and knowledge services. As the Director of software engineering, he is currently organizing the research and development of machine learning algorithms and human-machine cooperation solutions in information services in the field of IIoT.



**PENG LIAN** is currently an Assistant Professor with the Shenyang Northern Software College of Information Technology, China. He joined the Global Envoy Software Company Ltd., in 2002. His research interests include knowledge engineering and knowledge services. Since 2012, as the Chief Technology Officer, he presided over the development projects of software systems in the field of IIoT more than 30.

• • •