IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 7, 2019, accepted November 17, 2019, date of publication November 25, 2019, date of current version December 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2955498

Deep Entity Linking via Eliminating Semantic Ambiguity With BERT

XIAOYAO YIN[1], YANGCHEN HUANG[1], BIN ZHOU[1], AIPING LI[1], LONG LAN[1,2], AND YAN JIA[1]

[1]College of Computer, National University of Defense Technology, Changsha 410073, China
[2]State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha 410073, China

Corresponding authors: Long Lan (long.lan@nudt.edu.cn) and Yan Jia (jiayan@nudt.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0802204, Grant 2016QY03D0601, Grant 2016QY01W0101, Grant 2016QY03D0603, and Grant 2017YFB0803301, and in part by the National Natural Science Foundation of China under Grant 61732022, Grant 61732004, Grant 61502517, Grant 61472433, and Grant 61672020.

ABSTRACT Entity linking refers to the task of aligning mentions of entities in the text to their corresponding entries in a specific knowledge base, which is of great significance for many natural language process applications such as semantic text understanding and knowledge fusion. The pivotal of this problem is how to make effective use of contextual information to disambiguate mentions. Moreover, it has been observed that, in most cases, mention has similar or even identical strings to the entity it refers to. To prevent the model from linking mentions to entities with similar strings rather than the semantically similar ones, in this paper, we introduce the advanced language representation model called BERT (Bidirectional Encoder Representations from Transformers) and design a hard negative samples mining strategy to fine-tune it accordingly. Based on the learned features, we obtain the valid entity through computing the similarity between the textual clues of mentions and the entity candidates in the knowledge base. The proposed hard negative samples mining strategy benefits entity linking from the larger, more expressive pre-trained representations of BERT with limited training time and computing sources. To the best of our knowledge, we are the first to equip entity linking task with the powerful pre-trained general language model by deliberately tackling its potential shortcoming of learning literally, and the experiments on the standard benchmark datasets show that the proposed model yields state-of-the-art results.

INDEX TERMS Entity linking, natural language processing (NLP), bidirectional encoder representations from transformers (BERT), deep neural network (DNN).

I. INTRODUCTION

The Internet has entered the era of information explosion, and the problem of overloading information has brought enormous challenges to retrieval. Extracting the massive amounts of information into a structured knowledge base is the key to solve this problem. At the same time, the structured information obtained through information extraction may be redundant and erroneous to a certain extent, and due to the high ambiguity of natural language, it is very significant to connect network data with knowledge base to ensure information quality and understand the semantic information of network data. For now, a large number of knowledge bases represented by Wikipedia appear on the Internet, and they describe knowledge as a network of entities and relations. Linking natural language text in the web pages with entries in the knowledge base not only facilitates users to access, but also improves the accuracy of searching and the relevance of the answers.

Entity Linking (EL) aims at solving such problem, whose task is to associate a specific textual mention of an entity in a given document with an entry in a large target catalog of entities, commonly referred to a knowledge base (KB). Through EL, we can eliminate inconsistencies such as entity conflicts and unclear directions in heterogeneous data, then a large-scale unified knowledge base can be created to help machines to understand heterogeneous data from multiple sources and form high-quality knowledge. And this makes EL one of the primary tasks in the Knowledge-Base Population (KBP) track at the Text Analysis

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

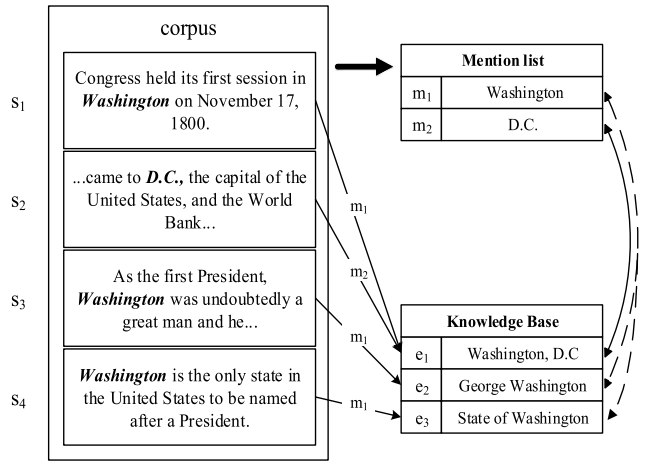

FIGURE 1. Examples for entity linking.

Conference (TAC) [1], [2] which is of great importance in the field of natural language process (NLP). On the one hand, EL helps people to better understand the meaning of the information they browse, and on the other hand, it can help to build an entity-centric information network and promote the development of the Semantic Web. EL has become a central component in building knowledge bases [3], [4] and is crucial for enhancing downstream NLP tasks such as information extraction [5], question answering [6] and search [7].

The target of EL is the entity phrases in the documents containing people, place and institution names which are called mentions in the study. And in the study of entity linking, given a mention m and the entity e in the KB that it refers to, we call m the referential mention of e and e the referred entity of m. In fact, a referred entity often has various expressions when mentioned in different contexts, such as full name, alias, abbreviation, etc., so that its mentions may involve a much larger number of listed words. In Fig. 1, for example, the entity *Washington, D.C* (e_1) is described both in the sentence s_1 and s_2 with two distinct mentions *Washington* and *D.C* respectively. While, remarkably, a mention in the text may also refer to multiple entities in the KB due to the internal ambiguity of natural language. As illustrated in Fig. 1, the same mention Washington (m_1) that appears in different sentences (s_1, s_3, s_4) may refer to the capital of the United States, the first president of American or the State of Washington. Therefore, the main challenge of EL is to solve the uncertainty of mentions and entities.

On the other hand, it is worth noting that with the booming development of deep neural networks (DNN) in natural language processing (NLP), many DNN models have shown excellent representation learning ability. Nowadays, the distributed representation of natural language text will not be susceptible to variations of wording or syntax used to express the same idea and the length of phrases, moreover, the representation of the same phrase can vary with its current semantic. Inspired by this, we found that the ambiguity problem of EL can be well solved as long as the distributed representation of mentions and entities can be established in a unified semantic space, so that the referred entity can be obtained by comparing the similarity of the representation vectors learned through DNN between mention in the text and candidate entities. Furthermore, the recent emergence of pre-trained language models in NLP has brought a brilliant solution to this problem. Instead of training the distributed representation of entities and mentions from scratch, researchers can fine-tune the pre-trained models that have been trained on large-scale corpus in a way that fits their tasks, then the semantic embedding of mentions and entities in the same continuous space can be acquired rapidly. Based on the above findings, we propose to get the mention and entity representation in a unified semantic space by fine-tuning the pre-trained model, and conduct entity linking with the learned embeddings.

To sum up, the main contributions of this work are:

(i) We propose an entity linking model based on BERT, which introduces the idea of pre-training language model into entity linking study. As far as we know, this is the first work using embeddings fine-tuned from BERT for EL.

(ii) We design a hard negative samples mining strategy for choosing the negative samples during the fine tuning of BERT, which drives the model to link mentions and entities with similar semantic information instead of similar strings, the strategy is proved to be efficient in the experiments later.

(iii) We perform experiments on the benchmark datasets CoNLL 2003 and TAC 2010 with accuracy of 95.04% and 90.34% respectively, which is state-of-the-art. What's more, the experiments results show that our model is comparable with some SOTA model using only the semantic features learned from BERT.

The rest of this paper is organized as follows. In Section II, we introduce the entity linking and give a brief review of text representation in NLP. In Section III, we present the benchmark datasets used in this work. Section IV describes the key modules in our model. Section V presents the detailed settings and processing of the experiment, along with the experiment results and analysis. We draw our conclusions in Section VI in the end.

II. RELATED WORK

A. ENTITY LINKING

Linking mentions in the text to a flat set of entities in KBs (e.g., Freebase, Wikipedia) is a long-lasting task in NLP [8]–[11]. During the research, it was found that entity linking is associated with some traditional NLP tasks. Firstly, EL is based on the results of named entity recognition (NER) tasks. After entities in the natural language text are identified by NER, the task of entity linking is to link these entities to specific entries in the knowledge base. Hence, some researchers [12], [13] proposed to conduct NER and EL jointly to make these two tasks mutually reinforcing.

Secondly, the difficulties of entity linking is to solve the ambiguity of natural language, which is similar to the study of word sense disambiguation (WSD). However, WSD is used to solve the specific meaning of a word (rather than a named entity) in the context, while EL must point out the entry the word referred to in the KB. What's more, the sense inventory of words is complete, but the knowledge base is not. Thirdly, coreference resolution, which is the task of finding multiple references to the same object in the document, is just like the entity linking task without knowledge base. The difference between them is that the former requires only to cluster the mentions that refer to the same object in the document (represent a real entity), but the latter needs to further map to an entity in the KB. Finally, entity linking are often confused with entity alignment (also called record linkage). Entity alignment is the task of matching records between several existing databases or knowledge bases that refer to the same entities. In this study, the entities in the knowledge base themselves have a plenty of structured attribute information to determine whether they can be aligned. Entity linking is the process of aligning named entities in unstructured text whose feature information requires extra strategies to extract.

The crucial problem of EL is to find feature representation of mentions and entities in the candidate entity ranking process, which is the fundamental task for the entity linking system. Feature representation is treated as the evidence to sort the candidate entities and select the appropriate entity as the referenced mapping entity for the mention. It can be divided into two categories. The context-dependent features, such as a suitable text window around the mention in the document [14], [15] and the whole [16] or the first description paragraph [17] of its Wikipedia page for each candidate entity, etc. The context-independent features, which just rely on the surface form of the mention and the knowledge about the candidate entity, can be the string comparison between the mention and the candidate entity [15], [18], the entity popularity (the prior probability of the appearance of a candidate entity) [19] or the entity type [20], [21] to constrain the behavior of an entity linking system, and so on.

Fortunately, with the development of text representation, we can use word vectors to take the place of complex manual feature engineering to represent the semantic information of text. In sight of this technique, the semantic features of mention and entity can be embedded into the same vector space for comparison. Learning the representations of mentions and entities for EL has been addressed in past literature. Sun *et al*. [22] depict semantic representations of mention, context and entity and then effectively leverage these vectors in the same continuous space for entity disambiguation. Cao *et al*. [23] present a multi-prototype mention embedding model that proposes a novel concept named mention sense to bridge the text and KB by jointly training words from contexts and entities derived from the KB to deal with the ambiguity in the EL task. Yamada *et al*. [24] describe a DNN model that collectively learns distributed representations of natural texts and entities in the KB to handle various NLP tasks including EL. These researches all used the method of retraining the existing word embedding model (e.g., word2vec) from scratch to obtain the representation of the mention and entity in the same semantic space.

B. FROM TEXT REPRESENTATION TO PRE-TRAINED LANGUAGE MODEL

In the field of NLP, text representation is the first and pivotal step. This transformation is necessary because the digitization of text features is the fundamental part to enable automated processing of the computer.

There is a long history of text representation, and we briefly review it. At the beginning of text representation, the researchers used discrete representation like one-hot coding method and bag-of-words model. They are simple and easy to imply, but the representation is sparse with high dimension and does not consider the sematic information of words in the sentence. As a result, text representation turns to the method of distributed representation [25]. The Neural Network Language model (NNLM) [26] was proposed in 2003, whose intermediate product, namely word embedding matrix, is just the text representation vector we expect to get. In this way, word embeddings are obtained by training a language model on a large-scale corpus, and one of the most widely used and well-known representative work is the word2vec [27]. The work fully proves that the distributed representation in the low-dimensional space not only solves the dimensional disaster problem but also mines the association between words, thus improving the semantic accuracy of vectors.

However, with the increasing complexity of the NLP task and the higher accuracy requirements for word embedding, the shortcomings of word2vec revealed, that is, the inability to solve the problem of polysemy. In other words, for the same word, even if it has different meanings in the context, its vector is unchanged. It is unacceptable for many NLP missions because incorrect semantic information in the vectors can lead to fatal errors in downstream tasks. Therefore, researchers began to train word vectors with context, such as ELMO model [28] whose main idea is to introduce context as new features in the second stage after network structure is pre-trained in the first stage to dynamically adjust embeddings of words. This feature-based pre-training method has achieved good results in solving the problem of polysemous words.

In fact, in the field of computer vision, researchers have repeatedly shown the value of transfer learning through pre-training model, for instance, many works have demonstrated the effectiveness of fine-tuning models pre-trained with ImageNet [29], [30]. Draw inspiration by this, NLP researchers have proposed line of work [31]–[33] lately which involves pre-training a neural network using a language modeling objective and then fine-tunes it on a target task with supervision. This breakthrough makes NLP applications easier, especially for those who don't have sufficient data or equipment to build an NLP model from scratch, which saves a lot of time and computing resources.

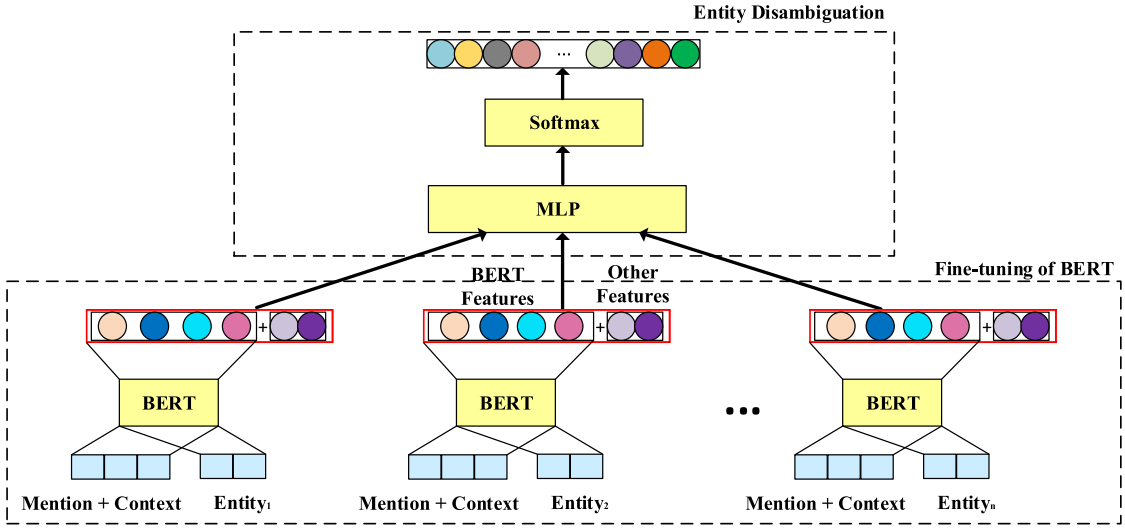

FIGURE 2. The Architecture of our proposed entity linking model.

Among these works, the most popular pre-training general language model is BERT (Bidirectional Encoder Representation from Transformers) [34], which is the first deeply bidirectional, unsupervised language representation, pre-trained with the BooksCorpus (800M words) and Wikipedia (2,500M words) on 4 Cloud TPUs in Pod configuration (16 TPU chips total) for 4 days. BERT's key technical innovation is applying the bidirectional training of transformer, a popular attention model, to get text representations and then the pre-trained BERT network can be fine-tuned as the basis of a new purpose-specific model. In the associated paper, it is demonstrated that BERT achieves state-of-the-art results on 11 NLP tasks, including question answering and language inference. In this approach, a pre-trained neural network produces word embeddings which are then used as features in NLP models.

III. DATASETS

We describe the datasets with which we evaluated our entity linking model as follows and Table 1 summarizes properties of them.

CoNLL 2003 The CoNLL dataset is a popular EL dataset constructed by Hoffart *et al.* [35]. It consists of hand-annotated proper noun annotations with corresponding entities in YAGO2 for 1393 Reuters newswire articles. The dataset is based on the data for the CoNLL-2003 shared task and split into train, test-a and test-b, containing 946, 216, and 231 documents respectively. For the three sets, we only consider mentions with a valid entry in the KB and get a total of 27816 mentions. We trained our model with the training set and tuned the parameters according to the performance on test-a set, which was treated as the development set. Then we reported the standard micro and macro accuracies of the entity linking task on test-b set.

TAC 2010 The TAC 2010 dataset is another widely used EL dataset constructed for the Text Analysis Conference (TAC). It includes news from various agencies and web log data and divides them into training set and test set, containing 1043 and 1013 documents each. Similar with CoNLL data, we keep only the mentions with a valid entry in the KB to train the model. The model was trained with the training set and tested on the test set, besides, we random sample 20% data from the training set as development set for tuning the parameters.

TABLE 1. Datasets properties.

	CoNLL 2003	TAC 2010
articles	1393	2056
mentions(total)	34929	3750
mentions with no entity	7113	1656
words per article (avg.)	188	743
mentions per article (avg.)	25	2
distinct mentions per article (avg.)	18	2
mentions with candidate in KB (avg.)	20	1

IV. METHOD

The network architecture of our model is shown in Fig. 2. The input of BERT is comprised of two parts as sentence A and sentence B. In our task, sentence A is the concatenation of the mention with the context in which it appears, while sentence B is the entity from among the entity candidate list. We fed the inputs into BERT, and got the special classification feature [CLS], the fine-tuned embedding of mention with context and entity and their dot product as the BERT features. Specifically, the [CLS] feature is learned via BERT to indicate the relation of sentence A and sentence B. And for further improvement and comparison, we add several other features which have been proved to be effective in previous work. Finally, we trained an entity disambiguation network with these features to get the referred entities for mentions.

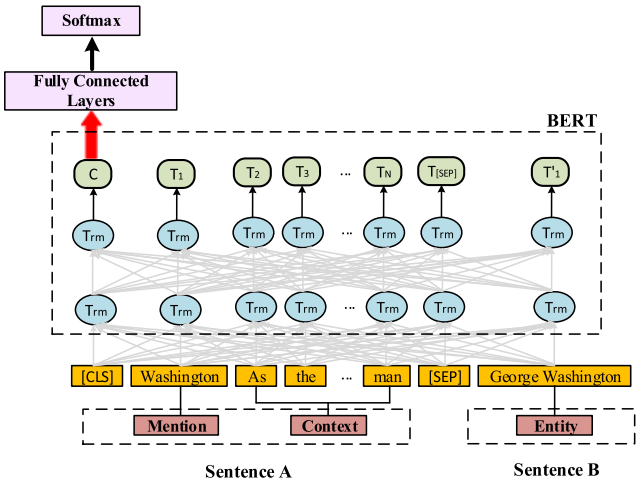

FIGURE 3. The network for fine-tuning BERT.

The key modules of the model will be described in detail in the following sections.

A. TASK FORMULATION

For EL, given a KB containing a set of entities $E = \{e_1, e_2, \ldots, e_j\}$, a text corpus $C = \{d_1, d_2, \ldots, d_n\}$ which consists of documents and each document d_1 contains a list of mentions $M = \{m_1, m_2, \ldots, m_l\}$. The goal of our task is to assign each mention $m_i \in M$ in the document a KB entity $e_i \in E$, and we use English Wikipedia as the KB in this work in which each entity has a unique entity string and ID in the knowledge base according to its properties.

Considering the large size of the corpus, to retrieve the entity that the mention refers to, the first step is to generate an entity candidate list L_e. So that, our task is reduced to choose the best option in L_e. Consequently, the second crucial step is the entity disambiguation part where we calculate a score between the given mention m and the candidate $l_i \in L_e$ to determine the best link.

Formally, given a text corpus C and a textual mention $m \in C$, we define the goal of EL as finding the best link $l_{best} \in L_e$:

$$l_{best} = \arg\max_{l_i \in L_e} P(l_i|m, ct) \tag{1}$$

where $P(l_i|m, ct)$ is the probability of linking m to l_i given m and its context ct.

B. FINE-TUNING OF BERT

Found in the previous studies, it is essential to embed mentions and entities into the same semantic space constraining similar mentions and entities being close to each other, while dissimilar ones being far away.

In this work, as shown in Fig.3, we learn the embeddings by fine-tuning the Google BERT model with a large number of mention-entity pairs from the English DBpedia abstract corpus [36], an open corpus of Wikipedia texts with entity annotations manually created by Wikipedia contributors. We train our model by iterating over the texts and mining the mention-entity pairs in the corpus. In the original BERT model, they utilize the next sequence prediction (NSP) strategy where two sentences A and B are fed into the model to predict whether B comes following A or not. This strategy drives the model to learn better embeddings of both tokens and sentences by mining their semantic information.

Some tricks are developed in constructing the training data of our model to make the training process more reasonable. Firstly, due to mention surfaces in various context may link to different entities, as shown in Fig.1, Washington can be linked to the president or the city according to its context information, we take the sentence containing the mention as the context information to form part of sentence A. Secondly, when there are multiple mentions in a sentence, the model will be confused at which mention we are interested in. To address this problem, we set the sentence A as the union of the mention and the sentence it lies in, which both includes the context information and emphasizes the mention. Generally speaking, we fine tune the model by treating the mention and its context as sentence A while the entity as sentence B, and they are separated with a special token ([SEP]). We convert sentence A and B to indices I and segments S, the indices I represents the id of each token of the sentences in the dictionary of BERT, while the segments S indicates which sentence each token belongs to. The label of mention-entity pair is 1 if there exists a link between the mention and entity and 0 otherwise. Similar with NSP, we predict the relation of mention and entity by adding a fully-connected layer and softmax on top of the [CLS] features from BERT, which is used as a classification signal. For details, the algorithm to learn the embeddings by fine-tuning BERT is shown in Table 2.

C. HARD NEGATIVE SAMPLES MINING IN FINE-TUNING

Fine tuning the BERT model with EL task brings some new problems according our experiment. It has been observed that, in about 80% of cases, mention has similar or even identical strings to the entity it refers to. Under this circumstance, the model will tend to choose a string similar entity over a semantically similar one to link with the mention. Since the NSP task requires balanced data, which means 50% of the training data should be positive meaning that the mention in sentence A refers to the entity in sentence B, and the other 50% data being opposite. In other words, we have to construct a reasonable negative sample for each positive ones. Based on the situation of entity link data described above, we believe that if we select the entity of most similar string with the mention as the negative sample, our model would be driven to mining more semantic information during training. According to all these analyses, we propose a hard negative samples mining strategy which tries to find the most difficult negative sample to distinguish from the corresponding positive one from 957027 entities for each mention. Specifically, its main idea is that when the mention is identical to the ground truth entity, we select the candidate entity which has most similar strings with the ground truth as negative sample, in the opposite case, we take the mention as negative sample.

TABLE 2. The algorithm to fine-tune BERT.

Algorithm FINE-TUNING of BERT

Input: list $\left(Q^1,Q^2,...,Q^N\right)$, where $Q^i=\left(m,ct,e,y\right)^i$; m is the mention surface, ct is the context of m, e is an entity, y is a binary label indicting whether m in ct links to e. The maximum number of epoch as *Epoches*, and the batch number per epoch as *batch_per_epoch*. The base BERT model as *Bert-base* and the maximum sentence length as *max_len*.

Output: Parameters of the fine-tuned BERT model.

For epoch **in** *Epoches*:

For batch **in** *batch_per_epoch*:

1. Concatenate the string m with context ct as the first sentence, the entity e as the second sentence.
2. Tokenize and pad the two sentences to *max_len* and get the indices I and segments S.
3. Feed I and S to *Bert-base* and get the [CLS] feature.
4. Apply fully connected layers and softmax on top of the [CLS] feature to predict the probability p indicting the probability of m linking to e.
5. Minimize the binary cross entropy loss defined on p and y with Adam optimizer.

Early stopping if the loss on validation data does not decrease for certain steps.

For example, the mention '*Reuters*' in context '*There is work on arranging such a meeting hosted by President Mubarak," one PLO official, who requested anonymity, told Reuters.*' is the same as the ground truth entity '*Reuters*'. In this case, we regard the candidate entity '*Reuters Group*' which is the most similar entity with the mention according to string similarity as negative sample. When it comes to the opposite situation, for instance, the mention '*Bill Jordan*' in context '*Bill Jordan, general secretary of the International Confederation of Free Trade Unions (ICFTU), told a news conference the withdrawal of a WTO invitation to ILO director general Michel Hansenne was "outrageous behaviour on the part of an organisation that wants to command respect in the world".*' is not the same as the entity '*Bill Jordan, Baron Jordan*' which it refers to, so we treat the mention '*Bill Jordan*' as negative sample. In this way, it is possible to effectively avoid the tendency of the model to choose a string similar sample with mention as linking object in the fine tuning process, and thus drives the model to deeply explore intrinsic semantic information.

D. ENTITY CANDIDATE GENERATION

For the candidate generation part, the recall of the candidate generation method is an important indicator which points out whether we miss the ground truth entity in the candidate list, so we define it as the proportion of candidate list containing the ground truth entity to the total candidate lists. Intuitively, the upper limit of entity linking accuracy is the recall of candidate generation method, when all the mentions are linked to their ground truth contained in the candidate list. Generally speaking, the recall would be high when the candidate set is large enough, one extreme situation is including all entities as candidate for any mention so that the recall goes up to 100%. However, large candidate set is computationally expensive and brings much redundant information. After comprehensive tradeoff, we decided to generate 30 candidates for each mention.

We use the candidate generation method proposed in [39] for the sake of compatibility with their state-of-the-art results. For the CoNLL 2003 dataset, we generate candidate entities by looking up two different mention-entities dictionaries built using: 1) a public dataset proposed in [40] (denoted as PPRforNED[1]) and 2) a standard YAGO.[2] For the TAC 2010 dataset, we built a dictionary based on the prior data representing the probability of a mention referring to an entity, which was generated from the English Wikipedia dump[3] with the version of 20190601 (earlier versions are no longer available). The prior probability of a mention m linking to an entity e is denoted as $\frac{O_{m,e}}{O_{m,*}}$, where $O_{m,e}$ is the count of mention m referring to entity e, and $O_{m,*}$ is the total occurrence of mention m, regardless of which entity it refers to. After that, entities are sorted by the prior probability to associate with the mention, then we take the top 30 entities as candidates of the mention. The final recall of the candidate generation was 100% and 94.8% on the test sets of the CoNLL 2003 and TAC 2010 datasets, respectively. It should be noted that there exist some errors in the PPRforNED dataset,[4] we did not fix them for fair comparison with other methods since fixing them would improve the model performance slightly.

E. ENTITY DISAMBIGUATION

The entity disambiguation was addressed by a multi-layer perceptron (MLP) with two hidden layers. Our model predicts the association of all the candidate entities with the mention at once instead of individually. The reasons are as follows. Firstly, the key disambiguation step in entity linking is to predict the correlation between the mention and each entity in the candidate list. For a model with 30 candidate entities, each mention needs to be predicted 30 times if we attempt to predict the association of each entity separately, which is much less efficient than predicting that of 30 entities at one time. Secondly, for the entities in the candidate list, all of them have a relatively high probability of being associated with the mention. Taking all candidate entities into account at once, the remaining 29 candidate entities can be used to assist the learning and prediction of the ground truth entity. To some

[1]https://github.com/masha-p/PPRforNED

[2]https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/ambiverse-nlu/aida/downloads/

[3]https://dumps.wikimedia.org/enwiki/20190601/

[4]https://github.com/masha-p/PPRforNED/issues

TABLE 3. Algorithm of the entity disambiguation module.

Algorithm ENTITY DISAMBIGUATION

Input: list $\left(Z^1, Z^2, ..., Z^N\right)$, where $Z^i = (m, ct, F, E, Y)^i$; m is the mention surface, ct is the context of m, $F = \{f_1, f_2, ..., f_{30}\}$ is the prior feature list corresponding to each candidate, $E = \{e_1, e_2, ..., e_{30}\}$ is the candidate list of length 30, which was generated by the Candidate Generation method. Y is the label indicting the index of ground truth entity in E. The maximum number of epoch as *Epoches*, and the batch number per epoch as *batch_per_epoch*. The BERT model as *BERT* to extract embeddings and class feature. *hidden_units* as the number of hidden units.

Output: Parameters of the learned entity disambiguation model.

For epoch **in** *Epoches*:

For batch **in** *batch_per_epoch*:

1. Calculate BERT features $BF = \{bf_1, bf_2, ..., bf_{30}\}$ by feed each triplets $\left(m, ct, \mathrm{e}_j\right)$ into BERT. bf_i is the concatenation of mention features, entity features and the [CLS] label.
2. Concatenate BF with F to get $TF = \{tf_1, tf_2, ..., tf_{30}\}$, where tf_j is all the features used for entity disambiguation.
3. Embedding the feature tf_j into *hidden_units* dimension, and concatenate 30 embeddings to get a *hidden_units**30 dimension feature.
4. Apply dropout, fully connected and Relu layer to further embed the features into dimension 30.
5. Apply fully connected layers and softmax to predict the probability P indicting the probability of each candidate entity as true entity to the mention surface.
6. Minimize the categorical cross entropy loss defined on P and Y with Adam optimizer.

Early stopping if the accuracy on validation data does not increase for certain steps.

extent, it is similar to a complex Siamese network, and its performance is better than a sequential prediction.

Table 3 describes the algorithm of this module. For each candidate entity in the candidates list, we acquire the following features from the fine-tuned BERT as the inputs of MLP, which is denoted as Base features in this work: (1) the vector of the candidate entity, (2) the vector of the mention and context and (3) the dot product of the two vectors. Inspired by [37], [38], we add the *prior* features and *string similarity* features to compare with the Base features in the following experiments. To be specific, the *prior* features include: (1) the prior probability of entity e, denoted as $\frac{O_e}{O_{all}}$ where O_e is the occurrence number of e and O_{all} is the total occurrence of all the entities. (2) the prior probability $\frac{O_{m,e}}{O_{m,*}}$ representing a mention m linking to an entity e in the candidate list, (3) the maximum prior probability of e referred by mentions in the context and (4) the number of candidate entities of m in the corpus. The *string similarity* features are comprised of whether the title of e exactly equals or contains the surface of m, and whether the title of e starts or ends with the surface of m.

The features of the entity disambiguation model can be listed as:

$$\begin{aligned} F_{m,ct,e} = \{&BERT_{m,ct}, BERT_e, BERT_{[CLS]}, BERT_{dot}, prior_e, \\ &prior_{m,e}, \max\_prior_{m,e}, candid\_num_m, e\_equal\_m, \\ &e\_contain\_m, e\_start\_m, e\_end\_m\} \end{aligned} \tag{2}$$

All these constitute a 1546 dimension feature, with dimension 768 for the first two features and 1 for the other features in $\mathrm{F_{m,ct,e}}$. The 1546 dimension feature for each entity was then embedded into a fixed dimension *dim* and then we concatenate the features of each candidate to get a *dim*∗30 dimension feature:

$$CF = \{dense\left(F_{m,ct,e30}\right), \ldots, dense\left(F_{m,ct,e30}\right)\} \tag{3}$$

On top of the concatenated feature, we stack a hidden layer with dropout, and the hidden layer was activated by Relu. We further add a hidden layer and an output layer to predict the most relevant entity using softmax over the entity candidates. This procedure can be formalized as:

$$\bar{y} = softmax\left(dense\left(Relu\left(dense\left(dropout\left(CF\right)\right)\right)\right)\right) \tag{4}$$

And we took categorical cross entropy cost function as the loss:

$$l_p(\bar{y}, y) = -\sum_{i=1}^{N} y_i \log \bar{y}_i \tag{5}$$

where $\bar{y}$ represents the predicted probability, and y is a boolean vector indicating whether the entity candidates are referred by the mention. N is the total number of candidates.

V. EXPERIMENT AND RESULTS ANALYSIS

A. PARAMETER SETTING

The model in this work has two networks to train, including the fine-tuning of BERT and the entity disambiguation module, we will detail the parameter setting separately.

Considering the memory of GPU, we take the BERT base-uncased model for fine tuning, the model consists of 12 layers, with 768 hidden units and 12 heads, so the BERT features we obtained are of length 768. According to the official suggestion[5] of BERT, we set the maximum sentence length as 128 and the mini-batch as 32 to avoid the memory overflow of GPU. The model was implemented using Python and Theano (Theano Development Team, 2016). The training took approximately 12 hours using a 12 GB NVIDIA TITAN XP GPU. We trained the model using stochastic gradient descent (SGD) and its learning rate was set to 1e-5 according to the suggestion of BERT for fine tune.

[5]https://github.com/google-research/bert

TABLE 4. Comparison of the proposed method and the state-of-the-art methods.

		Training time	Computing resource	CoNLL 2003(PPRforNED)		CoNLL 2003(YAGO)		TAC 2010
				Micro accuracy	Macro accuracy	Micro accuracy	Macro accuracy	Micro accuracy
Pershina *et al.* [40]		-	-	91.77	89.89	-	-	-
Globerson *et al.* [41]		-	-	-	-	92.70	-	87.20
Yamada *et al.* [39]		~5 days	40-core CPU on Amazon EC2	93.10	92.60	91.50	90.90	85.50
NTEE [24]		~6 days	NVIDIA K80 GPU	94.70	94.30	-	-	87.70
DeepType [21]		~3 days	Titan X Pascal GPU	94.88	-	-	-	**90.85**
Our method	Base-dot	<12h	Titan X Pascal GPU	89.62	89.55	88.04	87.73	87.11
	Base			92.42	92.57	90.33	89.96	89.35
	Base+*prior*			94.53	93.37	91.79	91.34	90.06
	Base+ *string_similarity*			93.61	92.55	91.05	91.23	89.63
	Base+*prior*+ *string_similarity*			**95.04**	**94.82**	**92.96**	**92.78**	90.34

The entity disambiguation module was trained with stochastic gradient descent (SGD) controlling the learning rate by Adam and we set the mini-batch size to 64. We tuned two hyper-parameters using the micro-accuracy on the development set of CoNLL 2003 with the candidates generated from the PPRforNED dataset since the recall is 100% in this case. The hyper-parameters are the dimension *dim* to embed and the dropout probability. For the former, we tested 500, 200, 100, 50, 30, 10 and 3, while for the latter, we evaluated 0.5, 0.3, 0.1 and 0. Finally, we select *dim* as 10 and dropout probability as 0.1 according to the model's performance on evaluation data.

B. RESULTS AND FEATURE STUDY

We compared the proposed method with the following state-of-the-art methods:

1) Pershina *et al.* [39] proposed a novel graph-based disambiguation approach based on Personalized PageRank (PPR) that combines local and global evidence for disambiguation and effectively filters out noise introduced by in-correct candidates.

2) Globerson *et al.* [40] improved the EL by exploring attention like mechanisms for coherence, where the evidence for each candidate is based on a small set of strong relations, rather than relations to all other entities in the document.

3) Pershina *et al.* [39] proposed a novel embedding method which jointly maps words and entities into the same continuous vector space. They extended the skip-gram model by using the KB graph model and the anchor context model to learn the relatedness of entities and align vectors such that similar words and entities occur close to each other in the vector space.

4) Yamada *et al.* [24] described a novel model of EL named NTEE by learning distributed representations of texts and knowledge base entities from a large amount of entity annotations from Wikipedia. The learning of text representations captures better semantics information than that of learning the representation of words.

5) Raiman *et al.* [21] introduced DeepType, a method explicitly integrated the symbolic information into the reasoning process of a neural network with a type system to address the entity linking problem. They achieved this by reformulating the design problem into a mixed integer problem and solved it with a two steps algorithm.

Table 4 displays the results of these methods compared with our model, and we test the model without dot product of mention with context and entity (denoted as Base-dot), the model with only features generated by BERT (denoted as Base), the model with BERT features and *prior* features (denoted as Base+*prior*), the model with all features (denoted as Base+*prior*+*string_similarity*) seperately. It is shown that our method is comparable to all these methods on both the CoNLL 2003 and the TAC 2010 datasets. Except for the accuracy on TAC 2010 is slightly lower than the best, our model is ahead of the all these models. It is worth noting that, according to the experimental results, only using the Base features generated by BERT in our model is better than some of the state-of-the-art models already.

TABLE 5. Case study of correct samples.

Mention	Ground truth	Middlesbrough F.C.				
Middlesbrough	**Candidate entities**	Middlesbrough F.C.	Middlesbrough	Middlesbrough Theatre	Middlesbrough College	Middlesbrough, Stockton and Thornaby Electric Tramways Company
	BERT score	**0.99**	0.90	2.79e-4	3.68e-5	1.72e-4
	Prediction score	**0.99**	1.01e-3	2.86e-5	2.77e-5	3.89e-5
	Context	Bowyer, who moved to the Yorkshire club in August for 3.5 million pounds ($5.8 million), was expected to play against **Middlesbrough** on Saturday.				
Mention	**Ground truth**	RAI				
RAI	**Candidate entities**	RAI	Rai (title)	Rai Radio 1	Raí	…	Raï
	BERT score	**0.99**	5.70e-3	0.01	**0.99**		**0.99**
	Prediction score	**0.41**	6.03e-3	9.02e-3	7.49e-2		1.31e-3
	Context	Fini told state radio **RAI** he met Mussolini thanks to the good offices of Giuseppe Tatarella, AN's leader in the Chamber of Deputies (lower house), and had overcome their differences.				
Mention	**Ground truth**	Telkom Indonesia				
Telkom	**Candidate entities**	Telkom Indonesia	Telkom (South Africa)	Telkom Kenya	Telkom University	
	BERT score	**0.05**	4.58e-2	0.02	**0.05**	
	Prediction score	**0.63**	0.28	0.02	0.03	
	Context	**Telkom** at $35 in London.				
Mention	**Ground truth**	AS Nancy				
Nancy	**Candidate entities**	AS Nancy	Nancy, France	FC Nancy	Nancy Drew	…	Nancy Sinatra
	BERT score	9.87e-1	0.94	**9.97e-1**	0.11		3.33e-5
	Prediction score	**0.78**	6.69e-3	0.20	2.01e-3		2.07e-4
	Context	Results of French first division matches on Friday: Lens 0 Nantes 4 Paris St Germain 1 **Nancy** 2.				

We further analyzed the contribution of each class of features. The excellent performance on the Base features demonstrates that the features we get from the fine-tuned BERT model indeed capture the semantic information and it also reflects the effectiveness of the hard negative samples mining strategy adopted on the training data. And the result gap between Base-dot and Base shows the importance of dot product which is a crucial measure of the semantic similarity between mention and candidate entity. When we add the *prior* features or the *string similarity* features, the accuracy improves slightly, illustrating that those features are helpful to the entity linking. Interestingly, we observe that the Base+*prior* model outperforms the Base+*string_similarity* model on different dataset and different candidate generation method consistently. We attribute this to the fact that many candidate entities do not make difference on string similarity. For example, the mention '*Isthmus*' referring to entity '*Isthmus of Tehuantepec*' generate candidate list as ['*Isthmus of Tehuantepec*', '*Isthmus (newspaper)*', '*Madison Isthmus*', '*Isthmus of uterine tube*', '*Isthmus Bay*', '*Isthmus of Corinth*']. In this case, the prior probabilities of mention referring to the candidates are [0.35, 0.048, 0.007, 0.005, 0.018, 0.12] plays an important role and drives the model to predict correctly. Taking all the three classes of features together, our model performs impressively better than any other methods.

C. CASE STUDY AND ANALYSIS

We show 6 representative examples from CoNLL 2003 data for case study. The examples shown in Table 5 are 4 mentions

TABLE 6. Case study of error samples.

Mention	Ground truth	European Union						
European	**Candidate entities**	European Union	European Netherlands	European School, Bergen	European Union Public Licence	Europe	…	The European (newspaper)
	BERT score	9.96e-1	2.00e-4	2.52e-5	1.71e-3	**9.97e-1**		9.90e-3
	Prediction score	0.07	1.19e-4	7.23e-6	3.67e-5	**0.92**		1.39e-5
	Context	Relations between Chancellor of the Exchequer Kenneth Clarke and Prime Minister John Major are good despite media reports of a rift over **European** policy, a spokesman for Major's office said.						
Mention	**Ground truth**	Philadelphia						
Philadelphia Eagles	**Candidate entities**	Philadelphia			Philadelphia Eagles			
	BERT score	0.04			**0.99**			
	Prediction score	0.42			**0.57**			
	Context	The injury-plagued Indianapolis Colts lost another quarterback on Thursday but last year's AFC finalists rallied together to shoot down the **Philadelphia Eagles** 37-10 in a showdown of playoff contenders.						

which correctly linked to their ground truth entities while the 2 mentions in the Table 6 are linked incorrectly with our model.

For cases like the first example, BERT features can well represent the semantic information in the context of mention, so as to accurately link mention to the corresponding entity. And as illustrated in Table 4, this situation occupies approximately 92.42% in the corpus, which means that 92.42% of the samples can be accurately linked only using the fine-tuned BERT features. The second situation is that the contextual semantic information of mentions is too vague, and the model cannot determine the referred entity with only using Bert features. In this case, the model works with the help of *string similarity* features and about 1.2% data in the corpus is like this. For the third situation, the mention context also provides few information, and so were the *string similarity* features, then the correct judgment of the model depends on the *prior* features. This kind of sample accounts for about 2.1% of corpus. Finally, in the fourth case, only when *string similarity* features and *prior* features are added together with Bert features can the model work correctly, which occupies approximately 0.5% of all data. This analysis shows that the features obtained by fine-tuning of Bert play a strong guiding role for entity linking and traditional manual features can only be used to improve the model on a small part of data, showing the superiority of the BERT features proposed in this work.

On the other hand, we find from the errors that 63.88% are caused by metonymy mentions, which owe more than one plausible candidates. Further, about half of the metonymy mentions erred in cases when the incorrect entity captures similar semantic information from BERT features but takes a higher prior probability compared with the ground truth entity. Taking the first case in Table 6 as an example, the mention '*European*' referring to the entity '*European Union*' is incorrectly linked to the entity '*Europe*'. In this case, the textual similarity from BERT is not distinguishable, although the string similarity prefers choosing '*European Union*', the prior of '*Europe*' is 0.106 while '*European Union*' is 0.049. This is consistent with the previous observation that *prior* features contribute more than *string similarity* features to the model performance.

Furthermore, another type of mistake occurred when one of the candidate is a part of another which also appears in the mention context, the BERT features prefer to assign higher semantic similarities to the longer candidate. Under this circumstances, the *string similarity* and *prior* features do not make much difference, so the model tends to select the longer one. As the last case in Table 6, the mention '*Philadelphia Eagles*' in sentence '*The injury-plagued Indianapolis Colts lost another quarterback on Thursday but last year's AFC finalists rallied together to shoot down the Philadelphia Eagles 37-10 in a showdown of playoff contenders.*' have the candidate entities '*Philadelphia*' and '*Philadelphia Eagles*'. For the reasons mentioned above, the model chooses the candidate '*Philadelphia Eagles*' to link. About 10.65% errors are caused by this reason.

VI. CONCLUSION

This paper presents a novel method for entity linking by drawing support from the powerful pre-trained language model BERT. The learned features from BERT in a unified semantic space provide crucial clues to disambiguate entities. Extensive experiments demonstrate the efficiency of our model, and we achieve a state-of-the-art performance on two benchmark datasets with the proposed method.

We owe this to two reasons. Firstly, the adoption of the pre-trained language model allows us to transfer the well-trained text representation in large scale corpus into the field

of entity linking, which is both effective and efficient. Secondly, the hard negative samples mining strategy we designed to generate negative samples in fine-tuning BERT selects the entities that are hardest to distinguish from the positive entities, which drives the model to learn more distinguishable semantic features that are essential for the entity linking task.

In summary, the introduction of fine-tuning on BERT and the hard negative samples mining strategy in our model makes it possible to embed the text and entities into a unified semantic space, where similar text and entities are near to each other and dissimilar text and entities are farther. For the further study, we will explore other features that include more clues to improve the performance of entity linking.

ACKNOWLEDGMENT

(Xiaoyao Yin and Yangchen Huang contributed equally to this work.)

REFERENCES

[1] H. Ji, J. Nothman, and B. Hachey, "Overview of TAC-KBP2014 entity discovery and linking tasks," in *Proc. Text Anal. Conf. (TAC)*, 2014, pp. 1333–1339.

[2] H. Ji, J. Nothman, B. Hachey, and R. Florian, "Overview of TAC-KBP2015 tri-lingual entity discovery and linking," in *Proc. TAC*, 2015, pp. 1–25.

[3] B. Min, M. Freedman, and T. Meltzer, "Probabilistic inference for cold start knowledge base population with prior world knowledge," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 601–612.

[4] D. Rao, P. McNamee, and M. Dredze, "Entity linking: Finding extracted entities in a knowledge base," in *Multi-Source, Multilingual Information Extraction and Summarization*. Berlin, Germany: Springer, 2013, pp. 93–115.

[5] Y. Yaghoobzadeh, H. Adel, and H. Schütze, "Noise mitigation for neural entity typing and relation extraction," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1183–1194.

[6] R. Das, M. Zaheer, S. Reddy, and A. McCallum, "Question answering on knowledge bases and text using universal schema and memory networks," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 2, 2017, pp. 358–365.

[7] J. Dalton, L. Dietz, and J. Allan, "Entity query feature expansion using knowledge base links," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2014, pp. 365–374.

[8] R. Bunescu and M. Paşca, "Using encyclopedic knowledge for named entity disambiguation," in *Proc. 11th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2006, pp. 1–8.

[9] G. Durrett and D. Klein, "A joint model for entity analysis: Coreference, typing, and linking," in *Proc. Trans. Assoc. Comput. Linguistics*, vol. 2, Nov. 2014, pp. 477–490.

[10] W. Zeng, J. Tang, and X. Zhao, "Entity linking on Chinese microblogs via deep neural network," *IEEE Access*, vol. 6, pp. 25908–25920, 2018.

[11] W. Zeng, X. Zhao, J. Tang, and H. Shang, "Collective list-only entity linking: A graph-based approach," *IEEE Access*, vol. 6, pp. 16035–16045, 2018.

[12] S. Guo, M.-W. Chang, and E. Kiciman, "To link or not to link? A study on end-to-end tweet entity linking," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2013, pp. 1020–1030.

[13] K. Q. Pu, O. Hassanzadeh, R. Drake, and R. J. Miller, "Online annotation of text streams with structured entities," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 29–38.

[14] L. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2011, pp. 1375–1384.

[15] W. Shen, J. Wang, P. Luo, and M. Wang, "Linking named entities in tweets with knowledge base via user interest modeling," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 68–76.

[16] X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu, "Entity linking for tweets," in *Proc. 51st Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2013, pp. 1304–1311.

[17] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti, "Collective annotation of Wikipedia entities in Web text," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 457–466.

[18] S. Monahan, J. Lehmann, T. Nyberg, J. Plymale, and A. Jung, "Cross-lingual cross-document coreference with entity linking," in *Proc. TAC*, 2011, pp. 1–10.

[19] J. Zhang, J. Li, X.-L. Li, Y. Shi, J. Li, and Z. Wang, "Domain-specific entity linking via fake named entity detection," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2016, pp. 101–116.

[20] S. Murty, P. Verga, L. Vilnis, I. Radovanovic, and A. McCallum, "Hierarchical losses and new resources for fine-grained entity typing and linking," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 97–109.

[21] J. R. Raiman and O. M. Raiman, "DeepType: Multilingual entity linking by neural type system evolution," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[22] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, and X. Wang, "Modeling mention, context and entity with neural networks for entity disambiguation," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 1–7.

[23] Y. Cao, L. Huang, H. Ji, X. Chen, and J. Li, "Bridge text and knowledge by learning multi-prototype entity mention embedding," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1623–1633.

[24] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, "Learning distributed representations of texts and entities from knowledge base," in *Proc. TACL*, vol. 5, Dec. 2017, pp. 397–411.

[25] G. E. Hinton, "Learning distributed representations of concepts," in *Proc. 8th Annu. Conf. Cogn. Sci. Soc.*, vol. 1, 1986, pp. 1–12.

[26] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, Feb. 2003.

[27] 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/pdf/1301.3781

[28] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. NAACL-HLT*, 2018, pp. 2227–2237.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[30] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[31] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 328–339.

[32] M. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1756–1765.

[33] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. (2018). *Improving Language Understanding by Generative Pre-Training*. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/languageunderstandingpaper.pdf

[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2019, pp. 4171–4186.

[35] J. Hoffart, M. A. Yosef, I. Bordino, H. Fürstenau, M. Pinkal, M. Spaniol, B. Taneva, S. Thater, and G. Weikum, "Robust disambiguation of named entities in text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 782–792.

[36] M. Brümmer, M. Dojchinovski, and S. Hellmann, "Dbpedia abstracts: A large-scale, open, multilingual NLP training corpus," in *Proc. 10th Int. Conf. Lang. Resour. Eval. (LREC)*, 2016, pp. 3339–3343.

[37] A. Chisholm and B. Hachey, "Entity disambiguation with Web links," in *Proc. Trans. Assoc. Comput. Linguistics*, vol. 3, 2015, pp. 145–156.

[38] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, "Joint learning of the embedding of words and entities for named entity disambiguation," in *Proc. 20th SIGNLL Conf. Comput. Natural Lang. Learn.*, Berlin, Germany, 2016, pp. 250–259.

[39] M. Pershina, Y. He, and R. Grishman, "Personalized page rank for named entity disambiguation," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2015, pp. 238–243.
[40] A. Globerson, N. Lazic, S. Chakrabarti, A. Subramanya, M. Ringgaard, and F. Pereira, "Collective entity resolution with multi-focal attention," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, Berlin, Germany, 2016, pp. 621–631.

XIAOYAO YIN received the B.S. degree in mathematics from Nanjing University, China, and the M.S. degree in biomedical engineering from the National University of Defense Technology, China, in 2015, where he is currently pursuing the Ph.D. degree with the College of Computer. His research interests include bioinformatics, computer vision, and data mining.

YANGCHEN HUANG has been taking successive postgraduate and doctoral programs of study for doctoral degree at the National University of Defense Technology, Changsha, China. Her current research interests include data mining and natural language processing.

BIN ZHOU is currently a Professor of computer science with the National University of Defense Technology, Changsha, China. His main research interests include web text mining, online social network (OSN) analysis, and big data processing. He has published over 100 research articles (over 20 were SCI indexed and over 60 EI indexed) on these topics. He also received several academic rewards, including two National Science and Technology Progress Awards (second class), four Science and Technology Progress Awards of Hunan Province (first class twice and second class twice). Recently, he has been involved in several international conference program/organization committees relating to OSN and big data processing, such as APWeb2014, ASONAM2014, and CCF Bigdata2014.

AIPING LI received the B.S. and Ph.D. degrees in computer science and technology from the National University of Defense Technology, China, in 2000 and 2004, respectively. Since 2013, he has been a Professor with the College of Computer, National University of Defense Technology. He visited the University of New South Wales, Australia. His research interests include big data processing, uncertain databases, data mining, spatial databases, and time series databases. He has published over 100 research articles on these topics. He also received several academic rewards, including three National Science and Technology progress Awards (second class).

LONG LAN received the Ph.D. degree in computer science from the National University of Defense Technology, in 2017. He was a Visiting Ph.D. Student with the University of Technology, Sydney, from 2015 to 2017. He is currently a Lecturer with the College of Computer, National University of Defense Technology. His research interests include multiobject tracking, computer vision, and discrete optimization.

YAN JIA was born in 1960. She is currently a Professor and a Ph.D. Supervisor with the College of Computer, National University of Defense Technology, China. Her main research interests include database, social network analysis, and data mining.