Knowledge Enhanced Sequential Entity Linking

1st Yu Liu
Institute of Computing Technology, Chinese Academy of Sciences & University of Chinese Academy of Sciences
Beijing, China
liuyu19s@ict.ac.cn

2nd Shi Wang*
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
wangshi@ict.ac.cn

3rd Kangli Zi
Institute of Computing Technology, Chinese Academy of Sciences & University of Chinese Academy of Sciences
Beijing, China
zikangli19b@ict.ac.cn

4th Jicun Li
Institute of Computing Technology, Chinese Academy of Sciences & University of Chinese Academy of Sciences
Beijing, China
lijicun19s@ict.ac.cn

5th Cungen Cao
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
cgcao@ict.ac.cn

2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9533841

***Abstract*—Entity Linking (EL) is the task of mapping mentions in texts to the corresponding entities in knowledge bases. Existing studies mostly focus on joint disambiguation based on the topical coherence, including graph and sequence models. Sequence models alleviate the complexity caused by graph models, but exist the error propagation that incorrectly disambiguated entities are likely to induce further errors when predicting future mentions. Moreover, it is a huge expense to construct the relationship between entities to explore structured knowledge. To address these problems, we propose a novel method, Knowledge Enhanced Sequential Entity Linking (KESEL), which converts global EL into a sequence decision problem and applies a pre-trained language model to better fuse entity knowledge. Specifically, we firstly utilize multiple features to learn local contextual representations of mentions and candidates respectively. Next, a sequential ERNIE model is introduced to generate knowledgeable representations by dynamically integrating the knowledge of previously referred entities into subsequent mentions disambiguation. Finally, by concatenating the above learned contextual and knowledgeable representations, we make full use of multi-semantic information to improve the performance of EL. Extensive experiments show that our method can achieve competitive or state-of-the-art results.**

***Index Terms*—Sequential entity linking, Knowledge enhancement, Pre-trained language model**

I. Introduction

Entity Linking (EL) aligns disambiguated mentions in texts to corresponding entities in a knowledge base (KB). It serves as a fundamental stage in natural language process (NLP) [5], [8], [30], such as question answering, information extraction and knowledge expansion etc. However, this task is challenging due to the mentions' inherent ambiguity. As shown in Figure 1, without considering its context, we assume that the mention "Connecticut" can refer to three entities in Wikipedia, *Connecticut_College*, *State_of_Connecticut* and *Connecticut_River*. Thus, how to determine the target entity among them is our goal in this paper.

*Corresponding author.

Typically, the previous methods can be classified into two categories: local model and global model. Local models [1], [15] rely only on local contexts to determine mentions independently. Global models [8], [27], [28] jointly disambiguate mentions based on topical consistency that entities appearing in the same document share similar topics, generally including graph-based and sequence-based methods. The graph-based methods [5], [9] construct entity graphs for integrating structured information, which perform well but suffer from high complexity. To mitigate this issue, sequence-based methods [7], [8] utilize previously disambiguated entities to facilitate the subsequent entity disambiguation, and achieve a better balance of effectiveness and efficiency. In this paper, we take into account topical consistency by regarding EL as a sequence disambiguation problem. For example in Figure 1, if we already know that "Yale University" refers to the famous *Yale_University*, and there is a relationship "president" between *Peter_Salovey* and *Yale_University* in the KG, then it is apparent to refer "Salovey" to *Peter_Salovey*.

However, the current sequence-based methods primarily adopt reinforcement learning or graph attention network in a sequential manner, which still exist two problems. First, error propagation is the well-known problem in sequence models, the incorrectly predicted entity for the current mention is likely to induce further errors when disambiguating future mentions because the former is utilized in the process of predicting the latter. Second, many researchers [5], [9], [26] tend to incorporate structured knowledge into EL by means of constructing the relationships between entities, but these practices mostly expend tremendous energy and inevitably cause some noise.

Since the emergence of pre-trained language models, they have shown promising results in numerous NLP fields [5], [9], [10], such as machine translation, reading comprehension and question answering etc, which are capable to capture semantic patterns and contain rich knowledge. Moreover, fine-tuning

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

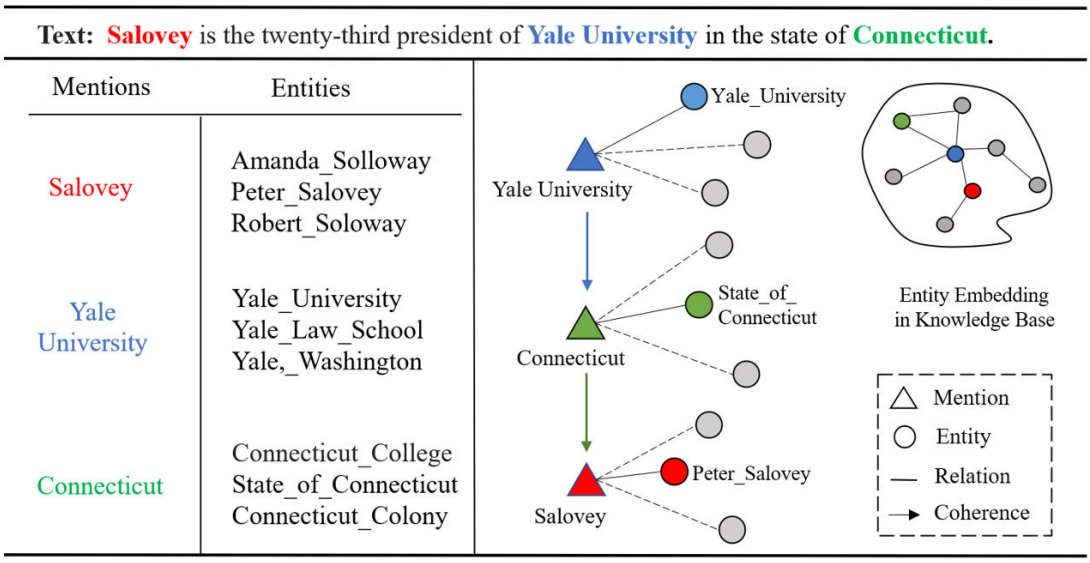


Fig. 1. Illustration of our EL model process. The left is mentions in the free text and their candidate entities in the knowledge base. Each mention is given three candidate entities in Wikipedia. The right is the sequential entity linking process with dynamic structured knowledge. We first rank mentions, and easier mentions are sorted in front of the sequence. Next, knowledge from previously referred mentions is accumulated to enhance future inference. Note that this is just a simple example, it may have more or less than three candidate for each mention.

can reduce the difficulty of these downstream tasks. Motivated by their works, we design a method to address above problems by applying them in sequential EL, and face two challenges: (i) how to perform sequential joint disambiguation to alleviate error propagation; (ii) how to integrate knowledge enhanced pre-trained language models.

In order to tackle these challenges, we propose a new sequential model, Knowledge Enhanced Sequential Entity Linking (KESEL), which explores a pre-trained language model in a sequence way to enhance disambiguation in EL. It consists of two modules: Local Encoder and Global Encoder. For each mention and its candidate entities, local encoder utilizes multiple features to learn their local contextual representations respectively. More importantly, in global encoder, we introduce a sequential ERNIE model, which augments entity linking with dynamic entity knowledge in KG to emphasize topical coherence and decrease model complexity. To our knowledge, we are the first to apply the pre-trained model into EL in a sequentail way. Specifically, mentions are firstly ranked in a sequence according to their ambiguity degree, and for each mention, in addition to its candidates, the previously referred entities are also encoded into global encoder. Then, informative entities in knowledge graph are injected by ERNIE to generate the knowledgeable representation. Finally, we make full use of multi-semantic information by concatenating the learned contextual and knowledgeable representations.

In summary, we make the following contributions:

1. We propose a novel EL model, Knowledge Enhanced Sequential Entity Linking (KESEL), which applies pre-trained language model ERNIE in a sequential way to alleviate the error propagation with knowledge enhancement of previously disambiguated entities.

2. We treat global encoder as a sequence model to capture topical consistency, which makes a trade-off between effectiveness and efficiency in that sequence models not only conduct entity linking from a global perspective but also reduce the complexity of graph models.

3. We conduct extensive experiments on cross-domain datasets, and analyze the impact of key modules, results demonstrate the effectiveness of our proposed method.

II. Related Work

A. Entity Linking

Existing state of the art EL methods can be divided into two categories. Local models independently resolve mentions by focusing on textual information from the surrounding context [1], [15]. The classical DBpedia [21] cast the disambiguation task as a ranking problem, then ranked entities according to the similarity score between mentions and candidate entities. Global models jointly disambiguate mentions based on topical consistency in the same document [6], [9], [32]. To our knowledge, Ganea et al. [2] firstly proposed DeepED with a neural attention-based model, and got an excellent result. Researches [3], [14] showed the benefits of adopting multi-dimensional features into EL, such as multiple relations and entity types. Recently, Fang et al. [7] firstly transformed the global linking into a sequence model, and proposed RLEL model dependent on reinforcement learning to make decisions. They [8] further designed a SeqGA model which combined the advantages of sequence and graph methods. Moreover, many efforts [5], [9], [13] were devoted to graph-based methods, such as PageRank, GCN, or GAT, which utilized structured information to establish the relationship between entities. Although graph-based models achieve better results, they undoubtedly increase the computational complexity.

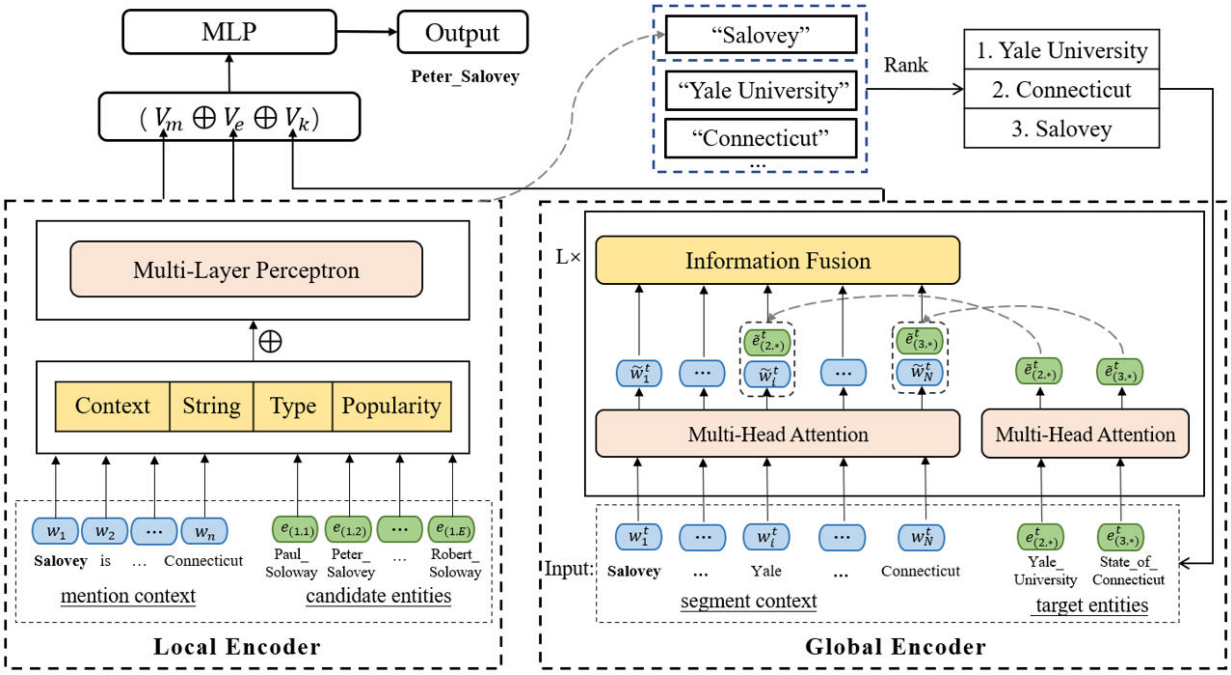


Fig. 2. The overall framework of our novel model **KESEL**. It consists of two parts: Local Encoder and Global Encoder. Local encoder utilizes multiple features to learn contextual representations V_m and V_e of mentions and candidate entities respectively. Global encoder fuses segment context with previous target entities according to the ranking results, to generate the knowledgeable representation V_k. Finally, we complement EL by concatenating three representations and passing them into MLP. For mention "Salovey", its context and candidates are fed into local encoder; Segment context and previous target entities("Yalu University" and "Connecticut") are fed into global encoder. Finally, we determine the target entity is *Peter_Salovey*.

B. Pretrained Language Model

Traditional EL methods employ entity embeddings bootstrapped from word embeddings [2], [15], [29] to carry on linking tasks. With the advance of pre-trained language models, they provide rich semantic representations that have been widely applied in various NLP tasks. Broscheit et al. [12] investigated entity knowledge in BERT [10], [31] that worked surprisingly well. Wu et al. [23] introduced an effective two stage approach for zero-shot linking fine-tuned on BERT model. A bi-encoder independently embedded mention context and entity descriptions. Next, a cross-encoder examined each candidate by concatenating mention and entity text.

In addition, there are numerous variations based on BERT, including ERNIE [11], XLNet [24] and RoBERTa [25] etc. Here we give a brief description to the model ERNIE used in our paper. ERNIE integrates entity knowledge into language representation models, with the knowledgeable aggregator to fuse heterogeneous information from corpora and KGs, and the task dEA to inject informative entities knowledge during pre-training process. It has made remarkable progress in some knowledge-driven tasks (e.g., entity typing and relation classification). Intuitively, correctly predicted entities can contribute to subsequent EL task with additional knowledge in KGs. In this paper, we take advantage of ERNIE, but different from it directly using off-the-shelf tool Tagme to link mentions to corresponding entities, we dynamically determine target entities during the running process, which is the key task of entity linking.

III. Methodology

As shown in Figure 2, the whole architecture of our novel EL model consists of two modules: (1) Local Encoder respectively learns local contextual representations of mentions and candidates; (2) Global Encoder encodes topical coherence in a sequence way to enhance semantic representations by incorporating dynamical knowledge of previously referred entities. In this section, we will present the technical details of our model.

A. Preliminaries

Formally, given a document $\mathcal{D}$ containing a set of mentions $\mathcal{M} = \{m_1, m_2, ..., m_M\}$, and each mention m_i has a set of candidate entities $\mathcal{E}_i = \{e_{(i,1)}, e_{(i,2)}, ..., e_{(i,E)}\}$, where M and E are the number of mentions and candidates respectively. The goal of entity linking is to align mention m_i to its corresponding target entity $e_{(i,*)}$ or return "NIL" if there is no gold entity existed in KB.

Specifically, the task of EL is implemented by two stages: *Candidate generation* selects a set of potential candidate entities in KB, and *Entity disambiguation* ranks all candidates and determines the top one as the target entity. Similar to previous work [2], candidate entities are generated according to both local similarity and prior probability $\hat{P}(e_j|m_i)$ of entity e_j conditioned on mention m_i. The prior probability is the empirical distribution computed from massive web corpus. We choose the top E entities as candidates to ensure a high recall rate and optimize memory space. In our work, we

primarily focus on the second stage entity disambiguation, which is extremely critical for entity linking model. In the following parts, we will present the key components of our novel model, namely local encoder and global encoder.

B. Local Encoder

Given each mention m_i and its candidate entities $\mathcal{E}_i = \{e_{(i,1)}, e_{(i,2)}, ..., e_{(i,E)}\}$, local encoder leverages lexical and statistical features fusion to learn local contextual representations of mentions and candidates respectively.

1) Features Fusion: Local features focus on the compatibility between mentions and candidates. Except for the prior probability referred above, there are three kinds of local features considered in our local encoder:

String Similarity We define string similarity by measuring the similarity between mention surface form and entity title, denoted by $\Psi_{\mathcal{S}}(m_i, e_{(i,j)})$. If their strings are more identical, it indicates that the entity is closer to the mention. For instance, compared with candidates *Paul_Soloway* and *Peter_Salovey*, the mention "Salovey" is more likely to refer to the latter. In practice, we adopt levenshtein distance to compute string similarity.

Type Consistency It is crucial to checks the type consistency between mention and its entity. Here, similar to named entity recognition, we use coarse-grained types information including person names, organizations, locations, and unknown type, which are acquired by a existing system [18]. The definition of type consistency is as follows:

$$\Psi_{\mathcal{T}}(m_i, e_{(i,j)}) = h(m_i)^T \cdot h(e_{(i,j)}) \tag{1}$$

where $h(\cdot)$ denotes a function that converts discrete type values into continuous variables.

Context Compatibility Similar to [2], we first extract n surrounding words as mention context $\mathcal{C}_i = \{w_{(i,1)}, w_{(i,2)}, ..., w_{(i,n)}\}$. Then, an attention mechanism is applied to get contextual representation of mention m_i. Context compatibility is defined as follows:

$$\Psi_C(m_i, e_{(i,j)}) = V_{e_{(i,j)}}^T \mathbf{B}\, V_{m_i} \tag{2}$$

where $\mathbf{B}$ is diagonal matrix, the vector V_{m_i} and $V_{e_{(i,j)}}$ denote the learned contextual representation of mention m_i and its candidate $e_{(i,j)}$ respectively.

2) Training: Finally, we obtain local similarity score $\Psi(m_i, e_{(i,j)})$ by concatenating four features and pass them into a multi-layer perceptron(*MLP*).

$$\Psi(m_i, e_{(i,j)}) = MLP(\Psi_{\mathcal{S}}, \Psi_{\mathcal{T}}, \Psi_{\mathcal{C}}, \hat{P}) \tag{3}$$

With the aim to discriminate the correct gold entity and wrong candidate entities, we utilize a max-margin that ranks ground truth higher than other entities. The loss function is defined as follows:

$$L_{local} = \max\big(0, \gamma - \Psi(m_i, e_{(i,*)}) + \Psi(m_i, e_{(i,j)})\big) \tag{4}$$

where $\gamma > 0$ is a margin parameter and $e_{(i,j)}$ denotes other entities expect for target $e_{(i,*)}$. After training the local encoder, we acquire the local contextual representation of mentions V_m and candidates V_e respectively, which will be utilized for ranking mentions and disambiguating entities in global encoder.

C. Global Encoder

In order to enrich the semantic representation, Global Encoder incorporates knowledge information of former target entities into a sequence method by utilizing a pre-trained language model ERNIE. Note that it is different from ERNIE directly using off-the-shelf tool Tagme to link mentions to corresponding entities, we dynamically determine target entities in the document during the running process.

Figure 3 shows the sequence process of global encoder. We first rank the mentions in the segment according to ambiguity degree, and then add previously predicted entities into the model to guide subsequent mentions disambiguation.

1) Ranking Mention: As we know, mentions usually have different ambiguity according to prior knowledge and contextual information, and mentions with lower ambiguity tend to be easier to disambiguate. Taking the previous example, when it comes to mention "Yale", it has a high probability of being linked to the famous *Yale_University*. Conversely, it is not easy to immediately judge which entity mention "Salovey" should refer to. Therefore, it is significant to take into account the disambiguation order in a sequence model.

As illustrated in the [7], disambiguation difficulty plays an important role to ensure that easier mentions are sorted in front of the sequence. Here, we take advantage of local similarity score $\Psi(m_i, e_{(i,j)})$ to define disambiguation difficulty of mention:

$$\Phi_{m_i} = \max\{\ \Psi(m_i, e_{(i,j)})\ \} \tag{5}$$

where Φ_{m_i} denotes disambiguation difficulty of m_i, the index of candidate entity $j \in \{1, 2, .., E\}$. If the value of Φ_{m_i} greater than Φ_{m_j}, m_i will be ranked before m_j in the sequence.

In general, we firstly divide adjacent mentions into segments by their natural orders in the document according to the observation that the topical consistency decreases along with the mentions distance. Then we arrange the mentions in a segment based on disambiguation difficulty Φ_m and put the mention with the lower ambiguity at the beginning of the sequence.

2) Sequential ERNIE Module: Inspired by pre-trained model ERNIE [11], we attempt to apply a pre-trained language model in global encoder to enhance semantic representation. Then, a sequential ERNIE model is proposed to dynamically integrate knowledge, which can utilize previously referred entities that may promote the subsequent mentions disambiguation, and alleviate computational complexity.

For each segment at time $t \in T$, we connect the sentences of every mention in the segment as context sequence $\mathcal{Q}_c = \{w_1^t, ..., w_N^t\}$, where N is the length of the segment context. Meanwhile, we denote the entity sequence as $\mathcal{Q}_e = \{e_1^t, ..., e_Q^t\}$, where Q is the number of mentions in

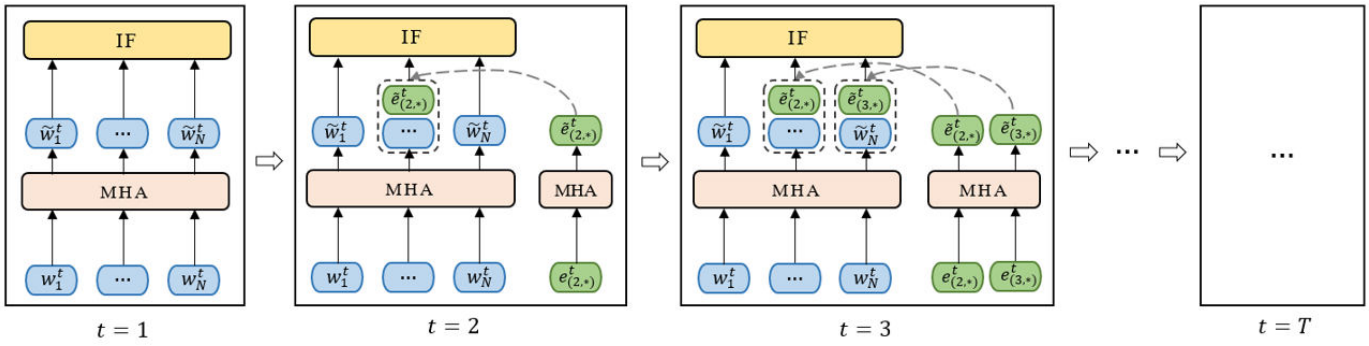


Fig. 3. Process of sequential ERNIE module with time t. Initially, at $t = 1$, no target entity has been generated yet since none of mentions in the sequence have been disambiguated. We thus only input the segment context of the first mention in the sequence into MHA module and calculate candidates scores to obtain the corresponding target entity. At $t = 2$, we feed the segment text of the second mention and the previously disambiguated target entity into MHA, and then fuse them with IF module to obtain target entity. Repeat the above steps until all mentions in the sequence are disambiguated at time $t = T$.

the segment[1]. Note that, the entity sequence consists of the predicted entities of disambiguated mentions in the segment, which changes dynamically over time t. Besides, the global encoder is comprised of L-layers knowledge encoder.

As shown in Figure 2, the knowledge encoder(*K-Encoder*) is designed for encoding both mentions and entities as well as fusing their heterogeneous features, which consists of two sub-modules: multi-head attentions and information fusion. Multi-head attentions [19], namely *MHA*, adopts self attention mechanism to extract multiple semantic meanings from context sequence and entity sequence separately. Information fusion, abbreviated as *IF*, applies a simple multi-layer perceptron to integrate contextual information and structural knowledge to solve their heterogeneous problems.

To be specific, we firstly initialize context sequence with word embeddings, and entity sequence with entity graph embeddings, similar to [11]. In the (l)-th layer at time t, the embeddings of context and entities are fed into two multi-head self attentions respectively,

$$\begin{aligned} \{\tilde{\mathbf{w}}_1^{t(l)}, ..., \tilde{\mathbf{w}}_N^{t(l)}\} &= \mathit{MHA}(\{\mathbf{w}_1^{t(l)}, ..., \mathbf{w}_N^{t(l)}\}) \\ \{\tilde{\mathbf{e}}_1^{t(l)}, ..., \tilde{\mathbf{e}}_Q^{t(l)}\} &= \mathit{MHA}(\{\mathbf{e}_1^{t(l)}, ..., \mathbf{e}_Q^{t(l)}\}) \end{aligned} \tag{6}$$

Then, information fusion layer mutually fuses context of mentions and target entities, and calculates the output embedding for each token and entity by a multi-layer perceptron. The process is as follows:

$$\begin{aligned} \{\mathbf{w}_1^{t(l+1)}, ..., \mathbf{w}_N^{t(l+1)}\}, \{\mathbf{e}_1^{t(l+1)}, ..., \mathbf{e}_Q^{t(l+1)}\} = \\ \mathit{IF}(\tilde{\mathbf{W}}_m^{t(l)} \tilde{\mathbf{w}}_i^{t(l)} + \tilde{\mathbf{W}}_e^{t(l)} \tilde{\mathbf{e}}_j^{t(l)} + \tilde{\mathbf{b}}^{t(l)}) \end{aligned} \tag{7}$$

where $\tilde{\mathbf{W}}^{t(l)}$, $\tilde{\mathbf{b}}^{t(l)}$ are the weight matrix and bias. The output of information fusion will be used as the input of $(l+1)$-th layer. For more details of ERNIE can refer to [11]. Finally, we acquire the knowledgeable representation by taking the final output embedding of special token [CLS] position.

In summary, for each segment at time t, we enter mention context sequence $\{w_1^t, ..., w_N^t\}$ and predicted entity sequence $\{e_1^t, ..., e_Q^t\}$ into global encoder to incorporate knowledge information.

$$V_k = \textit{K-Encoder}(\{w_1^t, ..., w_N^t\}, \{e_1^t, ..., e_Q^t\}) \tag{8}$$

[1]For simplify, we omit the first subscript of mention index in context sequence and entity sequence, e.g., $w_{(i,j)}^t$ to w_j^t, and $e_{(i,j)}^t$ to e_j^t.

where V_k denotes the knowledgeable representation of mentions. Specially, facing with the cold start problem in sequence model, we choose the target entity of the first mention in the segment based on local model results.

3) Joint Representations: Aiming to combine the global interdependence with the local compatibility, we concatenate the local contextual representation and global knowledgeable representation. For mention m_i and its candidate entity $e_{(i,j)}$, the concatenated vector is as follows:

$$V_{(m_i, e_{(i,j)})} = V_{m_i} \oplus V_{e_{(i,j)}} \oplus V_{k_i} \tag{9}$$

where $\oplus$ represents vector concatenation. V_{m_i} and $V_{e_{(i,j)}}$ respectively denote the local contextual vector of m_i and $e_{(i,j)}$, V_{k_i} is the global knowledgeable representation of m_i. Then we feed this vectors into a multi-layer perceptron with a softmax function to calculate the probability of each candidate entity:

$$p(\hat{y} = e_{(i,j)}) = \mathit{softmax}(\mathbf{W}_v * V_{(m_i, e_{(i,j)})} + \mathbf{b}_v) \tag{10}$$

where $\mathbf{W}_v$, $\mathbf{b}_v$ are the weight matrix and bias for the multi-layer perceptron, and $p(\hat{y} = e_{(i,j)}) \in (0, 1)$. To train the global encoder, we adopt the following cross entropy loss function:

$$L_{global} = -\sum_{j=1}^{n} y \log p(\hat{y} = e_{(i,j)}) \tag{11}$$

where $y \in \{0, 1\}$ indicates the true label of the candidate entity, where 1 means a correct target entity, and 0 otherwise.

IV. Experiment

To verify the effectiveness of our method, we train our model and validate it on standard benchmark datasets that are also used by [2], [8], [9], [17]. We will introduce experiment setup, show results and analyse performance in the following sections. Like most previous models, we adopts Micro F1 as the evaluation metric, which is a trade-off between precision and recall. Our source code will be available at https://github.com/casict-kgan.

A. Experiment Setup

1) Datasets: We conduct experiments on a series of popular datasets considering in-domain and out-domain setting. For in-domain setting, we utilize AIDA-CoNLL [1] for training (AIDA-train), validation (AIDA-A) and testing (AIDA-B), which correspondingly includes 946, 216, and 231 documents. For out-domain setting, we validate the model on the following five test sets. MSNBC, AQUAINT, and ACE 2004 datasets are cleaned and updated by [16], which contain 20, 50 and 36 documents respectively. WNED-CWEB

TABLE I

MICRO-AVERAGED F1 ON BENCHMARK DATASETS IN DIFFERENT METHODS. THE BEST SCORES ARE IN BOLDFACE AND THE SECOND-BEST ARE UNDERLINED. THE UPPER PART IS THE RESULT OF LOCAL MODELS, AND THE LOWER PART IS THE RESULT OF GLOBAL MODELS.

System	AIDA-B	MSNBC	AQUAINT	ACE2004	CWEB	WIKI	Avg
Local models							
Prior	71.9	89.3	83.2	84.4	69.8	64.2	77.13
DeepED(local) [2]	88.8	90.59	86.01	87.73	73.64	75.14	83.6
DGCN(local) [9]	89.0	91.0	86.5	**89.2**	**75.5**	74.2	83.3
our(local)	**91.73**	**93.04**	**87.27**	88.53	72.13	**75.62**	**84.77**
Global models							
AIDA [1]	-	79	56	80	58.6	63	67.32
WNED [17]	89.0	92	87	88	77	**84.5**	86.25
DeepED [2]	92.22	93.7	88.5	88.5	77.9	77.5	86.38
Ment-Norm [3]	93.07	93.9	88.3	89.9	77.5	78.0	86.77
RLEL [7]	94.3	92.8	87.5	**91.2**	78.5	82.8	**87.85***
DGCN [9]	93.13	92.5	**89.4**	90.6	**81.2**	77.6	87.41
our(global)	**94.48**	**94.27**	88.33	89.47	78.4	81.24	87.70

and WNED-WIKI [17] are larger but less reliable, automatically extracted from ClueWeb and Wikipedia corpus with 320 articles each.

2) Baselines: For the sake of fairness, we compare our model against EL systems that report state-of-the-art results on the test datasets. Specifically, the benchmark methods can be divided into two categories, local models and global models.

Local models: Prior model only uses the prior popularity to rank candidates. DeepED(local) [2] proposes a neural attention-based model. DGCN(local) [9] utilizes local and global features(except dynamic graph updating feature) to disambiguate mentions.

Global models: AIDA [1] constructs a dense subgraph of entities that approximates the best joint mention-entity mapping. WNED [17] applies random walk and uses iteratively greedy algorithm to link mentions. Ment-Norm [3] introduces multi- relational to entity linking model. RLEL [7] converts the global linking into a sequence decision model with reinforcement learning algorithm. DGCN [9] presents a dynamic graph convolutional network model for characterizing the connections between mention-entity pairs.

3) Settings: In candidate generation, we directly use the candidates provided by [2]. Figure 4 shows the gold recall on different datasets, we can see that when the number of candidate entities reaches 6, the recall rate of target entities in the candidate entity set tends to be stable. Thus we select top 6 candidates to ensure high recall and memory optimization. For fair comparison, the settings of our local model are same as DeepED(local) where the length of mention context n is 100, and the dimensions of word and entity embedding are 300. For training, the batch size is 64, and the rank margin γ is set to 0.1. In addition, we set the number of MLP layers to 2, and expand four features by 5 times, so that the feature dimension is expanded to 20. In our global encoder, we choose the 3 adjacent mentions to form a sequence, so time T equals 3. Following the ERNIE model, we set a maximum segment context length N to 256, the dimension of hidden representation to 768, and the layers of knowledge encoder L to 6. In fine-tuning process we set epoch to 3 and batch size to 32 to avoid overfitting. All global features are fed into a two-layer multi-layer perceptron with 100 hidden units and dropout rate of 0.1, then we apply Adam as optimizer with learning rate of 5e-3. Early stop trick is adopted in both encoders when the performance is not improved. Our model is implemented in Pytorch framework and trained on Nvidia Tesla V100 GPU.

*RLEL achieves the best results, but it uses additional datasets in Wikipedia for training which is not public in the paper.

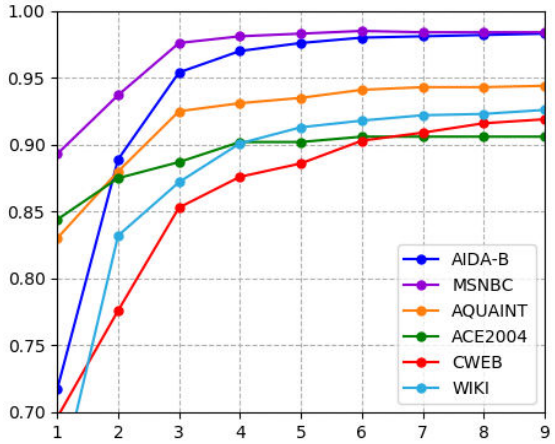


Fig. 4. Gold recall on different datasets. Gold recall refers to the percentage of mentions that the candidate entity contains the groundtruth entity, and the horizontal axis represents the number of entities in the candidate entity set, and the vertical axis represents the gold recall rate.

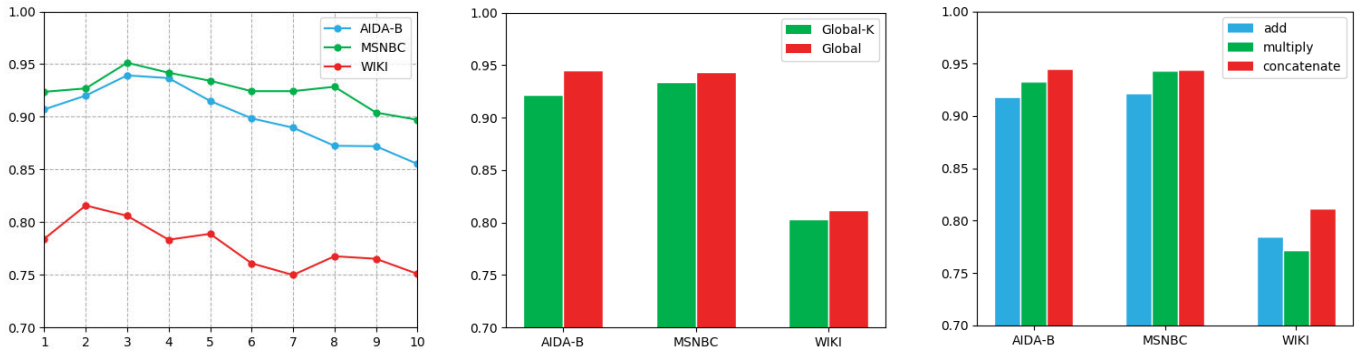
TABLE II

SELECTION OF DIFFERENT LOCAL MODELS. BERT IS THE PRE-TRAINED UNCASED BERT-BASE MODEL, BERT+MF IS FUSED WITH MULTIPLE FEATURES(MF) BASED ON BERT, DEEPED(LOCAL) IS ONLY THE LOCAL PART OF DEEPED, AND OUR(LOCAL) IS THE LOCAL ENCODER IN KESEL.

Models	AIDA-B	MSNBC	WIKI	Avg
BERT	83.10	81.25	59.60	74.73
BERT+MF	86.42	89.21	68.44	81.36
DeepED(local)	88.80	90.59	75.14	84.84
our(local)	91.73	93.04	75.62	86.80

B. Experimental Results

Table I shows experimental results on benchmark datasets. The methods are divided into two groups: local models and global models. As shown, Our local model achieves the best result on average F1 score. We utilize multiple features fusion to enhance local model performance. To analyze the influence of features, we compare our model with Prior model and DeepED(local) model. Experiments show prior popularity $\mathcal{P}$ plays a big role, which adheres to our



(a) Effect of sequence length (b) Comparison of global variants (c) Study of vector Combination

Fig. 5. The comparative experiments of our model. (a) Its horizontal axis represents the length of sequence, which is also the number of adjacent mentions divided in each segment. (b) Global represents our proposed model KESEL, and Global-K represents removing knowledge from our global model. (c) Three ways: addition, multiplication and concatenation are adopted to combine the contextual and knowledgeable representations respectively.

TABLE III
COMBINATION OF DIFFERENT FEATURES IN THE LOCAL ENCODER. THE FOUR FEATURES ARE CONTEXT COMPATIBILITY $\mathcal{C}$, PRIOR PROBABILITY $\mathcal{P}$, STRING SIMILARITY $\mathcal{T}$, AND TYPE CONSISTENCY $\mathcal{T}$.

Features	AIDA-B	MSNBC	WIKI	Avg
$\mathcal{C}$	80.22	79.11	64.79	74.71
$\mathcal{P}$	71.90	89.30	64.20	75.13
$\mathcal{C}$+$\mathcal{P}$	88.80	90.59	75.14	84.84
$\mathcal{C}$+$\mathcal{P}$+$\mathcal{T}$	87.98	91.51	75.35	84.94
$\mathcal{C}$+$\mathcal{P}$+$\mathcal{L}$	90.79	93.19	75.44	86.47
$\mathcal{C}$+$\mathcal{P}$+$\mathcal{T}$+$\mathcal{L}$	91.73	93.04	75.62	86.80

intuition that well-known entities have a high probability to be mentioned in the real world. Compared with DeepED(local) based on $\mathcal{C}$ and $\mathcal{P}$ features, our local model acquires great improvements of 1.40% in average F1. Furthermore, in comparison with DeepED and DGCN, it is worth noting that the better the local model, the better the global model.

In global models, We observe that novel global model outperforms all previous methods on both AIDA-B and MSNBC dataset, despite the WNED, RLEL and DGCN models achieve the best performance on other datasets. Since the low-quality datasets CWEB and WIKI, there are more noises among mentions, our sequential model is not so good. Moreover, our global model yields a competitive average result, only lower than RLEL which also regards EL as a sequence problem, but it utilizes additional training set from Wikipedia data. Overall, our model performs well on most of datasets, which reflects the effectiveness of augmenting dynamic knowledge to a sequence model.

C. Performance Analysis

1) Influence of Sequence length: In order to analyze the effect of the number of mentions in a segment on global disambiguation, we conduct experiments on sequences of different lengths. Figure 5(a) presents the results on three datastes. As we can see, for artificial datasets AIDA-B and MSNBC, when the length of the sequence reaches 4 or more, noise data may be introduced, so the F1 value shows a downward trend. In the low-quality dataset WIKI, the connection between mentions may not be very close, when the sequence length is greater than 2, the disambiguation result has become worse. Comprehensively, we select 3 adjacent mentions to form a sequence.

2) Selection of local models: Since the advent of BERT, it has been applied in many NLP tasks. To explain why it was not selected in our local model, we conduct experiments on different local models. Mention context and entity description are concatenated and input to BERT trained with a max-margin loss. The experimental results are shown in Table II. We can see that BERT even fused with multiple features(MF) performs poor on popular datasets, which is consistent with the conclusion in SeqGAT [8]. We think the local DeepED takes advantage of entity embeddings [2] which may learn topic-level entity relationship, while BERT only captures the contextual feature. Compared with DeepED(local), our local model also achieves significant F1 improvements of 2.31%.

In order to explain whether it has a significant influence on the effectiveness by using some of these features in the local encoder, we conduct an experiment under different features combinations. As shown in Tabel III, we can observe that $\mathcal{C}$ and $\mathcal{P}$ have a significant influence, while $\mathcal{T}$ and $\mathcal{L}$ have a slight improvement on our local model. The fusion of four features achieves the best result, which is greatly improved by 16.2% on average F1 compared with only one context feature $\mathcal{C}$.

3) Comparison of global variants: In global model, we incorporate the knowledge of previously disambiguated entities into sequential entity linking. With the aim to judge whether entity knowledge contributes to entity linking, we remove the knowledgeable representation V_k from the global encoder. The experimental results are shown in Figure 5(b), where the model Global-K represents removing knowledge from our global model. From it, we can draw a conclusion that the previously referred entities have a great effect on the disambiguation of subsequent entities. By adding the knowledge contained in KG into sequential entity linking, the performance of our model is considerably improved.

Moreover, we concatenate local contextual representations V_m and V_e, as well as global knowledgeable representationV_k in global model. Motivated by the input of BERT model which adds the token, sentence and positional embeddings together, we attempt to add the three representations. Besides, there are similar relations between local and global model, so we multiply them. Figure 5(c) displays the comparison results. We see that the concatenation method is the best among the three methods. The explanation is that concatenation expands the dimension of features and has ability to express more information.

V. CONCLUSION

In this paper, we propose a novel model, Knowledge Enhanced Sequential Entity Linking (KESEL), which regards global EL as a sequence model and adopts a pre-trained language model to better incorporate entity knowledge. KESEL consists of two modules, local encoder utilize multiple features to learn contextual representations

respectively, and global encoder has the ability to augment EL with dynamic entity knowledge in KG to emphasize topical consistency based on a sequential ERNIE model. In general, we take full advantage of multi-semantic information, including lexical, statistical, and structured knowledge simultaneously for collective EL model. Experiments show the effectiveness of our model. In the future, we plan to to build the entity graph with reasoning entities dynamically during the linking process. We would like to explore richer information by joining external knowledge graph containing sufficient relationships among entities, which can find strong associations to enhance the performance of EL.

VI. Acknowledgements

This research is supported by the National Key Research and Development Program of China (2017YFB1002300,2017YFC1700300), National Natural Science Foundation of China (61702234), Beijing NOVA Program (Cross-discipline, Z191100001119014).

References

[1] 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," *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pp. 782-792, 2011.

[2] O. E. Ganea and T. Hofmann, "Deep joint entity disambiguation with local neural attention," *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2619-2629, 2017.

[3] P. Le and I. Titov, "Improving entity linking by modeling latent relations between mentions," *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1595-1604, 2018.

[4] M. C. Phan, A. Sun, Y. Tay, J. Han, and C. Li, "NeuPL: Attention-based semantic matching and pair-linking for entity disambiguation," *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1667-1676, 2017.

[5] Y. Cao, L. Hou, J. Li, and Z. Liu, "Neural collective entity linking," *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 675-686, 2018.

[6] X. Yang, X. Gu, S. Lin, S. Tang, Y. Zhuang, F. Wu, Z. Chen, G. Hu, and X. Ren, "Learning dynamic context augmentation for global entity linking," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 271-281, 2019.

[7] Z. Fang, Y. Cao, Q. Li, D. Zhang, Z. Zhang, and Y. Liu, "Joint entity linking with deep reinforcement learning," *The World Wide Web Conference*, pp. 438-447, 2019.

[8] Z. Fang, Y. Cao, R. Li, Z. Zhang, Y. Liu, and S. Wang, "High quality candidate generation and sequential graph attention network for entity linking," *Proceedings of The Web Conference 2020*, pp. 640-650, 2020.

[9] J. Wu, R. Zhang, Y. Mao, H. Guo, M. Soflaei, and J. Huai, "Dynamic graph convolutional networks for entity linking," *Proceedings of The Web Conference 2020*, pp. 1149-1159, 2020.

[10] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 4171-4186, 2018.

[11] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu, "ERNIE: Enhanced language representation with informative entities," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1441-1451, 2019.

[12] S. Broscheit, "Investigating entity knowledge in BERT with simple neural end-to-end entity linking," *Proceedings of the 23rd Conference on Computational Natural Language Learning*, pp. 677-685, 2020.

[13] Ö. Sevgili, A. Panchenko, and C. Biemann, "Improving neural entity disambiguation with graph embeddings," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 315-322, 2019.

[14] S. Chen, J. Wang, F. Jiang, and C. Y. Lin, "Improving entity linking by modeling latent entity type information," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, pp. 7529-7537, 2020.

[15] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, "Joint learning of the embedding of words and entities for named entity disambiguation," *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 250-259, 2016.

[16] L. Ratinov, D. Roth, D. Downey, and M. Anderson, "Local and global algorithms for disambiguation to wikipedia," *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 1375-1384, 2011.

[17] Z. Guo and D. Barbosa, "Robust Named entity disambiguation with random walks," *Semantic Web*, vol. 9, no. 4, pp. 459-479, 2018.

[18] J. Xin, Y. Lin, Z. Liu, and M. Sun, "Improving neural fine-grained entity typing with knowledge attention," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N., Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000-6010, 2017.

[20] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Neural Information Processing Systems*, pp. 1-9, 2013.

[21] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer, "DBpedia spotlight: Shedding light on the web of documents," *Proceedings of the 7th international conference on semantic systems*, pp. 1-8, 2011.

[22] L. Logeswaran, M. W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee, "Zero-shot entity linking by reading entity descriptions," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3449-3460, 2019.

[23] L. Wu, F. Petroni, M. Josifoski, S. Riedel, and L. Zettlemoyer, "Scalable zero-shot entity linking with dense entity retrieval," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 6397-6407, 2020.

[24] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," *Advances in Neural Information Processing Systems 32*, pp. 5753-5763, 2019.

[25] 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," *arXiv preprint arXiv:1907.11692*, 2019.

[26] K. Sayali, A. Singh, G. Ramakrishnan, and S. Chakrabarti. "Collective annotation of wikipedia entities in web text." *In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 457-466. 2009.

[27] K. Nikolaos, O. Ganea, and T. Hofmann. "End-to-End Neural Entity Linking." *In Proceedings of the 22nd Conference on Computational Natural Language Learning*, pp. 519-529. 2018.

[28] L. Phong, and I. Titov. "Improving Entity Linking by Modeling Latent Relations between Mentions." *In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 1595-1604. 2018.

[29] Y. Ikuya, H. Shindo, H. Takeda, and Y. Takefuji. "Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation." *In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 250-259. 2016.

[30] L. Phong, and I. Titov. "Boosting entity linking performance by leveraging unlabeled documents." *arXiv preprint arXiv:1906.01250* 2019.

[31] L. Lajanugen, M. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee. "Zero-Shot Entity Linking by Reading Entity Descriptions." *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3449-3460. 2019.

[32] H. Feng, R. Wang, J. He, and Y. Zhou. "Improving Entity Linking through Semantic Reinforced Entity Embeddings." *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6843-6848. 2020.