INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Entity Linking for Short Text Using Structured Knowledge Graph via Multi-grained Text Matching

Binxuan Huang[1], *Han Wang*[2], *Tong Wang*[2], *Yue Liu*[2], *Yang Liu*[2]

[1]Carnegie Mellon Unversity
[2]Amazon Alexa AI

binxuanhuang@gmail.com, {wnghn,tonwng,lyu,yangliud}@amazon.com

Abstract

Entity Linking (EL) recognizes textual mentions of entities and maps them to the corresponding entities in a Knowledge Graph (KG). In this paper, we propose a novel method for EL on short text using entity representations base on their name labels, descriptions, and other related entities in the KG. We then leverage a pre-trained BERT model to calculate the semantic similarity between the entity and the text. This method does not require a large volume of data to jointly train word and entity representations, and is easily portable to a new domain with a KG. We demonstrate that our approach outperforms previous methods on a public benchmark dataset with a large margin.

1. Introduction

Entity Linking (EL) is the task of recognizing named entities in the text and disambiguating them with the corresponding entities in a Knowledge Graph (KG) such as Freebase and Wikidata. For example, in the question “which basketball player is married to monica?”, we need to link “monica” to the American singer “Monica” who married to NBA player Shannon Brown. EL is an important component for many applications related to KGs, such as question answering over KG [1], conversational AI [2]. In voice-based intelligent assistant system such as Alexa, EL is an essential component to fulfill users’ requests of taking actions on entities [3].

A majority of research in EL targets long documents such as Wikipedia abstracts [4]. Most of the previous methods are evaluated over datasets such as TAC-KBP [5, 6], AIDA-CoNLL [7], and ACE [8]. In the case of short text, which is the task we are focusing on in this paper, there is usually no sufficient context information for the disambiguation task, and the text is often transcribed spoken language or written in an informal format. Researchers have proposed several EL methods designed specifically for short text, such as Tagme [9], S-MART [10], and Falcon [11].

Recent work uses deep neural networks to learn the similarity between entity candidates and the mention along with its context in the text. LSTMs and BERT models were used to encode the contexts of mentions as well as entity descriptions respectively in [12] and [13]. However, these approaches ignore the structure information of the KG, such as the relations between entities specified in the KG. Some other work explicitly utilizes the KG structure by learning KG embeddings. [14] learns KG embeddings using real and complex bilinear maps, and generates the final similarity score using such KG entity embeddings and CNN-based context representations. [15] further adds relation embeddings to their ranking model. Their entity embeddings and relation embeddings are pre-trained using TransE [16]. One potential limitation of using pre-trained KG embeddings is that the KG embeddings and text representations are not learned in the same vector space. To overcome this issue, some work tries to jointly learn KG and text embeddings. [17] combines three alignment models to guarantee the vectors of entities and words are in the same space. [18] also proves that aligning word representation and entity embedding is beneficial for entity disambiguation.

In this paper we propose a method that differs from previous work in: (I) We not only utilize an entity’s name label and description, but also involve its connected neighbors in the KG. We encode this rich representation of the entities directly using a language model pre-trained on large scale unlabeled data, specifically the BERT model, Hence, such representations are naturally aligned with the text representations. (II) We do not use rules, but aggregate multi-grained text matching similarities in a ranking model. (III) Our method does not require a large data set for the task domain to jointly train the entity and word embeddings.

Our contributions are thus the following: (1) We propose a new method for representing entities in the KG, and based on such semantic representation, we perform multi-grained text matching to derive a final similarity score between the text and an entity. (2) We demonstrate that our EL system achieves the state-of-the-art performance on a public benchmark dataset, and conduct detailed analysis to understand our model’s strength and limitation.

2. Method

Similar to other EL methods, our system has three components: (1) Entity tagging that identifies all the entity mentions in the text; (2) Candidate retrieval from the KG for the extracted mentions; and (3) Entity ranking that evaluates the retrieved entity candidates using various features and selects the best candidate. Our key contribution is in the third component, which we will describe in details after we briefly explain the first two parts.

The first step of entity mention detection is considered as a sequence tagging problem, similar to named entity recognition [19]. Here we do not differentiate among different types of entities. With BIO tagging schema [20], we assign the tag “B”, “I”, and “O” to tokens at the beginning of, inside, and outside the entity mentions, respectively. We fine-tune a BERT base model [21] using the training data for this task, that is, we feed the final hidden representation of each token into a classification layer over the BIO tag set and fine-tune the whole model.

For the second step of entity candidate search, we first create an ElasticSearch [22] index with the entity labels from the KG. For each mention, we apply both exact match and Levenshtein edit distance based fuzzy match. To be resilient to possible errors made in the mention detection step, we also expand or shrink the entity mention span hypothesis by one token, and add the search

http://dx.doi.org/10.21437/Interspeech.2020-1934

results for these adjusted mentions to the entity candidate set as well.

2.1. Entity Ranking: Entity Fitness Scores

We propose to represent each entity candidate using three kinds of information from the KG: entity name labels, entity descriptions, and relations between entities, and thus use these to measure the goodness of each candidate via multi-grained text matching.

2.1.1. (A) Character-level Similarity Based on Entity Names

This measures the surface form similarity between the extracted mention and the name label of an entity candidate. Rather than using the string edit distance, we use a CNN to extract the character-level features. For each character in a candidate's name, we map it to an embedding and then apply a convolutional filter to obtain a feature vector $\boldsymbol{\theta^e} = [\theta^e_1, \theta^e_2, ..., \theta^e_{k-w+1}] \in R^{k-w+1}$, where k is the number of characters in the entity label and w denotes the filter window width. With n filters, we can get n such feature vectors and form a feature matrix $M^e \in R^{(k-w+1)\times n}$. Similarly, we apply the same n convolutional filters to the mention and compute its feature matrix $M^t \in R^{(j-w+1)\times n}$, where j is the number of characters in the mention.

Inspired by previous work [23, 24], we compute a character-level similarity matrix S^c, where each element S^c_{ij} represents the correlation between w consecutive characters in the entity mention and the candidate's label:

$$S^c_{ij} = cosine(M^t_{i,:}, M^e_{j,:})$$

2.1.2. (B) Token-level Similarity Based on Entity Descriptions

This measures the similarity between the text and the description of an entity candidate, in order to evaluate if the entity matches the context semantically. We use a BERT-base model to get the semantic representation of each token. For the entity description token sequence $(w^e_1, w^e_2, ..., w^e_n)$, we generate a semantic feature matrix $V^e \in R^{n\times D}$, where D is the dimension of word representation vectors. Similarly the semantic feature matrix for the text with m tokens is $V^t \in R^{m\times D}$. Now, we compute a token-level similarity matrix S^w, where each element S^w_{ij} represents the semantic similarity between one word w^e_i in the candidate's description and w^t_j in the text:

$$S^w_{ij} = cosine(V^t_{i,:}, V^e_{j,:})$$

2.1.3. (C) Similarity based on Neighboring Entities

This still aims at measuring if an entity candidate matches the text context. But for each entity candidate, we use its neighbors together with the inter-relations in the KG as its representation. This is expected to capture richer information of the entity and its different usage. For example, representing "Meg Griffin" with relational information such as "present in work: Family Guy" would be useful for the EL task for the question "who originally voiced meg on family guy?". We thus propose to represent an entity with a list of KG relational triples, each of which consists of the entity (i.e., the *subject*), one of its neighbors (i.e., the *object*), and the relation between them (i.e. the *predicate*).

To obtain the semantic representation of a relational triple such as <"Meg Griffin", "present in work", "Family Guy">, we simply concatenate the words in the triple to form a sentence and then obtain its BERT embedding. Assuming for an entity we have o relational triples in total, we can compute o similarity matrices, between the text and each relational triple as follows:

$$S^{rk}_{ij} = cosine(V^t_{i,:}, V^{rk}_{j,:}) \quad k = 1, 2, ..., o$$

where V^{rk} is the semantic representations for the k-th relational triple.

In this work, we sample a fixed-size of triples for each entity candidate, as opposed to using all of its relational triples since some entities may have thousands of neighbors. We propose a context-aware KG sampling algorithm to sample M triples from the KG for an entity candidate. The algorithm focuses on first sampling those triples whose subject and object entities are both mentioned in the original text. The pseudocode of this algorithm is shown as follows.

```
Data: graph, subj, context_candidates, M
Result: a list of relation triples
triples = [] ;
for entity in context_candidates do
    if entity is connected with subj via relation rel then
        triples.append([subj, rel, entity]);
    end
end
if len(triples) > M then
    triples = sample(triples, M);
else
    relations = sample(graph[subj], $M - len(triples)$);
    for rel in relations do
        obj = sample(graph[subj][rel], 1) ;
        triples.append([subj, rel, obj];
    end
    if len(triples) < M then
        remaining = unselected [subj, rel, obj] triples ;
        triples.append(sample(remaining,
          M-len(triples));
    end
end
return triples
```

Algorithm 1: Context-aware KG Neighbor Sampling

2.1.4. (D) Overall Similarity

This is performed to compute the overall similarity between the text and each entity candidate. Specifically, for the character-level name and description based similarity matrices, we use row-wise and column-wise max-pooling to get the most representative features:

$$z^{cr}_i = max_j(S^c_{ij})$$
$$z^{cc}_j = max_i(S^c_{ij})$$
$$z^{wr}_i = max_j(S^w_{ij})$$
$$z^{wc}_i = max_i(S^w_{ij})$$

For the relation-level similarity matrices, we apply row-wise max-pooling and row-wise mean-pooling to get the features:

$$z^{rr}_i = max_{jk}(S^{rk}_{ij})$$
$$z^{rr'}_i = \frac{1}{M}\sum_{jk}(S^{rk}_{ij})$$

The final similarity is the concatenation of these aggregated features, $F = [z^{cr}; z^{cc}; z^{wr}; z^{wc}; z^{rr}; z^{rr'}]$.

2.2. Entity Ranking: Ranking Model

We use a multi-layer perception (MLP) model to compute the final similarity score as:

$$score = \sigma(b_2 + W_2 \cdot ReLU(W_1 \cdot F + b_1))$$

where W_1, W_2, b_1, b_2 are the matrix parameters and bias terms. We use $ReLU$ as the activation function in the first MLP layer and sigmoid in the second layer as the final output.

To train our model, we minimize a ranking loss, which maximizes a margin between a relevant entity e^+ and an irrelevant entity e^- randomly sampled from the entity candidate set,

$$loss = max(\alpha + score(t, e^-) - score(t, e^+),\ 0)$$

where α is a constant margin (we use 0.5).

Because our model includes a deep language model (BERT) and shallow modules (CNN, MLP), in practice it is hard to train both parts at the same time. We propose an iterative update approach with mixed learning rates to train our model. We first train the shallow parts with a large learning rate r_l for E epochs. Afterwards, we fine-tune the deep language model for one epoch with a small learning rate r_s, then update the shallow parts with learning rate r_l. This fine-tuning procedure is repeated for additional 5 epochs.

3. Experiments

3.1. Datasets and Experimental Setup

We use one public dataset [15], which is compiled from the question answering dataset WebQSP [25]. All the entities mentioned in the question are extracted and linked to Wikidata by [15]. Each question has a main entity that is essential to find the answer. The mentions of the main entities are annotated in the dataset. For other non-main entities, we extract their mentions by matching their entity labels with the text. Table 1 shows the information of the data set.

	# Questions	# Entities
WebQSP Train	3098	3794
WebQSP Test	1639	2002

Table 1: *Dataset Statistics*

The detailed parameters and setups of our model are shown in Appendix. We compare our method with four baselines:

Heuristics is a simple baseline. After the Entity Candidate Search step, it ranks the entity candidates by their frequencies on Wikipedia.

DBpedia Spotlight is also adopted as a baseline for EL using the DBpedia online endpoint[1] [15].

S-MART is an EL system designed specifically for short text. It is a tree-based structured learning framework [10]. It was first trained on the NEEL 2014 Twitter dataset and later adapted to this QA dataset [26].

VCG combines a comprehensive set of features for EL, including entity characters, description tokens, and KG embeddings [15], and proposes a context-aware neural network to aggregate all these features.

For our method, in entity mention detection step, we fine-tune the BERT base model for 3 epochs with a learning rate 10^{-5}. In the entity candidate search step, we set the number of returned results N to be 50 for each search described in Section 2. This results in 137.8 entity candidates on average. For our ranking model, we set the dimension of character embeddings to be 50, and use 100 convolutional filters of width 3, 4, and 5. In the final MLP, the dimension of the hidden layer is 30. We set the KG sampling size M to be 5. Margin α is set to be 0.5 in the loss function. During training, we first train the shallow parts (i.e., CNN and MLP) for 3 epochs with a learning rate of 10^{-3}. Afterwards, we fine-tune the BERT module with a learning rate of 10^{-5} for one epoch and then update the shallow parts with a learning rate of 10^{-3} for another one epoch. We repeat this fine-tuning for 5 iterations.

3.2. Results

Following the previous work [15], we use precision, recall, and F1 score to measure the model performance. Same as [27, 10], we define the evaluation metrics on a per-entity basis. An extracted entity is considered correct if it is present in the set of gold entities. For the main entity, we also consider the entity boundary, that is, an extracted main entity is correct if its detected mention boundary overlaps with the correct one and the entity IDs are the same.

Table 2 shows the results of the F1 scores. The recall and precision values have similar trends, and are not presented here due to space limit. Our proposed method achieves the best F1 score, outperforming all of the baselines. Compared to the previous state-of-the-art method VCG, we achieve an absolute F1 improvement of 5% for all the entities, and more than 7% for the main entity. We also perform an ablation study to show the contribution of each feature. Results in Table 3 suggest that the relation information is the most significant contributor.

	Main entity	All entities
Heuristic	0.401	0.404
DBPedia Spotlight	0.629	0.595
S-MART	0.744	0.715
VCG	0.780	0.730
Ours	**0.851**	**0.780**

Table 2: *EL results of our method and baselines on the WebQSP dataset.*

	Main entity	All entities
label	0.431	0.370
description	0.616	0.534
relation	0.822	0.756
description+relation	0.829	0.712
label+relation	0.832	0.747
label+description	0.829	0.743
Full	**0.851**	**0.780**

Table 3: *EL results using different features.*

We use utterance “where did andy murray started playing tennis?” as an example to show the semantic similarity obtained from the BERT model. The two top entity candidates for mention “andy murray” are:

- Andy Murray, description “British tennis player”
- Andy Murray, description “Canadian ice hockey coach”

Figure 1 is a visualization of the heat map showing the token-level similarity matrix between the text and the entity description.

[1] http://www. dbpedia-spotlight.org/api

As shown in this figure, our model successfully captures the semantic similarity between the context words and the correct entity's description.

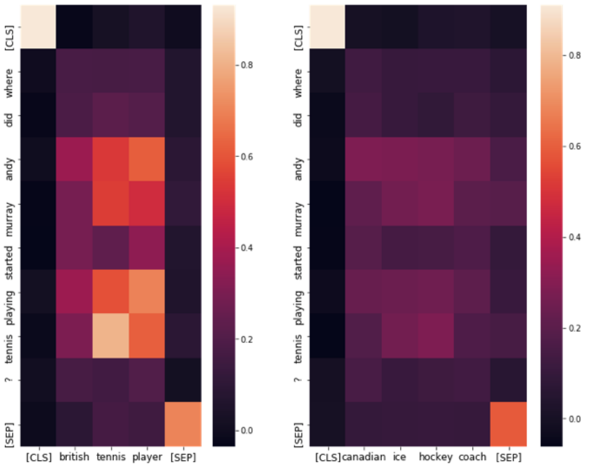

Figure 1: *Visualizations for the token-level similarity matrices between the text and entity description. Left figure is a heat map of the token-level similarity matrix between text "where did andy murray started playing tennis?" and the description of the correct entity "British tennis player". Right figure shows the similarity matrix between the same text with the description of an incorrect entity "Canadian ice hockey coach".*

3.3. Error Analysis

We found the first type of errors is from mention extraction. For instance, in 'what county is st paul va in?", the correct entity is a town called St. Paul in Virginia. However, our system extracts "st paul va" as one entity mention and links it to the city St. Paul of Oregon. This error may be avoided if we can detect "st paul" and "va" as two entity mentions, and then leverage the relational information to correctly link the first entity mention. Another type of errors comes from entity candidate search. Our system has a recall rate of about 95% for the search component. Take "who developed the tcp ip reference model?" as an example, our system could not retrieve the expected entity "Transmission Control Protocol" for mention "tcp ip" because we have not incorporated any alias mapping in the candidate search step. The third error type can be attributed to limited context to disambiguate entities. For example, "where george lopes was born".

4. Conclusion

We present a novel EL system based on multiple level text semantic matching, where we utilize a pre-trained language model to encode the entities with their names, descriptions and other related entities in the KG. We significantly improve over previous methods on one public dataset, and our analyses suggest the relation information has contributed the most to capturing the semantic similarity between the mention context and entities. Our proposed method is especially useful for a new domain where there is no large volume of text data to jointly train the word and KG embedding since we apply the pre-trained language model to the KG. For future work, we would like to adapt the language model to the KG as well as the task domain. In addition, we plan to perform sentence level disambiguation, rather than for each mention separately, when a sentence has multiple entities.

5. References

[1] D. Lukovnikov, A. Fischer, J. Lehmann, and S. Auer, "Neural network-based question answering over knowledge graphs on word and character level," in *Proceedings of the 26th international conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 1211–1220.

[2] H. Zhou, T. Young, M. Huang, H. Zhao, J. Xu, and X. Zhu, "Commonsense knowledge aware conversation generation with graph attention." in *IJCAI*, 2018, pp. 4623–4629.

[3] A. Ram, R. Prasad, C. Khatri, A. Venkatesh, R. Gabriel, Q. Liu, J. Nunn, B. Hedayatnia, M. Cheng, A. Nagar *et al.*, "Conversational ai: The science behind the alexa prize," *arXiv preprint arXiv:1801.03604*, 2018.

[4] W. Shen, J. Wang, and J. Han, "Entity linking with a knowledge base: Issues, techniques, and solutions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 2, pp. 443–460, 2014.

[5] H. Ji, J. Nothman, B. Hachey, and R. Florian, "Overview of tac-kbp2015 tri-lingual entity discovery and linking." in *TAC*, 2015.

[6] H. Wang, J. G. Zheng, X. Ma, P. Fox, and H. Ji, "Language and domain independent entity linking with quantified collective validation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 695–704.

[7] 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 *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 782–792.

[8] L. Bentivogli, P. Forner, C. Giuliano, A. Marchetti, E. Pianta, and K. Tymoshenko, "Extending english ace 2005 corpus annotation with ground-truth links to wikipedia," in *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, 2010, pp. 19–27.

[9] P. Ferragina and U. Scaiella, "Tagme: on-the-fly annotation of short text fragments (by wikipedia entities)," in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 1625–1628.

[10] Y. Yang and M.-W. Chang, "S-MART: Novel tree-based structured learning algorithms applied to tweet entity linking," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 504–513. [Online]. Available: https://www.aclweb.org/anthology/P15-1049

[11] A. Sakor, I. O. Mulang, K. Singh, S. Shekarpour, M. E. Vidal, J. Lehmann, and S. Auer, "Old is gold: linguistic driven approach for entity and relation linking of short text," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 2336–2346.

[12] N. Gupta, S. Singh, and D. Roth, "Entity linking via joint encoding of types, descriptions, and context," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2681–2690.

[13] L. Logeswaran, M.-W. Chang, K. Lee, K. Toutanova, J. Devlin, and H. Lee, "Zero-shot entity linking by reading entity descriptions," *arXiv preprint arXiv:1906.07348*, 2019.

[14] S. Murty, P. Verga, L. Vilnis, I. Radovanovic, and A. McCallum, "Hierarchical losses and new resources for fine-grained entity typing and linking," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 97–109.

[15] D. Sorokin and I. Gurevych, "Mixing context granularities for improved entity linking on question answering data across entity categories," *arXiv preprint arXiv:1804.08460*, 2018.

[16] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” in *Advances in neural information processing systems*, 2013, pp. 2787–2795.

[17] W. Fang, J. Zhang, D. Wang, Z. Chen, and M. Li, “Entity disambiguation by knowledge and text jointly embedding,” in *Proceedings of the 20th SIGNLL conference on computational natural language learning*, 2016, pp. 260–269.

[18] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, “Joint learning of the embedding of words and entities for named entity disambiguation,” *arXiv preprint arXiv:1601.01343*, 2016.

[19] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Lingvisticae Investigationes*, vol. 30, no. 1, pp. 3–26, 2007.

[20] L. A. Ramshaw and M. P. Marcus, “Text chunking using transformation-based learning,” in *Natural language processing using very large corpora*. Springer, 1999, pp. 157–176.

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

[22] C. Gormley and Z. Tong, *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. ” O’Reilly Media, Inc.”, 2015.

[23] L. Pang, Y. Lan, J. Guo, J. Xu, S. Wan, and X. Cheng, “Text matching as image recognition,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[24] Y. Qu, J. Liu, L. Kang, Q. Shi, and D. Ye, “Question answering over freebase via attentive rnn with similarity matrix based cnn,” *arXiv preprint arXiv:1804.03317*, vol. 38, 2018.

[25] W.-t. Yih, M. Richardson, C. Meek, M.-W. Chang, and J. Suh, “The value of semantic parse labeling for knowledge base question answering,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2016, pp. 201–206.

[26] S. W.-t. Yih, M.-W. Chang, X. He, and J. Gao, “Semantic parsing via staged query graph generation: Question answering with knowledge base,” 2015.

[27] D. Carmel, M.-W. Chang, E. Gabrilovich, B.-J. P. Hsu, and K. Wang, “Erd’14: entity recognition and disambiguation challenge,” in *Acm Sigir Forum*, vol. 48, no. 2. Acm, 2014, pp. 63–77.