Collective Entity Linking Models via Graph Neural Network

Yizhou Zhang
zhangyiz@usc.edu

Abstract

Collective Entity linking maps mentions of entities in document to corresponding entries in a Knowledge Base (KB) under the constraint which keep the coherence of mention relations and entity relations. This course project aims at handling collective entity linking via Graph Neural Network (GNN). Specifically, mentions in the same document are parsed as a mention graph and the sub-graph consisting of their candidate entities are extracted from the whole Knowledge Base. Then, on the mention and entity graphs, two graph neural networks update the node representations and the matching scores in an iterative manner. In this way, the matching scores and node representations can improve each other continuously, so that a better mapping can be obtained. Experiments shows that the proposed model outperforms the existing graph neural network based methods and has performance close to the state of the art method based on reinforcement learning.

1 Introduction

Entity linking, which aims at mapping the mention of entities in corpus to corresponding entries in a given knowledge base, supports many NLP tasks like question answering(Sorokin and Gurevych, 2018). A typical paradigm of handling this task is generating the candidate entity set based on some rule based method like string matching, then apply disambiguation method to select the correct entity from candidate set. The key challenge in this process is how to find referring entity from multiple entities with similar (even exactly same) surface name given the context of entity mention and the information from knowledge base.

A straight froward method of handling disambiguation is to directly compare the similarity of a mention and a candidate entity based on some extra information besides surface name, known as local entity linking. Previous works have proposed a lot of methods for such similarity comparison, including manually designed features and representation learning. The manually designed features are usually some statistic feature and rule based similarity score like edit distance, suffix and prefix relation, Jaro-Winkler distance and cosine similarity of word frequency vector(Alhelbawy and Gaizauskas, 2014; Pershina et al., 2015). The representation learning methods usually apply neural networks to encode the information from mention context and knowledge base to embedding vectors, then calculate the similarity based on them (Francis-Landau et al., 2016; Cetoli et al., 2018).

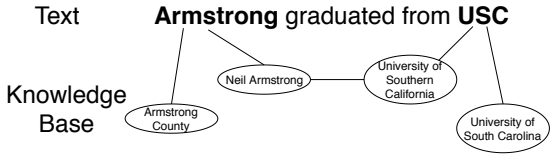

Figure 1: A toy example illustrating global coherence. The text contain two mentions: Armstrong and USC. The mention 'Armstrong' has two candidate entities: Armstrong County (a place) and Neil Armstrong. The mention 'USC' has to candidates: University of Southern California and University of South Carolina.

Existing local linking methods performs well when handling the disambiguation of a single mention. However, sometimes there are multiple correlated mentions referring to different entities in one document or sentence. In this scenario, a good mapping from mention to entities should preserve the global coherence between mention relation in the document and entity relation in the knowledge base. Figure 1 illustrate global coherence with a toy example. To address this issue, researchers develop different methods to find global coherent mappings. One typical idea is to con-

struct a candidat entity graph based on entity relation, then solve the entity linking as a graph ranking problem via graph ranking methods (Alhelbawy and Gaizauskas, 2014; Pershina et al., 2015) or graph neural network. (Cao et al., 2018).

However, the graph ranking based method only explicitly model the relation of candidate entities and their neighbors. The relations of mention are only partly implicitly encoded in the mention context feature via some manually designed rule instead of machine learning method. This course project proposes a new graph based method of capturing global coherence. This method construct a mention graph from the document and an entity graph from the knowledge base, then solve entity linking as a graph matching problem. An iterative graph attention network continuously update: (1) the matching score of each mention-candidate pair; (2) the representation of mention and entity representation and (3) the strength of each edge in both mention graph and entity graph. The matching score output by the last iteration is then combined with some other local features together to be the final feature containing both global and local information. A neural network select the best candidate based on the combined feature. In this way, we explicitly model the relation in both mention and entity graphs and make use of local and global information together. Experiment results show that this model outperform other graph-based method and get accuracy close to the state of the art method based on sequential decision process learningg.

In general, the contributions of this course project are:

- I formulate the task of global entity linking as graph matching problem on mention graph and entity graph.
- I propose an graph attention neural network to iteratively find the matching from mentions to entities. This method is able to overcome the drawback of traditional graph-ranking based method.
- I conducted experiments on entity linking datasets and compare the proposed method with some baselines. I also compare the different variant of proposed methods to analyze the influence of model designing and parameters.

2 Related Work

Entity linking aims at mapping mentions of entities in document to their corresponding entries in a Knowledge Base (KB). Existing methods can be divided into two classes: local methods and global methods.

2.1 Local Entity Linking

Local methods directly measure the similarities of a mention and its candidate entities given mention context and entity knowledge from KB based on manually designed features and/or representation learning. Manually designed features have two popular branches: (1) string similarity like edit distance, suffix and prefix relation, Jaro-Winkler distance and cosine similarity of word frequency vector (Pershina et al., 2015; Alhelbawy and Gaizauskas, 2014) (2) and statistic features like the Freebase Popularity Score which measures how popular an entity is in Freebase and Wikipedia (Nebhi, 2013). While representation learning methods usually apply neural network like Convolutional Neural Network (CNN) and Transformer to encode mention context, entity description (Francis-Landau et al., 2016; Ganea and Hofmann, 2017).

2.2 Global Entity Linking

Different from local methods, global methods try to find mention-entity mappings that not only consider the local similarity but also preserve the coherence of mention relations and entity relations. To this end, researchers develop different methods:

2.2.1 Graph ranking method

A typical idea of capturing global coherence is to construct a candidate entity graph, where each node is a candidate entity and each edge represent the relation between two entity in the knowledge base. Then, some ranking algorithm are applied to calculate a score for each node as the coherence feature. In (Alhelbawy and Gaizauskas, 2014) PageRank algorithm is applied to measure the coherence of one entity to all possible mappings. In (Pershina et al., 2015), Personalize PageRank algorithm is applied to calculate the coherence of one entity to each single single candidate of other mentions based on both local similarity and candidate entity graph structure. In (Cao et al., 2018), a learnable neural method based on graph neural

network is proposed to learn a supervised model that predict the golden candidate directly.

2.3 Sequential decision process learning

Different from above graph based methods, DCA (Yang et al., 2019) treats entity linking as a sequential decision process. In each decision step, the model decides to link an entity to current mention m_{t+1} based on the embedding of all candidate entities of m_{t+1} and the contextual representation from entities linked in previous steps. The model can be trained in two manners: supervised learning manner which trains the model to make decisions with gold linked entities and reinforcement learning manner which asks the model to make decisions based on its own previous decisions and only provides reward signals instead of golden truth.

3 Task Definition and Challenge

3.1 Entity linking

Let D denote a document containing a set of mentions $M = \{m_1, m_2, \ldots, m_{|M|}\}$. Given a knowledge base K containing a set of entities $E = \{e_1, e_2, \ldots, e_{|E|}\}$ and a link set L containing their relations, entity linking aims at find a mapping $f : M \rightarrow E$ that links mentions to the referring entities. As it is unrealistic to use machine learning model to calculate the similarity scores of all mention-entity pairs in $M \times E$, people usually first use some heuristic algorithm to filter out most of pairs and only generate a candidate entity set for each mention, then calculate the matching scores of mention-candidate pairs and link each mention to the candidate with highest score. We focus on the latter step known as candidate entity ranking and assume a set $C = \{c(m_1), c(m_2), \ldots, c(m_{|M|})\}$ where $c(m_i)$ is the candidate set of mention m_i has already been obtained.

To get a good mapping, the entity linking model should not only consider the similarity of each mention-entity pair independently, but also preserve **global coherence**.

3.2 Global coherence and challenge

Global coherence means that the relations between mentions in a corpus should be consistent to the relations among corresponding entities. Figure 1 is a toy example of global coherence. In this example, as the two mentions in text are correlated, their golden entities are very likely to be correlated in the knowledge base. Therefore, the 'Armstrong' should be linked to 'Neil Armstrong' and the 'USC' should be linked to 'University of Southern California'.

A straightforward method of preserving global coherence is to add a 'global coherence score' term into the general matching score. However, precisely calculating the global score is confronted of a challenge brought by its dependency to golden entities. For example, in the toy example of Figure 1, when calculating the global coherence score of a candidate entity of 'Armstrong', the model need to know whether it is correlated to the golden entity of 'USC', whose global score will be determined by the model after the golden entity of 'Armstrong' is decided. Consequently, the calculation of global coherence score becomes a 'chicken-and-egg' problem. Although this challenge vanishes if we design a model that directly enumerate all possible candidate entity combination and calculate the matching scores between the combinations and mention set, such model will not be practical as it has a searching space with complexity $O(\overline{c}^{|N|})$, where $\overline{c}$ is the average size of candidate entity sets and $|N|$ is the number of mentions.

4 Proposed Method

Although precisely calculate the global score for a mention correlated with multiple mentions without golden entity is very challenging, it is possible to estimate the global score of a candidate entity via iterative algorithms. Previous works propose to use PageRank algorithm and its variant to calculate a global score as the estimation of precise global coherence. However, having no trainable parameters, they can not learn knowledge from data by themselves. This course project propose a graph neural network model that update the matching score in an iterative manner so that the estimation can be improved continuously.

The overview of proposed model is shown in Figure 2. In the initial stage, the a mention graph is constructed from corpus and an entity graph is extracted from knowledge base. Through a pre-trained local entity linking model, each mention m_i is represented as a local feature vector $f(m_i)$ and each entity e_j get an embedding vector $v(e_j)$. After that, initial normalized matching scores of mention-candidate pairs are calculated by the pre-

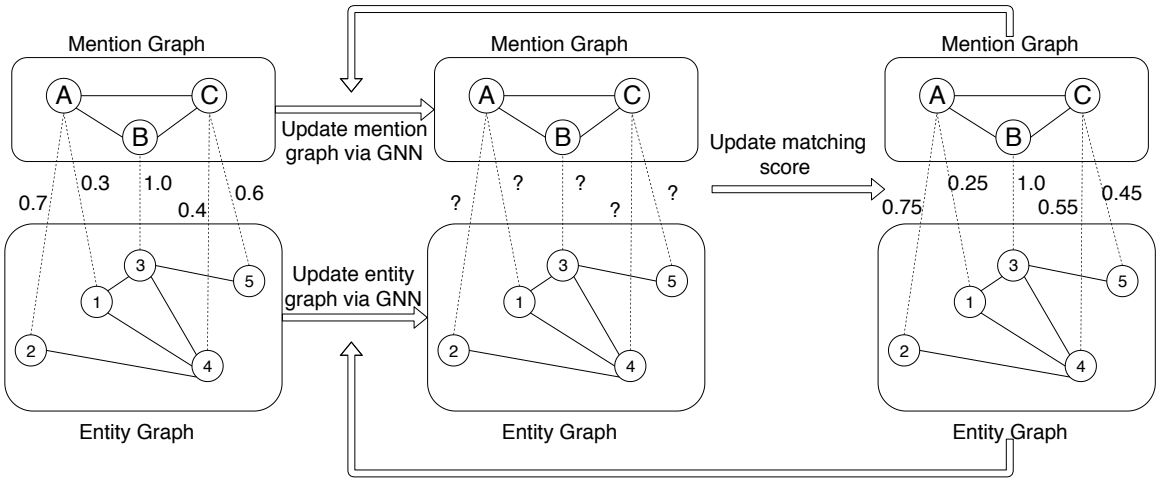

Figure 2: The overview of proposed model. A pre-trained model represents mentions and entities as a local feature vector. And initial matching scores of all mention-candidate pairs are calculated based on initial features. Then, the representations and matching scores are updated in an iterative manner. In each iteration the representations are updated based on current matching scores, and then the matching scores are updated based on new representations.

trained model. Then the model start an iterative process. Each iteration contains two stages: representation updating and matching score updating.

4.1 Representation updating

In each iteration, mention representation and entity representations are updated via two graph neural networks respectively. Denoting $h^l_{e_i}$ as the hidden representation of entity e_i in l-th layer, the entity representations are calculated via a graph attention aggregation layer similar as GAT(Veličković et al., 2018):

$$h^{l+1}_{e_i} = v(e_j)^l \oplus \sum_{e_j \in \mathcal{N}(e_i)} a^{l+1}_j \cdot v(e_j) \quad (1)$$

where $\mathcal{N}(e_i)$ is the neighbor set of e_i, W_e is the parameter in this layer, a^{l+1}_j is the weight of neighbor e_j in $l+1$-th layer and $\oplus$ is concatenate operator. This operation aggregate the information of a candidate entity and its neighbors. To dynamically model the importance of different neighbors, we calculate a^{l+1}_j as follows:

$$a^{l+1}_j = softmax(h^l_{m_k} \cdot v(e_j)) \quad (2)$$

where m_k is the mention whose candidate entity is e_i and $h^l_{m_k}$ is its representation in l-th layer. The softmax operation is done among all neighbors of one candidate entity. Here, we calculate the importance based on the representation of mention instead of the candidate entity. This is because a neighbor entity related to the mention and its context is more important than a neighbor with strong correlation with the candidate.

At the same time, the mention graph is updated by another graph attention layer.

$$\begin{gathered} h^{l+1}_{\mathcal{N}(m_i)} = Atten(\{\hat{h}^l_{mj} | m_j \in \mathcal{N}(m_i) \cup \{m_i\}\}) \\ h^{l+1}_{m_i} = f(m_i) \oplus h^{l+1}_{\mathcal{N}(m_i)} \end{gathered} \quad (3)$$

where $Atten$ is a graph attention layer from (Veličković et al., 2018), which calculate the aggregation feature of of neighbors and the central mention based on the summation weight from attention mechanism. The local feature $f(m_i)$ is concatenated to the representation from attention layer to enhance the representation.

4.2 Matching score updating

In this step, the normalized matching scores are calculated based on mention and entity representations representations. The matching strength in l layer is calculated via a bi-linear model:

$$\phi^l(m_i, e_j) = (h^l_{m_i})^{\mathrm{T}} \cdot A^l \cdot h^l_{e_j} \quad (4)$$

where A^l is the parameter of bi-linear model in layer l. As we hope the matching score of a mention-candidate pair can represent the probability that it is the golden pair, so that we can use the score to estimate the golden entity representation. So, we use a softmax function to normalize the matching strength:

$$\Phi^l(m_i, e_j) = \frac{\exp\left(\phi^l(m_i, e_j)\right)}{\sum_{e_k \in c(m_i)} \exp\left(\phi^l(m_i, e_k)\right)} \quad (5)$$

Method	AIDA-B	ACE2004	AQUAINT	MSNBC	CWEB	WIKI
PPRforNED	91.82	x	x	x	x	x
NCEL	80	88	87	x	x	x
Pure Feature Extractor	90.88	86.92	84.6	91.97	70.07	74.37
DCA	94.64	90.14	88.25	93.8	75.59	78.84
Our Method	92.4	88.5	87.0	93.8	73.1	76.3

Table 1: The current result of baseline performance.

4.3 Entity Disambiguation

Following (Yang et al., 2019), we treat the matching score of mention-entity pair (m_i, e_j) output by the last iteration as global coherence feature, denoted as $Coh(m_i, e_j)$. Then we concatenate it together with the three local similarity scores: (1) Mention-entity Prior $\hat{P}(m_i, e_j)$, which is the prior possibility that mention m_i refers to entity e_j a estimated from Wikipedia; (2) Context Similarity which is the local similarity score calculated based on the output of the local feature extractor and (3) Type similarity, which represent the similarity of the entity's type (person, orgnization .....) and the possible type of the mention inferred by a typing system (Xu and Barbosa, 2018). Then we use a multi-layer perceptron (MLP) to predict the the golden entity based on the concatenation of local and global feature. Cross entropy loss function is applied to train the model end-to-end.

5 Experiment Design and Current Result

5.1 Datasets and Experiment Setting

Following existing works, we use Wikipedia as knowledge base. The datasets in this course project include:

- **AIDA** contains 1393 documents and 27724 mentions. It contain three subsets: AIDA-train with 946 documents, AIDA-A with 216 documents and AIDA-B with 231 documents.
- **AQUAINT** contains 50 documents and 727 mentions.
- **ACE2004** contains 36 documents and 257 mentions.
- **MSNBC** contains 20 documents and 656 mentions.
- **CWEB** contains 320 documents and 11154 mentions.
- **WIKI** contains 320 documents and 6821 mentions.

5.2 Baseline for comparison

Three global models that represent different types of entity linking will be evaluated as baselines:

- **PPRforNED**(Pershina et al., 2015) applies personalized PageRank algorithm to calculate the global score of an entity. This model has no trainable parameters and represent the type of entity linking algorithms that do not apply machine learning.
- **NCEL**(Cao et al., 2018) applies graph convolutional networks(Kipf and Welling, 2017). It represent graph based global entity linking models. This model does not make use pretrained feature extractor. Instead, it use some rule based score to transform word embedding to features, then concatenate the features to the entity's knowledge embedding as node feature.
- **DCA**(Yang et al., 2019) considers entity linking as a sequence decision problem and preserve the global coherence to linked mentions. In this way, the computational complexity is reduced. This model has two version: supervised learning version which has the best In-domain performance and reinforcement learning version which has the best Cross-domain performance. As in our experiments, most dataset are cross-domain, we select the result of DCA of reinforcement learning version.

As the first method require Freebase Popularity Score from a API whose access has been closed by Google as an important feature, we can only compare our model with it on the AIDA-B dataset. Meanwhile, as the author of the second method only provide the embedding they used in on ADIA-B, AQUAINT and ACE2004 dataset,

Variant	AIDA-B	ACE2004	AQUAINT	MSNBC	CWEB	WIKI
Variant 1 (No Neighbor Entity)	92.5	89.3	87.2	93.0	72.5	76.1
Variant 2 (No Attention Layer)	60.3	80.1	84.8	67.2	65.7	61.0
Our Method	92.4	88.9	87.0	93.8	73.1	76.3

Table 2: The ablation test result of removing neighbors of candidate entities and replacing graph attention layer with graph convolution layer.

Iteration Number	AIDA-B	ACE2004	AQUAINT	MSNBC	CWEB	WIKI
2	92.4	88.1	86.0	93.2	73.5	76.0
3	91.9	88.9	86.6	94.0	73.3	76.1
4	92.4	88.9	87.0	93.8	73.1	76.3
5	92.0	87.7	87.0	94.4	73.4	76.0

Table 3: The parameter analysis of iteration number.

we only compare our model with NCEL on these three datasets.

5.3 Hyper-parameter and Experiment Setting

For the hyper parameter of our method, we set the hidden dimension as 1024 and iteration number as 4. I construct the mention graph as a fully connected graph where all mentions within the same document are connected to each other. I select the attention based feature extractor in (Ganea and Hofmann, 2017) as the pre-trained local feature extractor for both of our method and DCA. As NCEL does not make use pre-trained feature extractor, we just follow the original setting in the paper to apply their own word embedding.

As the size of some public datasets in entity linking task are relatively small for training deep neural networks, following previous works, our model and DCA will be trained on AIDA-train and AIDA-A will be used as validation set. Then, the model will be evaluated on AIDA-B and other datasets. For NCEL, we use the parameter trained on a dataset constructed by the authors for training.

5.4 Comparison with Baselines

The current result is shown in Table 1. Besides the baselines of collective entity linking, the results of pure feature extractor are also included. From the results, we can have following conclusion:

- Our method outperform the pure feature extractor. This phenomenon indicate that our method is able to make use of global coherence to give better prediction than pure local feature.
- Our method outperform the both of the PageRank based and Neural Network based graph ranking based methods. This phenomenon indicate that treating entity linking as graph matching method and solving it via graph neural network is a better paradigm for graph based methods.
- Our model does not outperform the DCA method on any dataset. It shows that our model still require improvement. Improve the architecture of the graph attention layer might be a possible solution.

5.5 Ablation Test and Parameter Analysis

5.5.1 Ablation Test

In this section, we compare our model with two variants. The first variant (Variant 1) removes the graph neural network on entity graph and can not make use of the information from the neighbor of candidate entities. On the first three datasets (AIDA-B, ACE2004 and AQUAINT), our model's performance is lower than Variant 1, but very close to its performance (0.23% on average). But on the other three datasets (MSNBC, CWEB and WIKI), our model outperform the Variant 1 by 0.53% on average. This phenomenon shows that the neighbor information of candidate entities do have a little contribution to the model.

The second variant (Variant 2) replace the graph attention layer with simple graph convolutional layer. As we can see, this change causes the performance to drop seriously. Its performance is even not better than the pure feature extractor. This phenomenon indicate the importance of using attention mechanism to dynamically infer the

strength of different relations in mention and entity graph. The simple graph convolution operation brings noise rather than information when the mention graph is constructed as a fully connected graph.

5.5.2 Parameter Analysis of Iteration Number

The experiments of trying different Iteration Numbers is conducted to analyze the influence of this hyper parameter. The results are shown in 3. The performance of proposed model changes very little, indicating the robustness of our model to the iteration number. I select 4 as the hyper parameter in the previous experiments because on average this choice brings the best performance.

6 Future Work

The experiment results indicate both of the advantage and disadvantage of the proposed model based on iterative graph attention network. In the future, I plan to make more experiments like observing the attention weights to find the method of improve the model's performance further so that it can achieve state of the art and can be submitted to conference of NLP.

References

Ayman Alhelbawy and Robert Gaizauskas. 2014. Graph ranking for collective named entity disambiguation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 75–80, Baltimore, Maryland.

Yixin Cao, Lei Hou, Juanzi Li, and Zhiyuan Liu. 2018. Neural collective entity linking. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 675–686.

Alberto Cetoli, Mohammad Akbari, Stefano Bragaglia, Andrew D. O'Harney, and Marc Sloan. 2018. Named entity disambiguation using deep learning on graphs. *CoRR*, abs/1810.09164.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1256–1261.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. *CoRR*, abs/1704.04920.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Kamel Nebhi. 2013. Named entity disambiguation using freebase and syntactic parsing. In *LD4IE@ISWC*.

Maria Pershina, Yifan He, and Ralph Grishman. 2015. Personalized page rank for named entity disambiguation. In *NAACL HLT 2015 - 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pages 238–243.

Daniil Sorokin and Iryna Gurevych. 2018. Modeling semantics with gated graph neural networks for knowledge base question answering. In *COLING*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations*. Accepted as poster.

Peng Xu and Denilson Barbosa. 2018. Neural fine-grained entity type classification with hierarchy-aware loss. *CoRR*, abs/1803.03378.

Xiyuan Yang, Xiaotao Gu, Sheng Lin, Siliang Tang, Yueting Zhuang, Fei Wu, Zhigang Chen, Guoping Hu, and Xiang Ren. 2019. Learning dynamic context augmentation for global entity linking. In *Proceedings of EMNLP-IJCNLP*.