A Survey on Entity Linking Models based on Local Similarity and Global Coherence

Yizhou Zhang
zhangyiz@usc.edu

Abstract

Entity linking aims at mapping mentions of entities in document to their corresponding entries in a Knowledge Base (KB). A straightforward challenges of this task is how to measure the similarities of a mention and its candidate entities. Furthermore, when a document contains multiple correlative mentions, another challenge, which is to construct a combination of mapping entities for these mentions with not only high similarity scores but also good global coherence, comes behind. This survey will introduce some methods of extracting local features that is helpful for similarity calculation and finding global coherent mappings.

1 Local Feature Extraction

1.1 Manual features

Manual features are designed based on people's intuition about entity linking. A very straightforward similarity metric is string similarity. In (Pershina et al., 2015), edit distance, suffix and prefix relation are applied to measure the string similarity, while in (Alhelbawy and Gaizauskas, 2014) Jaro-Winkler distance and cosine similarity of word frequency vector are applied to measure it. Furthermore, to capture the information from the context, some contextual features are also proposed, such as the similarity between the sentence containing the mention and the entity description in Knowledge Base (Alhelbawy and Gaizauskas, 2014).

Besides, some manual features are also designed based on the entities' structural property in the Knowledge Base. In (Nebhi, 2013), the Freebase Popularity Score which measures how popular an entity is in Freebase and Wikipedia, is proposed to be an important feature for entity linking. This feature is then applied in (Alhelbawy and Gaizauskas, 2014) and (Pershina et al., 2015).

1.2 Representation from Neural Networks

To calculate the similarity in a data-driven paradigm, some researchers propose to use neural networks to extract the local features of mentions and entities. In this survey, we introduce two related papers.

The first paper (Francis-Landau et al., 2016) proposes to apply convolutional neural network (CNN) to learn representations for both mentions and entities. Specifically, each mention is expressed as its surface form denoted as $ment$, the sentence containing itself denoted as $context$ and the whole document denoted as doc. And each entity is expressed as the title and document content of its entry in knowledge base, denoted as $title$ and doc respectively. Then, a convolution operation is applied on each corpus to acquire the representation:

$$conv_g(w_{1:n}) = \sum_{j=1}^{(n-l)} max\{0, M_g w_{j:j+l}\} \quad (1)$$

where M_g is the filter parameter of channel g.

Then, for each mention and entity the CNN produces a vector set $\{s_{ment}, s_{context}, s_{doc}\}$ as the mention representation and another vector set $\{t_{title}, t_{doc}\}$ as the entity representation.

The second paper (Cetoli et al., 2018) proposes to encode mentions and entities separately via different neural network architectures. The mention's representation is acquired via a Long Short-Term Memory (LSTM). The LSTM processes the context of the mention and output a representation via a masked summation that only preserves the output of the iterations taking the words in the mention as input. Meanwhile, the entity is expressed as a graph containing itself and its neighbors in the knowledge base. Then a deep neural network on graph processes the graph and outputs a representation for the central entity. The authors tries two basic neural network architectures on graphs including graph convolutional network (GCN) (Kipf and Welling, 2017) and RNN, and their improved versions with additional attention mechanism. Empirical result shows that the RNN with attention has better performance.

2 Global Coherent Mapping

Multiple mentions in a same document are usually correlated. Their relations are coherent to the relations among corresponding entities. Therefore, a good mapping should also preserve the coherence between mention relations and entity relations. To address this issue, researchers develop different methods to find global coherent mappings.

2.1 Graph ranking method

In (Alhelbawy and Gaizauskas, 2014), PageRank algorithm is applied to measure the coherence of one entity to all possible mappings. In this method, a sub-graph containing all candidate entities and their relations is extracted from the knowledge base and Wikipedia documents. Then the PageRank score of each node in the sub-graph is calculated as the coherence score. To obtain a more robust mapping, two different rules are applied to combine local similarity and coherence score (summation and multiplication). And the candidate is selected via the rule that distinguish the top ranking nodes with higher confidence.

As the PageRank based method only calculates the coherence of an entity to all possible mappings rather than each single candidate of other mentions and does not utilize the local similarity during coherence calculation, a Personalized PageRank (PPR) based method (Pershina et al., 2015) is proposed. Unlike PageRank, PPR can calculate the personalized relevance of node e and node s, denoted as $PPR(s \rightarrow e)$. In the PPR based method, the personalized coherence of a mention m's candidate entity e to another entity s is:

$$coh_s(e) = PPR(s \rightarrow e) \cdot Sim_{local}(e, m) \quad (2)$$

where $Sim_{local}(e, m)$ is the local similarity of m and e. To reduce the noise, each mention's candidate set only selects one candidate as the contributor toward the calculation of e's overall coherence. The contributor $Contr(m)$ from the candidate set of mention m' ($m' \neq m$) is:

$$Contr(m) = \underset{s}{\operatorname{argmax}}\ coh_s(e), s \in \Phi(m') \quad (3)$$

where $\Phi(m')$ is the candidate set of m'. Then, the overall coherence of e, denoted as $coh(e)$ is the sum of its coherence scores to all contributors.

2.2 Graph neural network

Graph ranking methods are usually not trainable. To address this issue, graph neural network is applied to find the coherent entities (Cao et al., 2018). In this method, each candidate entity e of a mention m is assigned with a feature vector containing its embedding, string similarity to m, compatibility to the context and neighbor mentions of m. Then, the authors construct a graph whose nodes are all candidate entities of all mentions and edges represent the similarities of all node pairs' embedding vectors. With above graph and node features, a GCN is trained to predict the gold entity given a mention's candidate set.

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. To dynamically decide the relevance of a linked entity $\hat{e}_i$ to m_{t+1}, a bilinear model is applied:

$$u(\hat{e}_i) = \max_{e^j_{t+1} \in \Phi(m_{t+1})} e^j_{t+1} \cdot A \cdot \hat{e}_i \quad (4)$$

where $\Phi(m_{t+1})$ is the candidate set of m_{t+1} and A is a parameterized diagonal matrix. Top K linked entities are left while others are pruned. Then a softmax function normalizes the relevance score to attention weights, based on which the weighted sum of preserved linked entities' embedding is calculated as the contextual representation.

The authors propose two training methods for the decision model: supervised ranking method and reinforcement learning method. The supervised method trains the model to make decisions with gold linked entities. And the reinforcement learning method asks the model to make decisions based on its own previous decisions and only provides a reward signal after all mentions are linked. Empirical result of the paper shows that the reinforcement learning method performs better in Cross-domain scenario in general while the supervised method performs better under In-domain scenario with longer decision length.

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.

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.

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*.