Fine-Grained Named Entity Recognition in Question Answering with DBpedia

Shimin Zhong1, a, *, Yajun Du2, b and ZhenWei Gao3, c
1Xihua University, ChengDu, China
2Xihua University, ChengDu, China
3Xihua University, ChengDu, China
a1126667316@qq.com, bduyajun@mail.xhu.edu.cn, c1274337217@qq.com
*Corresponding author

Abstract. Named Entity Recognition in Question Answer tasks can help the language understanding. Traditional NER task usually solved by a supervised model, which need a large number of annotated corpora and extract context information as feature for training, however questions are usually short sentences, with little context information, little public annotated corpora. On the other hand coarse-grained named entities’ types cannot offer enough information for question understanding. In this paper, we propose a novel model to address both problems, using a distant supervised method. Firstly, we use the web search to obtain more relevant information. Secondly, we present a greedy n-grams algorithm to extract the entity mentions. Finally, we use the kNN to classification and get fine-gained entity types combining with the entity mentions which can link with DBpedia. Experimental results show that our model outperforms various state-of-art systems in public dataset--TREC.

1. Introduction

Entity Recognition play an important role in Natural Language Understanding(NLU), such as Question Answering Understanding [1] [2] [4]. Existing work on Question Answering systems includes rule-based approach, IR -based approach [3], Knowledge base-based approach [6] [7] [8] and Hybrid approach which joints the IR-based approach and Knowledge base-based approach [5] [2]. Knowledge base-based QA system needs to extract entity mention pair and uses distant supervision to select the relation between them based on existing knowledge bases, such as DBpedia [11] Wikipedia.

Previous studies on entity types, mainly focused on special domain and the coarse-grained entity types [12-15], for example person, location, organization. But coarse-grained entity types can only provide us with limited information about the question understanding. Recent studies, fine-gained entity recognition suggests to improvement for question understanding. Most existing work on fine-gained entity recognition includes traditional supervision model HMM, CRF [23], distant supervision model [16-18], deep learning-based model [20-22] [24] and hybrid approach [22]. However, at the QA system, those methods often have some problems. First exiting public available QA corpus almost don’t have fine-grained entity annotation. Second, QA sentences is too short to use context information.

In this paper, we propose a distant supervised fine-grained named entity recognition model to solve these problems. We use the search engine to extend question sentences’ information by using the IR
return sentences. We use the distant supervised method based DBpedia to salve the no annotation corpus’ problem. Main contributions of this paper are concluded as follows:

- We propose a entity detect model to obtain the entity mentions.
- We utilize a distant supervision entity types’ model based DBpedia.

2. Related Work

For named entity recognition, Dakka and Cucerzan [25] classified articles into ACE’s [26] four NE types (PER, ORG, LOC, MISC) using supervised machine learning algorithms similar to SVMs and naive Bayes. Fleischman and Hovy (2002) classifies person entities into 8 subcategories. The AMR [31] defined more than 100 types of entities. Ling et al [28]. generalized 112 entity categories used a distant supervision method based freebase. Sekine (2008) and Lee et al. (2006) separately proposed a tag set of around 150. Higashinaka et al. [29] proposed a supervised machine learning model for classifying Wikipedia articles into the 200 fine-grained NE types. Suzuki et al. [33] proposed a Fine-Grained Named Entity classification with Wikipedia Article Vectors. Huang et al. [18] presented a unsupervised framework based on the Wikipedia by using entity linking tool.

The task of named entity recognition in QA systems, is the key to question understanding. Cui et al.[19] proposed a templates based way to extract entity mentions in QA, and linked KB to deal QA task. Diego et al. [34] performed a step towards such a formal study of the ideal characteristics of a NER for the task of QA. Sun et al. [2] proposed a Hybrid approach, using public entity linking tool to combine candidate entity mention with the Wikipedia. Zou et al. [32] propose to represent a natural language question using a semantic graph query to be matched with a subgraph in KBs and reduce question answering to a subgraph matching problem. However, these methods heavily depend on the scale of knowledge base, the exit knowledge bases have limited the number of entity mentions. There are entities out of the knowledge bases, as the many words out of the vocabulary(OOV). Compare to these effort, in this paper, we want to use the results which returned by web search as the context information to detect entity mentions. Since no fine-gained annotation corpus (TREC dataset), We selected a distant supervised method to classify the entity mentions. For getting some annotated, we link the entity mention with DBpedia.

3. Method

This section will introduce our proposed fine-gained named entity recognition model in detail. Figure1 briefly illustrates the framework of our model which contains the following components in order: In 3.1, we first introduce the entity detect model. In 3.2, we introduce a distant supervision model based DBpedia to obtain the fine-gained entity types. We elaborate the details of each component as follows:

3.1. Entity detect model

The entity detect model contain tree steps: First, selecting related sentence via web search. Second, using greedy n-grams algorithm to detect candidate entity mentions. Third, we utilize a rule to filter the candidate entity mention.

Related sentence via web search: Question sentence is always too short to get enough context information. Similar to Sun [2] method, which uses search engine to collect the top50 relevant sentences that can potentially answer the question, We submit question sentence to search engine and select top10 returned snippets as related sentences.
Detect the candidate entity mentions: By extending the methodology used in [30], we propose a greedy n-grams algorithm to guide the filtering of low-quality phrases and generate candidate entity mentions. In doing so, we can partition sentences in the QA corpus into non-overlapping segments which frequent co-occurrence in related sentences. If the segments’ frequency meet a significance threshold, we regard this segments as candidate entity mention.

Filter the candidate entity mention: Although we use a greedy n-grams algorithm to generate candidate entity mention, there are many problerm, for example: the stop word “how” is frequent appear in related sentences but it isn’t entity mention. As the same, there are many segments which contain more than one word like “the China”, it neither entity mention, just “China” is entity mention.

To solve these problem, we use a rule to filter the candidate entity mention. As follows:
1. Remove the single stop word.
2. Remove the article word from entity mention segments which beging of article.

3.2. Distant supervision model
The distant supervision model contains two steps: First link entity mention with DBpedia to generate a labeled data for training. Second, the KNN model is used to classify the entity mentions and then we get fine-gained named entity types.

Linking entity mentions: We utilize a large entity types’ set based DBpedia which has comprehensive ontology classes. To classify, we need a massive amount of entity mentions which we know the types, like annotation corpus, but there doesn’t exist this data. So we use a distant supervision method to generate these data. DBpedia has a sparql endpoint which can help to query the entity mention information, we use this endpoint link entity mentions which detect from question sentences with DBpedia’s entity mention. If an entity mention which come from question sentences can link with DBpedia, then we extract this entity mention’s types path, for example, ChengLong→Actor→person→Things.

KNN classification model: The KNN (k-nearest neighbors algorithm) is a non-parametric method used for classification and regression, the input consists of the k closest training examples in the feature space, the output is a class membership. We utilize the feature extraction method which proposed by [16], we just use 3 lexical features in our work which is different from [16], include:
- A window of 1 word to the left of Entity mention and right of Entity mention.
- A window of 1 word's POS-tag to the left of Entity mention and the right of Entity mention.
- A window of 2 word's POS-tag to the left of Entity mention and the right of Entity mention.

In order to calculate the distance between entities, the distance D is defined as follows:
\[
D(m_1, m_2) = w_1 \cdot \cos(v_1, v_1) + w_2 \cdot \cos(v_2, v_2) + w_3 \cdot \cos(v_3, v_3), (w_1 + w_2 + w_3 = 1)
\]

4. Experiments
In this section, we present our main experimental results in two datasets. We use evaluation metrics to evaluate the proposed framework. As well as comparison with state-of-the art systems.

4.1. Data Preparation
We do the experiment in public available dataset: TREC dataset. TREC dataset: We selected 5000 questions from the TREC (Text Retrieval Conference) 1999-2007 QA data. The questions which are all factoid question have no annotation, for evaluating our model we also annotated the 5000 questions with reference to DBpedia’s ontology classes.

1. related sentences. We submitted the question to search engine, and return related sentences by web search. We used web crawler to get the related sentences text, and then form a related sentences corpus.
2. classify Feature Generation. To get the words’ feature vector representation, we used a pretrained word2vec model to gain the word’s representation. we applied Stanford POS tagger [10] on related sentence corpus form a POS tagger corpus. We trained a POS-tagger word2vec model on POS tagger corpus. We used POS-tagger word2vec model to gain the feature2 and feature3 representation.
4.2. Evaluation Metrics

We use F1 score computed from Precision and Recall in 2 different ways to evaluate the entity recognition performance. The two ways of computing precision / recall are listed as follows:

- **Entity mention**: The prediction is considered the entity mention’s extract work performers. We denote: In a question \( i \) the system-recognition entity mentions as \( M_i \), the ground truth annotated mentions in the evaluation set as \( A_i \), the evaluation corpus is \( Q \).

\[
\text{precision} = \frac{\sum_{i=1}^{|Q|} |M_i \cap A_i|}{\sum_{i=1}^{|Q|} |M_i|} \tag{2}
\]

\[
\text{recall} = \frac{\sum_{i=1}^{|Q|} |M_i \cap A_i|}{\sum_{i=1}^{|Q|} |A_i|} \tag{3}
\]

- **Entity type**: The prediction is considered the entity types name. \( M_{i,e} \) is system-recognition each entity mention’s type in question \( i \), \( A_{i,e} \) is the ground truth annotated entity mention’s type in question \( i \). If \( M_{i,e} = A_{i,e} \) or \( A_{i,e} \in P(M_{i,e}) \), we define it is a correct type name.

\[
\text{precision} = \frac{\sum_{i=1}^{|Q|} |M_{i,e} \cap A_{i,e}|}{\sum_{i=1}^{|Q|} |M_i|} \tag{4}
\]

\[
\text{recall} = \frac{\sum_{i=1}^{|Q|} |M_{i,e} \cap A_{i,e}|}{\sum_{i=1}^{|Q|} |A_i|} \tag{5}
\]

4.3. Comparison with State-of-the-art

We compare with the Stanford NER [27] on both entity mentions’ extract and coarse-grained types (Person, Location, and Organization). Stanford NER contain many features and is trained on 945 annotated documents (approximately 203,000 tokens). We utilized the 3000 questions sentences form evaluation corpus, because whole entity mentions in this set can be typed by these 3 types.

| Table 1. | Entity mention | Entity type |
|-----------|----------------|-------------|
| **Model** | Precision | Recall | F1 | Precision | Recall | F1 |
| Stanford(NER) | 0.791 | 0.777 | 0.727 | 0.742 | 0.713 | 0.727 |
| Model1 (no filter) | 0.556 | 0.864 | 0.676 | 0.511 | 0.801 | 0.623 |
| Model2 (add filter) | 0.788 | 0.801 | 0.7944 | 0.702 | 0.781 | 0.739 |
In table [1], Model 2 is our model, model1 is a similar to our model which removed the filter. From the table we can see, on entity mention level, compared with our model achieved comparable performance. At the same time, it shows, in our model, it is important to filter the entity mentions.

On entity type, though that Stanford(NER) is outstanding, it doesn’t work well for question sentence which is short text. Compare with Stanford(NER) our model is slightly outperform in our evaluation corpus.

Besides these tree coarse-gained types, there are many other types (such as movies, data, drug) discovered by fine-gained named entity recognition model. Figure [2] shows our fine-gained named entity recognition model’s performance. In our KNN model we set k=10, because there are some entity types which contain little labeled data. With the increase of types, the performance of our model becomes worse. The number of types is closed related to evaluate corpus. By comparing with the annotation data, we find 27 types are suitable for our corpus.

5. Conclusion and Future Work

We have addressed the task of fine-gained named entity recognition in QA text. To get more lexical features from short context, we extend the context by search engine. Specially we propose entity mention model to detect entity mentions which use the related sentence to extent information, and use a data-driven greedy n-grams algorithm to generate candidate entity mentions. We use a distant supervision method based DBpedia combine with the KNN algorithm to get fine-gained entity types. And the future work, includes: a) Our model heavily rely on the related sentences, we plane to use more efficient way to collect related sentences. 2) The last step need use manual feature to classify the entity mentions, maybe we can find more important feature to do this.

Acknowledgement

This work is supported by the National Natural Science Foundation of China(Grant nos. 61472329 and 61532009), youth fund of China(no. 61602389) and the Innovation Found of Postgraduate(ycjj2016170), Xihua University.

References

[1] D. Mollá, M. Van Zaanen, D. Smith, et al. Named entity recognition for question answering.
[2] H. Sun, H. Ma, W.-t. Yih, C.-T. Tsai, J. Liu, and M.-W.Chang. Open domain question answering
via semantic enrichment. In WWW.

[3] O. Etzioni. Search needs a shake-up. Nature, 476(7358):25–26, 2011.

[4] William A Woods. 1977. Lunar rocks in natural en-glish: Explorations in natural language question an-swering. In Linguistic structures processing. pages 521–569.

[5] David Ferrucci, Eric Brown, et al. 2010. Building watson: An overview of the deepqa project. AI magazine 31(3):59–79.

[6] Vanessa Lopez, et al. 2011. Is question answering fit for the semantic web?: a survey. Semantic Web 2(2):125–155.

[7] Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base.

[8] Xuchen Yao et al. 2014. Information extraction over structured data: Question answering with freebase. Association for Computational Linguistics, pages 956–966.

[9] K. Bollacker, et al. Freebase: a collaboratively created graph database for structuring human knowledge. In SIGMOD, pages 1247–1250. ACM, 2008.

[10] Kristina Toutanova, et al. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.

[11] C. Unger, et al. Gerber, and P. Cimiano. Template-based question answering over RDF data. In WWW, pages 639–648. ACM, 2012.

[12] R. Grishman et al. Message understanding conference-6: A brief history.

[13] S. Tardif, R.J. Curran, and T. Murphy, “Improved text categorisation for wikipedia named entities,” in Proceedings of ALTA Workshop, 2009.

[14] A. Toral and R. Muñoz, “A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia,” in Proceedings of Workshop on New Text, EACL, 2006. [Online].

[15] Y. Watanabe et al. A graph-based approach to named entity categorization in wikipedia using conditional random fields, Available: http://aclweb.org/anthology/D07-1068

[16] Mike Mintz, et al. Distant supervision for relation extraction without labeled data. Proceedings of the 47th Annual Meeting of the ACL.

[17] M. Fleischman and E. Hovy. Fine grained classification of named entities. In COLING.

[18] Lifu Huang, Jonathan May, Xiaoman Pan. Building a Fine-Grained Entity Typing System Oversight for a New X (X = Language, Domain, Genre). c 2016 ACM. DOI: 10.1145/1235

[19] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, Wei Wang. KBQA: Learning Question Answering over QA Corpora and Knowledge Bases. Proceedings of the VLDB Endowment, Vol. 10, No. 5 Copyright 2017 VLDB Endowment 2150-8097/17/01.

[20] Kate R J, Mooney R J. Joint Entity and relation Extraction using card-pyramid Parsing [c]/C onference on C omputational N atural L anguage learning. 2010:203-212.

[21] Miwa M, Bansal M. End-to-End Relation Extraction using LST Ms on S equences and tree structures [c]/ annual Meeting of the association for computational linguistics.

[22] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1508.01991.

[23] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In ICML, 2001.

[24] James Hammerton. 2003. Named entity recognition with long short-term memory. In Walter Daelemans and Miles Osborne, editors, Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, pages 172–175. ACL.

[25] W. Dakka and S. Cucerzan, “Augmenting wikipedia with named entity tags,” in of 3rd IJCNLP, 2008. [Online]. Available: http://aclweb.org/anthology/I08-1071

[26] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel, “The automatic content extraction (ace) program tasks, data, and evaluation,” in Proceedings of LREC, 2004. [Online]. Available: http://aclweb.org/anthology/L04-1011
[27] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In ACL
[28] Ling Xiao et al. Fine-gained entity recognition [C]//Proc of the 26th Conf on Association for the Advancement of Artificial Intelligence. Menlo Park. CA: AAAI. 2012: 94-100.
[29] R. Higashinaka, K. Sadamitsu, K. Saito, T. Makino, and Y. Matsuo, “Creating an extended named entity dictionary from wikipedia, Available: http://aclweb.org/anthology/C12-1071
[30] A. El-Kishky, et al. Scalable topical phrase mining from text corpora. VLDB, 2015
[31] L. Banarescu, C. Bonial, et. Abstract meaning representation for sembanking. In ACL Workshop on Linguistic Annotation and Interoperability with Discourse, 2013.
[32] L. Zou, et al. Natural language question answering over RDF: a graph data driven approach. In SIGMOD, pages 313–324. ACM
[33] Masatoshi Suzuki, et al. Fine-Grained Named Entity Classification with Wikipedia Article Vectors. 978-1-5090-4470-2/16 $31.00 © 2016 IEEE DOI 10.1109/WI.2016.78
[34] Diego Molla, et al. Named Entity Recognition for Question Answering. Proceedings of the 2006 Australasian Language Technology Workshop (ALTW2006), pages 51–58.