Query-Based Named Entity Recognition

Yuxian Meng, Xiaoya Li, Zijun Sun and Jiwei Li
Shannon.AI
{yuxian_meng, xiaoya_li, zijun_sun, jiwei_li}@shannonai.com

Abstract

In this paper, we propose a new strategy for the task of named entity recognition (NER). We cast the task as a query-based machine reading comprehension task: e.g., the task of extracting entities with PER is formalized as answering the question of *"which person is mentioned in the text ?"*. Such a strategy comes with the advantage that it solves the long-standing issue of handling overlapping or nested entities (the same token that participates in more than one entity categories) with sequence-labeling techniques for NER. Additionally, since the query encodes informative prior knowledge, this strategy facilitates the process of entity extraction, leading to better performances.

We experiment the proposed model on five widely used NER datasets on English and Chinese, including MSRA, Resume, OntoNotes, ACE04 and ACE05. The proposed model sets new SOTA results on all of these datasets.

1 Introduction

Named entity recognition （NER） is a basic task in building natural language processing (NLP) systems. The task is traditionally formalized as a sequence labeling problem, in which an algorithm needs to assign a tagging class to each word or character within a sequence. Depending on whether the prediction of the label is made based on its proceeding/surrounding labels, existing models can be divided into two major categories: (1) autoregressive ones such as CRF-based models (Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016a; Zhang and Yang, 2018) and (2) non-autoregressive ones such as BERT (Devlin et al., 2018).

Existing approaches, both autoregressive and non-autoregressive ones, come with some intrinsic drawbacks at both the formalization level and the algorithmic level. At the formalization level, most current models are incapable of handling overlapping or nested entities (Kim et al., 2003; Finkel and Manning, 2009; Lu and Roth, 2015; Katiyar and Cardie, 2018). This is because of the fact that one

token can only be assigned to one tag category. At the algorithmic level, tagging classes are merely indexes and do not encode any prior information about entity categories. This lack of clarity on what to extract leads to inferior performances.

In this paper, we propose a new paradigm to extract named entities. We formalize the task as a question answering task in machine reading comprehension: each entity type is characterized by a natural language query, and entities are extracted by answering these queries given context. For example, the task of assigning the PER label to *Washington* in *[Washington] was born into slavery on the farm of James Burroughs* is formalized as answering the question *which person is mentioned in the text ?*.

Such a type of formalization to a large extent solves the aforementioned issues: (1) the model is able to naturally handle the entity overlapping issue: regarding different entity categories, the model extracts corresponding entity spans by answering different (and independent) questions; (2) the query encodes significant prior information about the entity class to extract. For example, the semantic relatedness between the query *who is mentioned in the text* and the PER entities facilitatew the extracting process, potentially leading to better performances; (3) We are able to take advantages of current well-developed sophisticated MRC models.

Using the proposed strategy, we are able to achieve SOTA results on five NER datasets in English and Chinese, including MSRA, RESUME, Chinese OntoNotes, ACE04 and ACE05.

2 Related Work

2.1 Named Entity Recognition

Traditional sequence labeling models use CRFs (Lafferty et al., 2001; Sutton et al., 2007) as a backbone. The first work using neural models for NER goes back to 2003, when Hammerton (2003) attempted to solve the problem using unidirectional LSTMs. Collobert et al. (2011) presented the CNN-CRF structure, augmented with character embed-

dings by Santos and Guimaraes (2015). Lample et al. (2016) explored neural structures for NER, in which the bidirectional LSTMs are combined with CRFs with features based on character-based word representations and unsupervised word representations. Ma and Hovy (2016) and Chiu and Nichols (2016b) used a character CNN to extract features from characters. Recent large-scale language model pertaining methods such as BERT (Devlin et al., 2018) and Elmo (Peters et al., 2018) further enhance the performance of NER, yielding state-of-the-art performances.

Nested NER refers to a situation in which overlapping or nested entity mentions exist. This phenomenon was first noticed by Kim et al. (2003), in which rules were used to identify overlapping mentions. Finkel and Manning (2009) made the assumption that one mention is fully contained by the other when they overlap and built a model to extract nested entity mentions based on parse trees. Lu and Roth (2015) proposed to use mention hypergraphs for recognizing overlapping mentions. Xu et al. (2017) utilize a local classifier that runs on every possible span to detect overlapping mentions and Katiyar and Cardie (2018) used neural model to learn the hypergraph representations for nested entities.

2.2 Machine Reading Comprehension

MRC models extract answer spans from passages given questions (Seo et al., 2016; Wang et al., 2016; Wang and Jiang, 2016; Xiong et al., 2016, 2017; Wang et al., 2016; Shen et al., 2017; Chen et al., 2017). The task can be formalized as two multiclass classification tasks, i.e., predicting the starting and ending positions of the answer spans given questions.

Many NLP tasks can be transformed to the task of question answering. For example, Levy et al. (2017) transformed the task of relation extraction to a QA task: each relation type R(x,y) can be parameterized as a question q(x) whose answer is y. For example, the relation EDUCATED-AT can be mapped to "Where did x study?". Given a question $q(x)$, if a non-null answer y can be extracted from a sentence, it means the relation label for the current sentence is R. McCann et al. (2018) transforms NLP tasks such as summarization or sentiment analysis into question answering, For example, the task of summarization can be formalized as answering the question *What is the summary?*.

Input: sequence $X = \{x_1, x_2, ..., x_n\}$, QuestionTemplates,
Output: sequence labels $\{y_i, y_2, ..., y_n\}$
1:
2: $y \leftarrow [o] \times n$
3: **for** question q_y in QuestionTemplates **do**
4:     $\{x_{start}, ..., x_{end}\} = \text{MRC}(X, q_y)$
5:     **if** $\{x_{start}, ..., x_{end}\} \neq \text{NULL}$ **do**
6:         $\{y_{start}, ..., y_{end}\} = y$
7:     **endif**
8: **end for**
9: **return** y

Algorithm 1: Overview of the proposed model.

3 Model

3.1 System Overview

Given a word or character sequence $X = \{x_1, x_2, ..., x_n\}$, where n denotes the length of the sequence, we need to assign each token x_i a label $y_i \in Y$, indicating the label for x_i. y_i is selected from the predefined list Y for tag types (e.g., PER, LOC, etc).

For each tag type $y \in Y$, it is associated with a natural language question q_y. Given X and question q_y, the MRC model is run to predict the starting index $start \in [1, n]$ and the ending index $end \in [1, n]$. This leads to the extracted answer span $[x_{start}, x_{start+1}, ..., x_{end-1}, x_{end}]$. The MRC model allows returning a special NULL token, indicating that no substring within x should be used as the answer to q_y. If the return is not NULL, the label for $[x_{start}, x_{start+1}, ..., x_{end-1}, x_{end}]$ will be changed to y. We iterate this process for all tagging categories until the end. The overview of the algorithm is shown in Algorithm 1.

Entity	Natural Language Question
Facility	Which facility is mentioned in the text?
Location	Which location is mentioned in the text?
Person	Which is Person mentioned in the text?

Table 1: Examples for transforming different entity categories to question queries.

3.2 Extracting Answer Spans via MRC

Each type of the entity is associated with a natural language question generated from templates, the details of which are shown in Table 1.

Given the question q_y, we need to extract text span $x_{start}, ..., x_{end}$ from the text X given the question q_y using MRC frameworks. We use BERT (Devlin et al., 2018) as a backbone. BERT utilizes large-scale pretraining based on language models and achieves SOTA results on MRC datasets like SQUAD (Rajpurkar et al., 2016) To be in line with

BERT, the question q_y and the passage X are concatenated, forming the combined string [CLS, q_y, SEP, X, SEP], where CLS and SEP are special tokens.

There are two commonly adopted strategies for span prediction in MRC: the first strategy (Seo et al., 2016; Wang et al., 2016) is to have two n-class classifiers to specially predict staring and ending indexes. The other strategy is to have n three-class classifiers: for each token $x_i \in [1, n]$, the model predicts whether it is a start, an end or neither. These two strategies are the same in nature, but might lead to different performances empirically. In this work, we choose the latter since it yields better performances.[1]

3.3 Training Objective

One of the key issue with the starting and ending index prediction for MRC tasks is the data imbalance issue: given the query q_y and sentence X, there is at most one token labeled as starting or ending, while all the rest are non-starting or non-ending. To deal with this issue, we use dice loss (Milletari and Ahmadi, 2016) instead of cross entropy as the training objective. Dice loss is first proposed for medical image segmentation tasks to handle the situation where there is a strong imbalance between the number of foreground and background pixels. It can be thought as a objective function optimizing for F score rather than accuracy, which cross entropy is approximately optimizing for. The dice loss can be formulated as follows:

$$L_{dice} = 1 - \frac{2 \sum_{i=0}^{n} p_i g_i + \lambda}{\sum_{i=0}^{n} p_i^2 + \sum_{i=0}^{n} g_i^2 + \lambda} \quad (1)$$

wehre $p_i \in [0, 1]$ denotes the starting/ending probability output from the model for i^{th} token. $g_i \in \{0, 1\}$ denotes the golden probability. The hyperparameter λ controls the trade-off between precision and recall.

$$L_{dice} = 1 - \frac{2 \sum_{i=0}^{L} p_i g_i}{\sum_{i=0}^{L} p_i^2 + \sum_{i=0}^{L} g_i^2 + \lambda} - \frac{\lambda}{\sum_{i=0}^{L} p_i^2 + \sum_{i=0}^{L} g_i^2 + \lambda}$$

Looking at the denominator of the first part, we can see that for negative examples with $g_i = 0$, their p_i

---

[1]A further post-processing strategy is needed: if more than two positions within X are predicted as starting positions, we select the one with the smallest index; if more than two positions are predicted as ending indexes, we select the one with the largest index. The algorithm returns NULL if no starting or ending index is found.

won't contribute. One can think this as a specific objective for recall. For the second part, L_{dice} will still be penalized even if $g_i = 0$, shooting for high precision scores.

4 Experiments

4.1 Datasets and Training

Experiments are conducted in the following datasets: **Benchmark NER**: We use MSRA (Levow, 2006), Chinese OntoNotes 4.0 (Weischedel et al., 2011), and Resume dataset (Zhang and Yang, 2018); **Overlap NER**: We use ACE 2004 and ACE 2005 (Doddington et al., 2004).

For English datasets, we use the pre-trained BERT model with cased for initialization. And text is tokenized using WordPiece Tokenizer. For Chinese datasets, we use the pre-trained BERT model. All hyperparameters such as learning rate, dropout and batch size are tuned using grid search on development set.

4.2 Baseline Approaches

We consider the following modelsm as baselines:
- **Lattice LSTM:** the word-character lattice model proposed by Zhang and Yang (2018) constructs a word-character lattice.
- **Glyce Lattice LSTM**: Wu et al. (2019) utilizes glyph information of Chinese characters into Lattice LSTM model.
- **Hyper-graph LSTM:** Katiyar and Cardie (2018) proposes a hypergraph-based model that uses LSTM for learning feature representations.
- **Transition Model:** Wang et al. (2018) introduces a scalable transition-based method to model the nested structure of mentions.
- **Segmental Hypergraph Model :** Wang and Lu (2018) proposes a segmental hypergargh representation to model overlapping entity mentions.

4.3 Results and Discussions

Table 2 presents the comparisons between our model and the current state-of-the-art NER models. For MSRA, our model outperforms fine-tuning BERT by +0.95% in terms of F-scores, achieving the new state-of-the-art. On Chinese OntoNotes, our model achieve a huge gain of 2.95% improvement in terms of F-score. Resume is released by Zhang and Yang (2018) and it contains eight fine-grained entity categories. Since queries contain

MSRA			
Model	P	R	F
Zhang and Yang (2018)	93.57	92.79	93.18
Wu et al. (2019)	93.86	93.92	93.89
BERT Tagger	94.97	94.62	94.80
BERT Query	**96.18**	**95.12**	**95.75** (+0.95)

Resume			
Model	P	R	F
Zhang and Yang (2018)	94.81	94.11	94.46
Wu et al. (2019)	95.72	95.63	95.67
BERT Tagger	96.12	95.45	95.78
BERT Query	**97.33**	**96.61**	**96.97** (+1.19)

Chinese OntoNotes			
Model	P	R	F
Zhang and Yang (2018)	76.35	71.56	73.88
Wu et al. (2019)	82.06	68.74	74.81
BERT Tagger	78.01	80.35	79.16
BERT Query	**82.98**	**81.25**	**82.11** (+2.95)

ACE 2004			
Model	P	R	F
Katiyar and Cardie (2018)	73.6	71.8	72.7
Wang et al. (2018)	74.9	71.8	73.3
Wang and Lu (2018)	78.0	72.4	75.1
BERT Tagger	79.39	79.97	79.68
BERT Query	**84.05**	**84.23**	**84.14** (+4.46)

ACE 2005			
Model	P	R	F
Katiyar and Cardie (2018)	70.6	70.4	70.5
Wang et al. (2018)	74.5	71.5	73.0
Wang and Lu (2018)	76.8	72.3	74.5
BERT Tagger	78.21	82.74	80.41
BERT Query	**87.16**	**86.59**	**86.88** (+6.47)

Table 2: Results for NER tasks.

Figure 1: model performance decrease with less training data

Datasets	loss	f-1 score
Chinese OntoNotes	Entropy	80.20
Chinese OntoNotes	Dice Loss	82.11 (+1.91)
MSRA	Entropy	95.41
MSRA	Dice Loss	95.75 (+0.34)
Resume	Entropy	96.83
Resume	Dice Loss	96.97 (+0.14)

Table 3: Query Type Samples

Chinese OntoNotes			
Query Type	P	R	F
index	81.35	80.92	81.14
pseudo	82.52	81.13	81.82
natural	82.98	81.25	82.11

Table 4: Results of different types of queries

semantic prior knowledge, our model enhanced the performance compared with fine-tuning BERT tagger. On ACE 2004, our model achieved state-of-the-art performance with 84.14% in terms of F-scores. For ACE 2005, we enhance the F-score from 74.5% to 86.88%.

5 Ablation study

5.1 Size of Training Data

Since the natural language query encodes significant prior knowledge, we expect that the proposed framework works better with less training data. Figure1 verifies this point: on the OntoNotes training set, the query-based approach achieves comparable performance to BERT even when with half amount of training data.

5.2 Different Query Choices

To analyze the impact of query, we compared 3 kinds of query: 1) index query (e.g.,"one", "two",
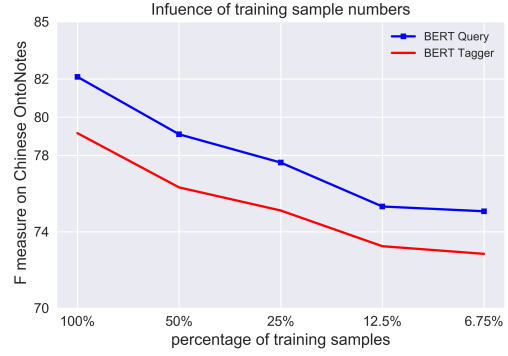
"three"), 2) pseudo query (e.g.,"person", "location","company") and 3) natural language query. Performances regarding different strategies are shown in Table4. We find natural language queries lead to best performance due to the concrete knowledge they encode.

5.3 Different Loss Functions

We compare the performances for dice loss and cross-entropy loss in Table 3. As can be seen, dice loss yields significant performance boost than cross-entropy loss.

6 Conclusion

In this paper, we reformalize the NER task as a MRC question answering task. This formalization comes with several key advantages: (1) being capable of addressing overlapping or nested entities; (2) the query encoding significant prior knowledge about the entity category to extract. This proposed strategy obtains SOTA results on five different NER datasets.

ss

References

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Jason PC Chiu and Eric Nichols. 2016a. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Jason PC Chiu and Eric Nichols. 2016b. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, page 1.

Jenny Rose Finkel and Christopher D Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 141–150. Association for Computational Linguistics.

James Hammerton. 2003. Named entity recognition with long short-term memory. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 172–175. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Gina-Anne Levow. 2006. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth Workshop on Chinese Language Processing, SIGHAN@COLING/ACL 2006, Sydney, Australia, July 22-23, 2006*, pages 108–117.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. *arXiv preprint arXiv:1706.04115*.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Nassir Navab Milletari, Fausto and Seyeds-Ahmad Ahmadi. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. *2016 Fourth International Conference on 3D Vision (3DV)*.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Cicero Nogueira dos Santos and Victor Guimaraes. 2015. Boosting named entity recognition with neural character embeddings. *arXiv preprint arXiv:1505.05008*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Yelong Shen, Po-Sen Huang, Jianfeng Gao, and Weizhu Chen. 2017. Reasonet: Learning to stop reading in machine comprehension. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1047–1055. ACM.

Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling

and segmenting sequence data. *Journal of Machine Learning Research*, 8(Mar):693–723.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. *arXiv preprint arXiv:1810.01817*.

Bailin Wang, Wei Lu, Yu Wang, and Hongxia Jin. 2018. A neural transition-based model for nested mention recognition. *arXiv preprint arXiv:1810.01808*.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Zhiguo Wang, Haitao Mi, Wael Hamza, and Radu Florian. 2016. Multi-perspective context matching for machine comprehension. *arXiv preprint arXiv:1612.04211*.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. 2011. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*.

Wei Wu, Yuxian Meng, Qinghong Han, Muyu Li, Xiaoya Li, Jie Mei, Ping Nie, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. *arXiv preprint arXiv:1901.10125*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*.

Caiming Xiong, Victor Zhong, and Richard Socher. 2017. Dcn+: Mixed objective and deep residual coattention for question answering. *arXiv preprint arXiv:1711.00106*.

Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1237–1247.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. *arXiv preprint arXiv:1805.02023*.