A Unified MRC Framework for Named Entity Recognition

Xiaoya Li[1*], Jingrong Feng[1*], Yuxian Meng[1], Qinghong Han[1], Fei Wu[2] and Jiwei Li[1]
[1] Shannon.AI
[2] Department of Computer Science and Technology, Zhejiang University
{xiaoya_li, jingrong_feng, yuxian_meng, qinghong_han, jiwei_li}@shannonai.com, wufei@zju.edu.cn

Abstract

The task of named entity recognition (NER) is normally divided into nested NER and flat NER depending on whether named entities are nested or not. Models are usually separately developed for the two tasks, since sequence labeling models, the most widely used backbone for flat NER, are only able to assign a single label to a particular token, which is unsuitable for nested NER where a token may be assigned several labels.

In this paper, we propose a unified framework that is capable of handling both flat and nested NER tasks. Instead of treating the task of NER as a sequence labeling problem, we propose to formulate it as a machine reading comprehension (MRC) task. For example, extracting entities with the PER label is formalized as extracting answer spans to the question "*which person is mentioned in the text?*". This formulation naturally tackles the entity overlapping issue in nested NER: the extraction of two overlapping entities from different categories requires answering two independent questions. Additionally, since the query encodes informative prior knowledge, this strategy facilitates the process of entity extraction, leading to better performances for not only nested NER, but flat NER.

We conduct experiments on both *nested* and *flat* NER datasets. Experimental results demonstrate the effectiveness of the proposed formulation. We are able to achieve vast amount of performance boost over current SOTA models on nested NER datasets, i.e., +1.28, +2.55, +5.44, +6.37, respectively on ACE04, ACE05, GENIA and KBP17, along with SOTA results on flat NER datasets, i.e., +0.24, +1.95, +0.21, +1.49 respectively on English CoNLL 2003, English OntoNotes 5.0, Chinese MSRA, Chinese OntoNotes 4.0. [1] [2] [3]

[1] This paper includes material from the unpublished manuscript "Query-Based Named Entity Recognition".
[2] Xiaoya and Jingrong contribute equally to this paper.
[3] Code is coming soon.

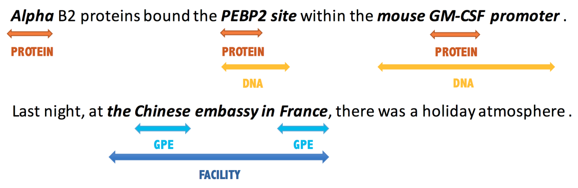

Figure 1: Examples for *nested* entities from GENIA and ACE04 corpora.

1 Introduction

Named Entity Recognition (NER) refers to the task of detecting the span and the semantic category of entities from a chunk of text. The task can be further divided into two sub-categories, flat NER and nested NER, depending on whether named entities are overlapping or nested. Nested NER refers to a phenomenon that the spans of entities (mentions) are nested, as shown in Figure 1. Entity overlapping is a fairly common phenomenon in natural languages.

The task of flat NER is commonly formalized as a sequence labeling task: a sequence labeling model (Chiu and Nichols, 2016; Ma and Hovy, 2016; Devlin et al., 2018) is trained to assign a tagging class to each unit within a sequence of tokens. This formulation is unfortunately incapable of handling overlapping entities in nested NER (Huang et al., 2015; Chiu and Nichols, 2015), where multiple categories need to be assigned to a single token if the token participates in multiple entities. Many attempts tried to reconcile sequence labeling models with nested NER (Alex et al., 2007; Byrne, 2007; Finkel and Manning, 2009; Lu and Roth, 2015; Katiyar and Cardie, 2018), mostly based on the pipelined systems. However, pipelined systems suffer from the disadvantages of error propagation, long running time, the intensiveness in developing hand-crafted features, etc.

Inspired by the current trend of formalizing NLP problems as question answering tasks (Levy et al., 2017; McCann et al., 2018; Li et al., 2019), we propose a new framework for NER that is capable of handling both flat and nested NER. Instead of treating the task of NER as a sequence labeling problem, we propose to formulate it as a SQuAD-style machine reading comprehension (MRC) task. Each entity type is characterized by a natural language query, and entities are extracted by answering these queries given the contexts. For example, the task of assigning the PER label to "*[Washington] was born into slavery on the farm of James Burroughs*" is formalized as answering the question "*which person is mentioned in the text?*". This strategy naturally tackles the entity overlapping issue in nested NER: the extraction of two entities that overlap requires answering two independent questions.

The MRC formulation also comes with another key advantage over the sequence labeling formulation. For the latter, golden NER categories are merely class indexes and lack for semantic prior information for entity categories. For example, the ORG class is merely treated as a one-hot vector in cross entropy in sequence labeling training. This lack of clarity on what to extract leads to inferior performances. On the contrary, for the MRC formulation, the query encodes significant prior information about the entity category to extract. For example, the query *"find an organization such as company, agency and institution in the context"* encourages the model to link the word *"organization"* in the query to location entities in the context. Additionally, by encoding comprehensive descriptions (e.g., *"company, agency and institution"*) of tagging categories (e.g., *ORG*) in the query, the model has the potential to disambiguate similar tagging classes.

We conduct experiments on both *nested* and *flat* NER datasets in order to show the generality of our approach. Experimental results demonstrate its effectiveness. We are able to achieve vast amount of performance boost over current SOTA models on nested NER datasets, i.e., +1.28, +2.55, +5.44, +6.37, respectively on ACE04, ACE05, GENIA and KBP17, along with SOTA results on flat NER datasets, i.e., +0.24, +1.95, +0.21, +1.49 respectively on English CoNLL 2003, English OntoNotes 5.0, Chinese MSRA, Chinese OntoNotes 4.0. We wish that our work would inspire the introduction of new paradigms for the entity recognition task.

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 for NER. 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 embeddings 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 (2016) used a character CNN to extract features from characters. Recent large-scale language model pretraining methods such as BERT (Devlin et al., 2018) and ELMo (Peters et al., 2018a) further enhance the performance of NER, yielding state-of-the-art performances.

2.2 Nested Named Entity Recognition

The overlapping between entities (mentions) was first noticed by Kim et al. (2003), who developed handcrafted rules to identify overlapping mentions. Alex et al. (2007) proposed two multi-layer CRF models for nested NER. The first is the inside-out model, in which the first CRF identifies the innermost entities, and the successive layer CRF is built over words and the innermost entities extracted from the previous CRF to identify second-level entities, etc. The other is the outside-in model, in which the first CRF identifies outermost entities, and then successive CRFs would identify increasingly nested entities. Finkel and Manning (2009) built a model to extract nested entity mentions based on parse trees. They made the assumption that one mention is fully contained by the other when they overlap. Lu and Roth (2015) proposed to use mention hyper-graphs for recognizing overlapping mentions. Xu et al. (2017) utilized a local classifier that runs on every possible span to detect overlapping mentions and Katiyar and Cardie (2018) used neural models to learn the hyper-graph representations for nested entities. Lin et al. (2019a) proposed the Anchor-Region Networks (ARNs) architecture by modeling and leveraging the head-driven phrase structures of nested entity mentions. Luan et al. (2019)

built a span enumeration approach by selecting the most confident entity spans and linking these nodes with confidence-weighted relation types and coreferences.

Recently, nested NER models are enriched with pre-trained contextual embeddings such as BERT (Devlin et al., 2018) and ELMo (Peters et al., 2018b). Fisher and Vlachos (2019) introduced a BERT-based model that first merges tokens and/or entities into entities, and then assigned labeled to these entities. Shibuya and Hovy (2019) provided inference model that extracts entities iteratively from outermost ones to inner ones. Straková et al. (2019) viewed nested NER as a sequence-to-sequence generation problem, in which the input sequence is a list of tokens and the target sequence is a list of labels.

2.3 Machine Reading Comprehension (MRC)

MRC models (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) extract answer spans from passages given questions. The task can be formalized as two multi-class classification tasks, i.e., predicting the starting and ending positions of the answer spans given questions.

Over the past one or two years, there has been a trend of transforming NLP tasks to MRC 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) transformed 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?*". Our work is significantly inspired by Li et al. (2019), which formalized the task of entity-relation extraction as a multi-turn question answering task. Different from this work, Li et al. (2019) focused on relation extraction rather than NER. Additionally, Li et al. (2019) utilized a template-based procedure for constructing queries to extract semantic relations between entities and their queries lack of diversity. In this paper, queries incorporate more factual knowledge such as synonyms and examples, and we present an in-depth analysis of the impact of strategies of building queries.

3 NER as MRC

3.1 Task Formalization

Given an input sequence $X = \{x_1, x_2, ..., x_n\}$, where n denotes the length of the sequence, we need to find each entity $x_{\text{start,end}}$ in X, and then lable a label $y \in Y$ to it, where y is selected from the predefined list Y for tag types (e.g., PER, LOC, etc).

Dataset Construction We first need to transform the tagging-style annotated NER dataset to a set of (QUESTION, ANSWER, CONTEXT) triples. For each tag type $y \in Y$, it is associated with a natural language question $q_y = \{q_1, q_2, ..., q_m\}$, where m denotes the length of the generated query. An annotated entity $x_{\text{start,end}} = \{x_{\text{start}}, x_{\text{start+1}}, \cdots, x_{\text{end-1}}, x_{\text{end}}\}$ is a substring of X satisfying start $\leq$ end. It's associated with a golden label $y \in Y$. By generating a natural language question q_y based on the label y, we can obtain the triple (q_y, $x_{\text{start,end}}$, X), which is exactly the (QUESTION, ANSWER, CONTEXT) triple that we need. Note that we use the subscript "$_{\text{start,end}}$" to denote the continuous elements from index 'start' to index 'end' in a sequence.

3.2 Query Generation

The question generation procedure is important since queries encode prior knowledge about labels and have a significant influence on the final results. Different ways have been proposed for question generation, e.g., Li et al. (2019) utilized a template-based procedure for constructing queries to extract semantic relations between entities. In this paper, we take annotation guideline notes as references to construct queries. Annotation guideline notes are the guidelines provided to the annotators of the dataset by the dataset builder. They are theoretical descriptions of the tag categories, which are described as generic and precise as possible so that human annotators can annotate the concepts or phenomena in any texts without running into ambiguity issues. Examples are shown in Table 1.

Entity	Natural Language Question
Location	Find locations in the text, including non-geographical locations, mountain ranges and bodies of water
Facility	Find facilities in the text, including buildings, airports, highways and bridges
Organization	Find organizations in the text, including companies, agencies and institutions

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

3.3 Model Details

Model Backbone Given the question q_y, we need to extract the text span $x_{\text{start,end}}$ from the text X given the question q_y using MRC frameworks. We use BERT (Devlin et al., 2018) as the backbone. To be in line with BERT, the question q_y and the passage X are concatenated, forming the combined string $\{[\text{CLS}], q_1, q_2, ..., q_m, [\text{SEP}], x_1, x_2, ..., x_n\}$, where [CLS] and [SEP] are special tokens. Then BERT receives the combined string and outputs a representation matrix $E \in \mathbb{R}^{n\times d}$, where d is the vector dimension of the last layer of BERT.

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 separately predict the start index and the end index, where n denotes the length of the context. Since the softmax function is put over all tokens in the context, this strategy has the disadvantage of only being able to output a single span given a query; the other strategy is to have two binary classifiers, one to predict whether each token is the start index or not, the other to predict whether each token is the end index or not. This strategy allows for outputting multiple start indexes and multiple end indexes for a given context and a specific query, and thus has the potentials to extract all related entities according to q_y. We adopt the second strategy and describe the details below:

Start Index Prediction Given representation matrix E output from BERT, the model first predicts the probability of each token being a start index as follows:

$$P_{\text{start}} = \text{softmax}_{\text{each row}}(E \cdot T_{\text{start}}) \in \mathbb{R}^{N\times 2} \quad (1)$$

$T_{\text{start}} \in \mathbb{R}^{d\times 2}$ is the parameter to learn. Each row of P_{start} means the probability distribution of each index being the start position of an entity given the query.

End Index Prediction The End Index Prediction procedure is exactly the same as Start Index Prediction, except that we have another matrix T_{end} to obtain probability matrix $P_{\text{end}} \in \mathbb{R}^{N\times 2}$.

Start-End Matching In the context, there could be multiple entities of the same category. This means that multiple start indexes could be predicted from the *start-index prediction* model and multiple end indexes predicted from the *end-index prediction* model. The heuristic of matching the start index with its nearest end index does not work here since entities could overlap. We thus further need a model to match a predicted start index with its corresponding end index.

Specifically, by applying argmax to each row of P_{start} and P_{end}, we will get the predicted indexes, i.e., 0-1 sequences $\hat{I}_{\text{start}}$ and $\hat{I}_{\text{end}}$of length n:

$$\begin{aligned} \hat{I}_{\text{start}} &= \text{argmax}_{\text{each row}}(P_{\text{start}}) \\ \hat{I}_{\text{end}} &= \text{argmax}_{\text{each row}}(P_{\text{end}}) \end{aligned} \quad (2)$$

Given any start index $i_{\text{start}} \in \hat{I}_{\text{start}}$ and end index $i_{\text{end}} \in \hat{I}_{\text{end}}$, a binary classification model is trained to predict the probability that they should be matched, given as follows:

$$p_{i_{\text{start}}, j_{\text{end}}} = \text{sigmoid}(m \cdot \text{concat}(E_i, E_j)) \quad (3)$$

where $m \in \mathbb{R}^{1\times 2d}$.

3.4 Train and Test

At training time, X is paired with two label sequences Y_{start} and Y_{end} of length n representing the ground-truth label of each token x_i being the start index or end index of any entity. We therefore have the following two losses for start and end index predictions:

$$\begin{aligned} \mathcal{L}_{\text{start}} &= \text{CE}(P_{\text{start}}, Y_{\text{start}}) \\ \mathcal{L}_{\text{end}} &= \text{CE}(P_{\text{end}}, Y_{\text{end}}) \end{aligned} \quad (4)$$

Let $Y_{\text{start, end}}$ denote the golden labels for whether each start index should be matched with each end index. The start-end index matching loss is given as follows:

$$\mathcal{L}_{\text{span}} = \text{CE}(P_{i_{\text{start}}, j_{\text{end}}}, Y_{\text{start, end}}) \quad (5)$$

The overall training objective to be minimized is as follows:

$$\mathcal{L} = \mathcal{L}_{\text{start}} + \mathcal{L}_{\text{end}} + \mathcal{L}_{\text{span}} \quad (6)$$

The three losses are jointly trained in an end-to-end fashion, with parameters shared at the BERT layer. At test time, start and end indexes are first separately selected based on P_{start} and P_{end}. Then the index matching model is used to align the extracted start indexes with end indexes, leading to the final extracted answers.[4]

3.5 Experiments on Nested NER

3.5.1 Datasets

For *nested* NER, experiments are conducted on the widely-used ACE 2004, ACE 2005, GENIA and KBP2017 datasets, which respectively contain 24%, 22%, 10% and 19% nested mentions. Hyperparameters are tuned on their corresponding development sets. For evaluation, we use span-level micro-averaged precision, recall and F1.

ACE 2004 and ACE 2005 (Doddington et al., 2005; Christopher Walker and Maeda, 2006) [5]: The two datasets each contain 7 entity categories. For each entity type, there are annotations for both the entity mentions and mention heads. For fair comparison, we exactly follow the data preprocessing strategy in Katiyar and Cardie (2018) and Lin et al. (2019b) by keeping files from `bn`, `nw` and `wl`, and splitting these files into train, dev and test sets by 8:1:1, respectively.

GENIA (Ohta et al., 2002) For the GENIA dataset, we use GENIAcorpus3.02p[6]. We follow the protocols in Katiyar and Cardie (2018).

KBP2017 We follow Katiyar and Cardie (2018) and evaluate our model on the 2017 English evaluation dataset (LDC2017D55). Training set consists of RichERE annotated datasets[7]. We follow the dataset split strategy in Lin et al. (2019b).

3.5.2 Baselines

We use the following models as baselines:

- **Hyper-Graph:** Katiyar and Cardie (2018) proposes a hypergraph-based model based on LSTMs.
- **Seg-Graph:** Wang and Lu (2018) proposes a segmental hypergargh representation to model overlapping entity mentions.
- **ARN:** Lin et al. (2019a) proposes Anchor-Region Networks by modeling and levraging the head-driven phrase structures of entity mentions.
- **KBP17-Best:** Ji et al. (2017) gives an overview of the Entity Discovery task at the Knowledge Base Population (KBP) track at TAC2017 and also reports previous best results for the task of nested NER.
- **Seq2Seq-BERT:** Straková et al. (2019) views the *nested* NER as a sequence-to-sequence problem. Input to the model is word tokens and the output sequence consists of labels.
- **Path-BERT:** Shibuya and Hovy (2019) treats the tag sequence as the second best path within in the span of their parent entity based on BERT.
- **Merge-BERT:** Fisher and Vlachos (2019) proposes a merge and label method based on BERT.
- **DYGIE:** Luan et al. (2019) introduces a general framework that share span representations using dynamically constructed span graphs.

3.5.3 Results and Discussions

Table 2 shows experimental results on *nested* NER datasets. We observe huge performance boosts on the nested NER datasets over previous SOTA models, achieving F1 scores of 85.98%, 86.88%, 83.75% and 80.97% on ACE04, ACE05, GENIA and KBP-2017 datasets, which are +1.28%, +2.55%, +5.44% and 6.37% over previous SOTA performances. It is worth noting that the proposed models are compared against the strong BERT-based baselines.

3.6 Experiments on Flat NER

3.6.1 Datasets

For *flat* NER, experiments are conducted on both English datasets i.e. CoNLL2003 and OntoNotes 5.0 and Chinese datasets i.e. OntoNotes 4.0 and MSRA. Hyperparameters are tuned on their corresponding development sets. We report span-level micro-averaged precision, recall and F1 scores for evaluation.

[4] We also tried end-to-end inference at test time, which directly ranks each span based on Eq. 5. We did not observe statistically significant improvement from the end-to-end inference strategy, and thus abandon it due to the computation cost, as it has to ranks each span within the text.

[5] https://catalog.ldc.upenn.edu/LDC2005T09 (ACE2004) and https://catalog.ldc.upenn.edu/LDC2006T06 (ACE2005)

[6] http://www.geniaproject.org/genia-corpus/pos-annotation

[7] RichRE annotated datasets include LDC2015E29, LDC2015E68, LDC2016E31 and LDC2017E02

English ACE 2004			
Model	P	R	F
Hyper-Graph (Katiyar and Cardie, 2018)	73.6	71.8	72.7
Seg-Graph (Wang and Lu, 2018)	78.0	72.4	75.1
Seq2seq-BERT (Straková et al., 2019)	-	-	84.40
Path-BERT (Shibuya and Hovy, 2019)	83.73	81.91	82.81
DYGIE (Luan et al., 2019)	-	-	84.7
BERT-MRC	**85.05**	**86.32**	**85.98** **(+1.28)**
English ACE 2005			
Model	P	R	F
Hyper-Graph (Katiyar and Cardie, 2018)	70.6	70.4	70.5
Seg-Graph (Wang and Lu, 2018)	76.8	72.3	74.5
ARN (Lin et al., 2019a)	76.2	73.6	74.9
Path-BERT (Shibuya and Hovy, 2019)	82.98	82.42	82.70
Merge-BERT (Fisher and Vlachos, 2019)	82.7	82.1	82.4
DYGIE (Luan et al., 2019)	-	-	82.9
Seq2seq-BERT (Straková et al., 2019)	-	-	84.33
BERT-MRC	**87.16**	**86.59**	**86.88** **(+2.55)**
English GENIA			
Model	P	R	F
Hyper-Graph (Katiyar and Cardie, 2018)	77.7	71.8	74.6
ARN (Lin et al., 2019a)	75.8	73.9	74.8
Path-BERT (Shibuya and Hovy, 2019)	78.07	76.45	77.25
DYGIE (Luan et al., 2019)	-	-	76.2
Seq2seq-BERT (Straková et al., 2019)	-	-	78.31
BERT-MRC	**85.18**	**81.12**	**83.75** **(+5.44)**
English KBP 2017			
Model	P	R	F
KBP17-Best (Ji et al., 2017)	76.2	73.0	72.8
ARN (Lin et al., 2019a)	77.7	71.8	74.6
BERT-MRC	**82.33**	**77.61**	**80.97** **(+6.37)**

Table 2: Results for *nested* NER tasks.

CoNLL2003 (Sang and Meulder, 2003) is an English dataset with four types of named entities: Location, Organization, Person and Miscellaneous. We followed data processing protocols in Ma and Hovy (2016).

OntoNotes 5.0 (Pradhan et al., 2013) is an English dataset and consists of text from a wide variety of sources. The dataset includes 18 types of named entity, consisting of 11 types (Person, Organization, etc) and 7 values (Date, Percent, etc).

MSRA (Levow, 2006) is a Chinese dataset and performs as a benchmark dataset. Data in MSRA is collected from news domain and is used as shared task on SIGNAN backoff 2006. There are three types of named entities.

OntoNotes 4.0 (Pradhan et al., 2011) is a Chinese dataset and consists of text from news domain. OntoNotes 4.0 annotates 18 named entity types. In this paper, we take the same data split as Wu et al. (2019).

3.6.2 Baselines

For English datasets, we use the following models as baselines.

- **BiLSTM-CRF** from Ma and Hovy (2016).
- **ELMo** tagging model from Peters et al. (2018b).
- **CVT** from Clark et al. (2018), which uses Cross-View Training(CVT) to improve the representations of a Bi-LSTM encoder.
- **Bert-Tagger** from Devlin et al. (2018), which treats NER as a tagging task.

For Chinese datasets, we use the following models as baselines:

- **Lattice-LSTM:** Zhang and Yang (2018) constructs a word-character lattice.
- **Bert-Tagger:** Devlin et al. (2018) treats NER as a tagging task.
- **Glyce-BERT:** The current SOTA model in Chinese NER developed by Wu et al. (2019), which combines glyph information with BERT pretraining.

English CoNLL 2003			
Model	P	R	F
BiLSTM-CRF (Ma and Hovy, 2016)	-	-	91.03
ELMo (Peters et al., 2018b)	-	-	92.22
CVT (Clark et al., 2018)	-	-	92.6
BERT-Tagger (Devlin et al., 2018)	-	-	92.8
BERT-MRC	**92.33**	**94.61**	**93.04** **(+0.24)**
English OntoNotes 5.0			
Model	P	R	F
BiLSTM-CRF (Ma and Hovy, 2016)	86.04	86.53	86.28
Strubell et al. (2017)	-	-	86.84
CVT (Clark et al., 2018)	-	-	88.8
BERT-Tagger (Devlin et al., 2018)	90.01	88.35	89.16
BERT-MRC	**92.98**	**89.95**	**91.11** **(+1.95)**
Chinese MSRA			
Model	P	R	F
Lattice-LSTM (Zhang and Yang, 2018)	93.57	92.79	93.18
BERT-Tagger (Devlin et al., 2018)	94.97	94.62	94.80
Glyce-BERT (Wu et al., 2019)	95.57	95.51	95.54
BERT-MRC	**96.18**	**95.12**	**95.75** **(+0.21)**
Chinese OntoNotes 4.0			
Model	P	R	F
Lattice-LSTM (Zhang and Yang, 2018)	76.35	71.56	73.88
BERT-Tagger (Devlin et al., 2018)	78.01	80.35	79.16
Glyce-BERT (Wu et al., 2019)	81.87	81.40	80.62
BERT-MRC	**82.98**	**81.25**	**82.11** **(+1.49)**

Table 3: Results for *flat* NER tasks.

3.6.3 Results and Discussions

Table 3 presents comparisons between the proposed model and baseline models. For English CoNLL 2003, our model outperforms the fine-tuned BERT tagging model by +0.24% in terms of F1, while for English OntoNotes 5.0, the proposed model achieves a huge gain of 1.95% im-

English OntoNotes 5.0	
Model	F
LSTM tagger (Strubell et al., 2017)	86.84
BiDAF (Seo et al., 2017)	87.39 (+0.55)
QAnet (Yu et al., 2018)	87.98 (+1.14)
BERT-Tagger	89.16
BERT-MRC	**91.11** (+1.95)

Table 4: Results of different MRC models on English OntoNotes5.0

English OntoNotes 5.0	
Model	F1-score
BERT-Tagger	89.16
Position index of labels	88.29 (-0.87)
Keywords	89.74 (+0.58)
Wikipedia	89.66 (+0.59)
Rule-based template filling	89.30 (+0.14)
Synonyms	89.92 (+0.76)
Keywords+Synonyms	90.23 (+1.07)
Annotation guideline notes	91.11 (+1.95)

Table 5: Results of different types of queries.

provement. The reason why more significant performance boost is observed for OntoNotes is that OntoNotes contains more types of entities than CoNLL03 (18 vs 4), and some entity categories face the more severe data sparsity problem. Since the query encodes significant prior knowledge for the entity type to extract, the MRC formulation is more immune to the tag sparsity issue, leading to more improvements in the OntoNotes dataset. The proposed method also achieves new SOTA on Chinese datasets. For Chinese MSRA, the proposed method outperforms the fine-tuned BERT tagging model by +0.95% in terms of F1. We also improve the F1 from 79.16% to 82.11% for Chinese OntoNotes4.0.

4 Ablation study

4.1 Improvement from MRC or from BERT

For flat NER, it is not immediately clear which is responsible for what proportion of the improvement, the MRC formulation or BERT (Devlin et al., 2018). On one hand, the MRC formulation facilitates the entity extraction process by encoding prior knowledge in the query; on the other hand, the good performance might also come from the large-scale pre-training in BERT.

To separate the influence from large-scale BERT pretraining, we compare the LSTM-CRF tagging model (Strubell et al., 2017) with other MRC based models such as QAnet (Yu et al., 2018) and BiDAF (Seo et al., 2017), which do not rely on large-scale pretraining. Results on English Ontonotes are shown in Table 5. As can be seen, though underperforming BERT-Tagger, the MRC based approaches QAnet and BiDAF still significantly outperform tagging models based on LSTM+CRF. This validates the importance of MRC formulation. The MRC formulation's benefits are also verified when comparing BERT-tagger with BERT-MRC: the latter outperforms the former by +1.95.

We plot the attention matrices output from the BiDAF model between the query and the context sentence in Figure 2. As can be seen, the semantic similarity between tagging classes and the contexts are able to be captured in the attention matrix. In the examples, *Flevland* matches *geographical*, *cities* and *state*.

4.2 How to construct queries

How to construct query has a significant influence on the final results. In this subsection, we explore different ways to construct queries and their influence, including:

- **Position index of labels** a query is constructed using the index of a tag to , i.e., "one", "two", "three".
- **Keyword** a query is the keyword describing the tag, e.g., the question query for tag ORG is *"organization"*.
- **Rule-based template filling** generates questions using templates. The query for tag ORG is *"which organization is mentioned in the text"*.
- **Wikipedia** a query is constructed using its wikipedia definition. The query for tag ORG is *"an organization is an entity comprising multiple people, such as an institution or an association."*
- **Synonyms** are words or phrases that mean exactly or nearly the same as the original keyword extracted using the Oxford Dictionary. The query for tag ORG is *"association"*.
- **Keyword+Synonyms** the concatenation of a keyword and its synonym.
- **Annotation guideline notes** is the method we use in this paper. The query for tag ORG is *"find organizations including companies, agencies and institutions"*.

Table 5 shows the experimental results on English OntoNotes 5.0. The BERT-MRC outperforms BERT-Tagger in all settings except *Position Index of Labels*. The model trained with the

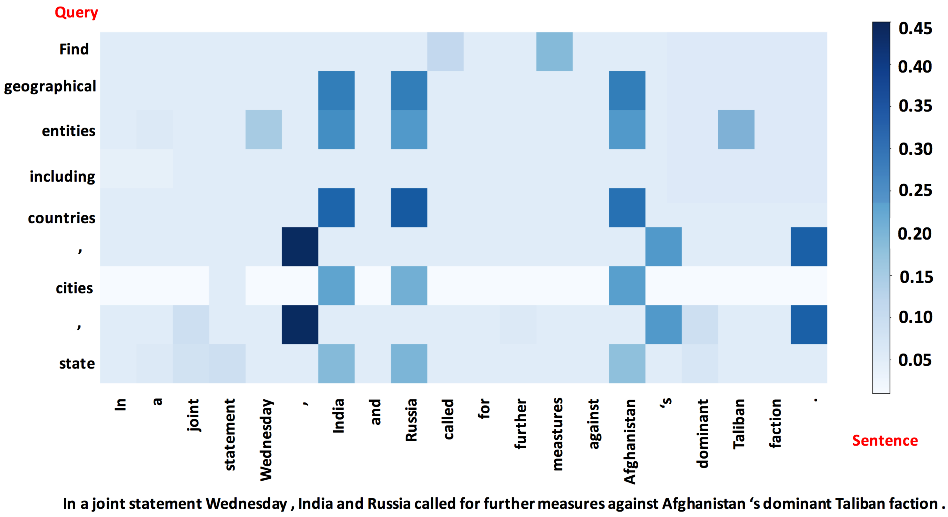

Figure 2: An example of attention matrices between the query and the input sentence.

Models	Train	Test	F1-score
BERT-tagger	OntoNotes5.0	OntoNotes5.0	89.16
BERT-MRC	OntoNotes5.0	OntoNotes5.0	91.11
BERT-tagger	CoNLL03	OntoNotes5.0	31.87
BERT-MRC	CoNLL03	OntoNotes5.0	72.34

Table 6: Zero-shot evaluation on OntoNotes5.0

Annotation Guideline Notes achieves the highest F1 score. Explanations are as follows: for *Position Index Dataset*, queries are constructed using tag indexes and thus do not contain any meaningful information, leading to inferior performances; *Wikipedia* underperforms *Annotation Guideline Notes* because definitions from Wikipedia are relatively general and may not precisely describe the categories in a way tailored to data annotations.

4.3 Zero-shot Evaluation on Unseen Labels

It would be interesting to test how well a model trained on one dataset is transferable to another, which is referred to as the zero-shot learning ability. We trained models on CoNLL 2003 and test them on OntoNotes5.0. OntoNotes5.0 contains 18 entity types, 3 shared with CoNLL03, and 15 unseen in CoNLL03. Table 6 presents the results. As can been seen, BERT-tagger does not have zero-shot learning ability, only obtaining an accuracy of 31.87. This is in line with our expectation since it cannot predict labels unseen from the training set. The question-answering formalization in MRC framework, which predicts the answer to the given query comes with more generalization capability, achieving acceptable results.

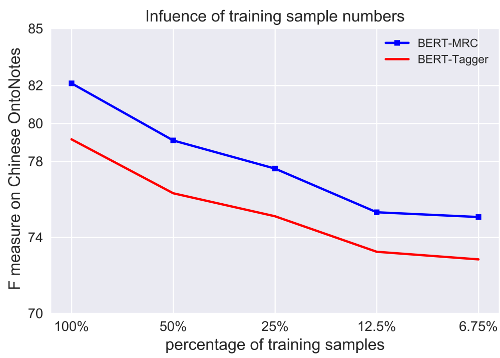

Figure 3: Model performance decreases with less training data.

4.4 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. Figure 3 verifies this point: on the Chinese OntoNotes 4.0 training set, the query-based BERT-MRC approach achieves comparable performance to BERT-tagger even with half amount of training data.

5 Conclusion

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

References

Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 65–72. Association for Computational Linguistics.

Kate Byrne. 2007. Nested named entity recognition in historical archive text. In *International Conference on Semantic Computing (ICSC 2007)*, pages 589–596. IEEE.

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. 2015. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*.

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

Julie Medero Christopher Walker, Stephanie Strassel and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia 57*.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1914–1925.

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, Stephanie M Strassel Lance A Ramshaw, and Ralph M Weischedel. 2005. The automatic content extraction (ace) program-tasks, data, and evaluation. *In LREC*, 2:1.

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.

Joseph Fisher and Andreas Vlachos. 2019. Merge and label: A novel neural network architecture for nested NER. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5840–5850.

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.

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

Heng Ji, Xiaoman Pan, Boliang Zhang, Joel Nothman, James Mayfield, Paul McNamee, and Cash Costello. 2017. Overview of TAC-KBP2017 13 languages entity discovery and linking. In *Proceedings of the 2017 Text Analysis Conference, TAC 2017, Gaithersburg, Maryland, USA, November 13-14, 2017*.

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 SIGHAN Workshop on Chinese Language Processing*, pages 108–117, Sydney, Australia. Association for Computational Linguistics.

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

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1340–1350.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019a. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5182–5192.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019b. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5182–5192.

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.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3036–3046.

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*.

Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. 2002. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the Second International Conference on Human Language Technology Research*, HLT '02, pages 82–86, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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

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

Sameer Pradhan, Mitchell P. Marcus, Martha Palmer, Lance A. Ramshaw, Ralph M. Weischedel, and Nianwen Xue, editors. 2011. *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, CoNLL 2011, Portland, Oregon, USA, June 23-24, 2011*. ACL.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147.

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

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

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.

Takashi Shibuya and Eduard H. Hovy. 2019. Nested named entity recognition via second-best sequence learning and decoding. *CoRR*, abs/1909.02250.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. *arXiv preprint arXiv:1702.02098*.

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*.

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*.

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.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

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