A Multi-task Approach for Machine Reading Comprehension Form Named Entity Recognition Tasks

Yu Zhang[1*], Jian Deng[2], Ying Ma[2], and Jianmin Li[3]

[1,2,3]*College of Computer and Information Engineering, Xiamen University of Technology, No.600 Ligong Road, Jimei District, Xiamen, 361024, Fujian Province, China*
[*]*Corresponding author. Email: zhang519904120@gmail.com*
zhang519904120@gmail.com, dengjian@xmut.edu.cn, maying@xmut.edu.cn, lijm@xmut.edu.cn

Abstract:
Named Entity Recognition (NER) is a basic NLP task that aims to provide class labels for words in free text, such as people, locations. The traditional NER task is treated as a label-sequence task, but the trend of recent jobs is to convert the NER task into a Machine Reading Comprehension (MRC) task to achieve better representation. However, this conversion is often accompanied by the problem of poor generalization of the model due to too few class-specific instances. Therefore, to solve this problem, we try to introduce different domain knowledge into our NER task, and we introduce MRC knowledge as well as NLI knowledge into our NER task through a multi-task learning approach. Our method is a one-stage model that combines two large NLI datasets (MNLI, SNLI) and a large traditional MRC dataset (SquAD) with our target NER dataset for multi-task learning. Through multi task learning, we learn the knowledge of NLI domain and MRC domain, so as to improve the performance of our model on the target dataset. We conducted enough experiments to validate the effectiveness of our method. Also, our model achieves 0.3% and 0.106% improvement compared to different baselines, respectively, proving that introducing external knowledge is effective in improving model performance.

Keywords: *Named Entity Recognition, Multi-task Learning, Natural Language Inference*

1 INTRODUCTION

NER is currently a popular work in the field of natural language processing, which aims to identify special objects from text whose semantic categories are usually predefined before recognition, with predefined categories such as people, addresses, organizations, etc. The traditional form of the task of NER is sequence annotation, a generalization of the classification problem, which requires classifying each basic token in the input sequence. Because of the relationship between the NER task and the MRC task, several works in recent years have treated NER as an MRC task rather than a sequence labelling problem. For example, we can treat the entity type as the root of the problem, the sequence corresponding to the entity as the answer, and the sentence in which the entity is located as the context, thus forming a typical Span-based MRC task question-answer pair.

However, this conversion has several problems. The root of this conversion is to define the problem by entity types. The number of entity types in the dataset will affect the overall data balance problem, and the entity types in the dataset will directly affect the generalization of the model in that category. As a result, the model will perform poorly in a specific class of problems. At the same time, comparing with the traditional MRC dataset, we can find that the whole dataset has a natural disadvantage in the length of the context. The context in the traditional MRC dataset is mostly composed of multiple long sentences in one passage, which has a strong enough corpus. For the NER dataset, the context content is often only one sentence after conversion to MRC data, and there is not enough corpus to provide learning. Therefore, we decided to solve these problems by introducing multi-task learning. We learned the target NER dataset together with the traditional MRC dataset (we chose to use SquAD as the MRC dataset to do multi-task learning) to obtain a model with stronger generalization ability. Also inspired by NLI-QA, we learned that the NLI task can be significantly improved in learning with the MRC task because NLI essentially requires that the premises (document context) contain all

B. Fox et al. (Eds.): IC-ICAIE 2022, AHCS 9, pp. 480-485, 2023.
https://doi.org/10.2991/978-94-6463-040-4_73

the necessary information to support the hypothesis (proposed answer to the question). So, we likewise incorporate the NLI task into multitask learning to improve the effectiveness of our model.

This paper analyses and discusses the work related to our training network in the Section 2. Section 3 describes the methodology and the network structure in detail. Section 4 gives the results and analysis. Section 5 summarizes this work.

2 RELATED WORKs

2.1 Natural Language Inference

Significant progress has been made in natural language inference (NLI) tasks through the development of large-scale datasets such as SNLI and MNLI. If we can generate vectors with good sentence representations by NLI tasks, then we can easily transfer the results of this part to other tasks. e-SNLI [3] is a task that extends SNLI, adds a layer of human-annotated natural language interpretation to explain the entailments between sentences, demonstrating how the interpreted corpus of e-NLI can be used for a variety of purposes, providing a basis for transfer of the NLI task to out-of-domain datasets. [10] performs multi-hop QA tasks with its unique architecture, Multee, which consists of a local module that aids in the location of key sentences and a global module that aggregates data by efficiently incorporating importance weights. [7] demonstrates that the connection of typological traits between languages can help to understand when sharing parameters obtained through meta-learning is advantageous. And work like SciTail [5] was born out of a science QA mission to promote models that have a direct impact on quality assurance. On the Aristo Reasoning Challenge, entailment models trained on this dataset exhibit minor improvements.

2.2 Multi-task Learning

Multitask learning is defined as a machine learning approach based on shared representation, where multiple related tasks are learned together, aiming to solve several different tasks simultaneously by exploiting the similarities between different tasks. Nowadays, many jobs try to use multitask learning to let their target models learn enough domain knowledge to improve the generalization ability of the models. Works like [1] propose an unique multi-task strategy that combines the primary goal of fine-grained Named Entity categorization with a more general secondary task of NE segmentation. Also, [12] proposes a unique deep neural multi-task learning framework with explicit feedback techniques for modelling recognition and normalization simultaneously. These papers have attempted to improve the generalization of their models via multi-task learning, resulting in more robust models

.

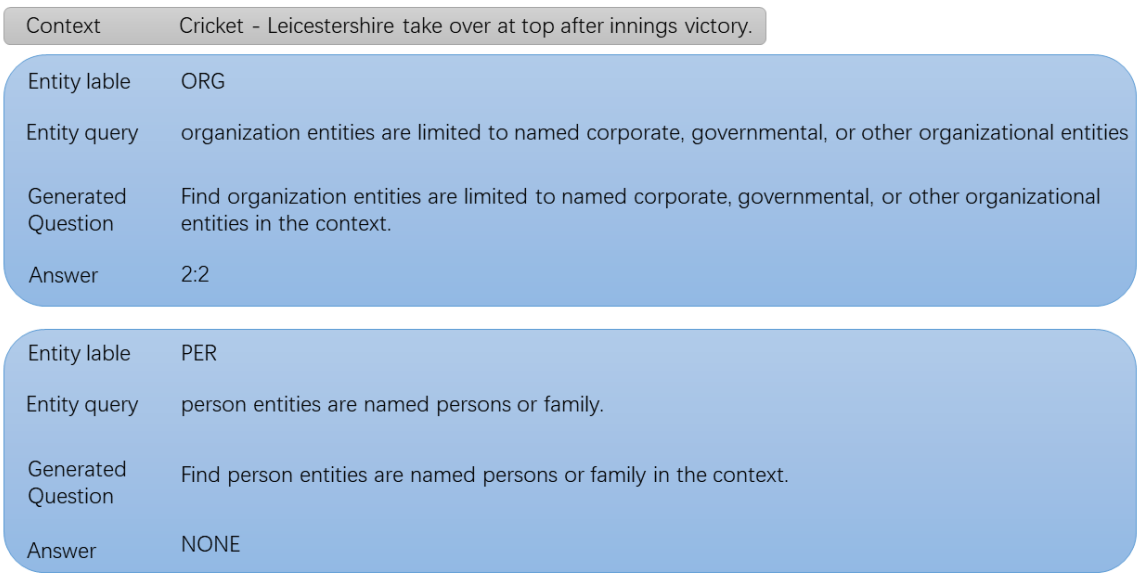

Figure 1: The transformation of traditional NER dataset to MRC dataset, which use entity query to generate question and the word of entity is the answer.

However, none of these works were applied to the transformed form of the dataset, so we used multitask learning on the NER task converted to MRC form to try to get a more expressive model.

3 METHOD

In this section we mainly show the settings and the structure of our model.

3.1 Training Settings

Our experiments use BERT-Large as our base model. We first identified that we used four different datasets from two tasks for multi-task learning, MNLI [11] and SNLI [2]. for the NLI task, and then SquAD [8] for the MRC task with the NER target dataset transformed into MRC form. The target NER dataset we choose is CoNLL-2003, a classic NER dataset. We subjected these four datasets to multitask learning to learn the common task representation, expecting to explore the common representation features of these two tasks by sharing parameters through multitask learning.

During the training process, our model will extract b instances from each of the different datasets and we used a proportional sampling technique, in which the likelihood of sampling a task is proportional to the size of each dataset relative to the total size of all datasets.

And for the transformation of our dataset, the generation of our question comes to be a difficult part of it. Based on the experience from previous work, we mainly use annotations to generate questions for each of our quiz pairs. We can see from Figure 1 that the data makers in the dataset have annotated the types of entities in the dataset, e.g., in the CoNLL2003 dataset, the annotation of the "PER" entity is "person entities are named persons or family, which our problem can be generated as "Find the person entities are named persons or family in the text.".

Once we have all our data formats, the next step is to feed the data into the model. The next section gives a general overview of our model and describes the pipeline of our work.

3.2 Training Structure

Figure 2 shows the overall structure of our model, we take four datasets (MNLI, SNLI, SquAD, Target

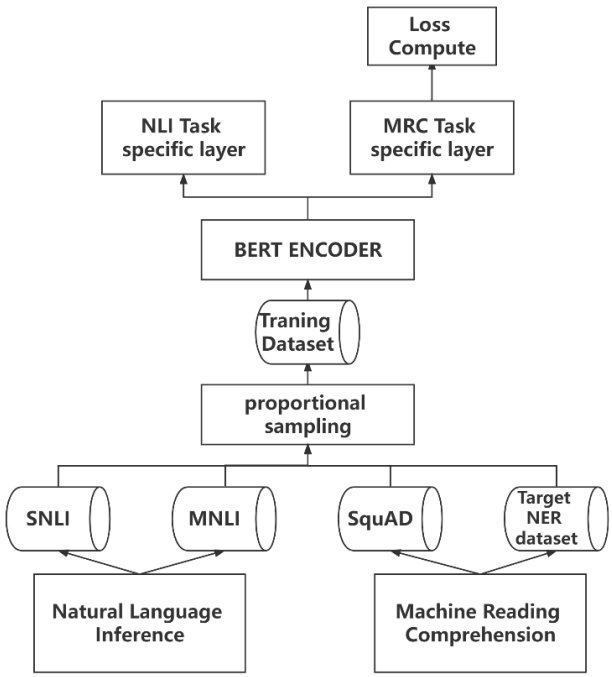

Figure 2: The whole structure of our method, which the BERT-ENCODER we use is BERT-Large. The target NER dataset we mainly use here is CoNLL-2003.

MRC-form NER dataset) from two different tasks (Natural Language Inference, Machine Reading Comprehension) for multi-task learning and sample each dataset according to the sampling method described in 3.1, and finally a total of b data are selected to enter the model training until all instances are trained then stop. The NLI task and the MRC task have different approaches to the details within the model.

For the NLI task, we use the original way of training in BERT for the classification task, and put the obtained sentences into BERT as the initial sequence to get the label of the corresponding sentences.

And for the MRC task, which contains the NER task in MRC-form, we take a different approach and the whole pipeline can be described as follow:

$$\{[CLS], [q_1], [q_2], \dots, [q_m], [SEP], [x_1], [x_2], \dots, [x_n]\}$$

where CLS and SEP are special token, representing start token and separate token respectively, we put this sequence into BERT to get a matrix E.

After obtaining the matrix E of the whole sequence from BERT, we can use this matrix to calculate the probability of all tokens as the starting token of the answer, which can be:

$$P_{start} = softmax(E \cdot T_{start}) \quad (1)$$

Where P_{start} is the probability distribution of each index being the start position of an entity given the query, and $T_{start}$is the weights to learn. P_{end} is calculated by the same method.

The next job for us is to match up the start index to the end index. But at the same time, we need to consider the problem of nested NER which indicates that the start-index prediction model might predict numerous start indexes, and the end-index prediction model could predict many end indices. As a result, we'll also require a mechanism for matching a predicted start index to its associated end index.

We take argmax for each row of P_{start} and P_{end} to get I_{start} and I_{end}, which represent two 0-1 sequences of length n as the probability of start token or end token in this sequence, respectively. For Flat NER entities, we usually just need to do a matchup of I_{start} and I_{end} to get the answer predicted by our model. However, for Nested NER, this is not possible. The traditional approach is to combine the start token with the nearest end token, but since Nested NER will get multiple answers from different entities, if we simply choose the nearest token, the correct rate will be significantly reduced. So, we need to train a binary classification model to determine the matching predicted values in I_{start} and I_{end} as the entity answer representation, which can be expressed as the following equation:

$$P_{i_{start}, j_{end}} = sigmoid(m \cdot concat(E_{i_{start}}, E_{j_{end}})) \quad (2)$$

where m represents the weights of our model.

After obtaining all the predicted values P_{start}, P_{end} and $P_{start,end}$, we use Cross-Entropy(CE) loss of the start token and end token and the span-answer pair with all three in the training sample Y_{start}, Y_{end} and $Y_{start,end}$, then the final cross-entropy loss of our model can be calculated as:

$$Loss_{strat} = CE(P_{start}, Y_{start}) \quad (3)$$

$$Loss_{strat} = CE(P_{start}, Y_{start}) \quad (4)$$

$$Loss_{end} = CE(P_{end}, Y_{end}) \quad (5)$$

$$Loss_{span} = CE(P_{start,end}, Y_{start,end}) \quad (6)$$

$$Loss = \alpha Loss_{strat} + \beta Loss_{end} + \gamma Loss_{span} \quad (7)$$

where α, β, γ are hyper-parameters controls the contributions of each loss for the whole structure.

4 EXPERIMENTS

The section shows the specific settings used for our experiments with the results we obtained, and provides some level of analysis of the results obtained.

4.1 Baseline

Two baselines were chosen for our article, which are:

- BERT-TAGGER from [4]. Bidirectional Encoder Representation from Transformers, is a language representation model that has been pre-trained. It stresses the use of the novel masked language model (MLM) for pre-training, rather than the classic unidirectional language model or shallow splicing of two unidirectional language models, to build deep bidirectional language representations;
- BERT-MRC from [6]. BERT-MRC model is a model in the field of entity recognition, which is more effective than other models in the case of small data volume. The reason is that BERT-MRC model can add some prior knowledge through the problem to reduce the problem caused by too small data volume.

4.2 Dataset

Two baselines were chosen for our article, which are:

- MNLI from [11]. The Multi-Genre Natural Language Inference Corpus (MNLI) is a crowdsourced collection of textual entailment annotations of phrase pairs for natural language inference. The aim of the task is to anticipate if a premise statement contains a hypothesis (entailment), contradicts the hypothesis (contradiction), or neither (neutral);
- SNLI from [2]. The SNLI corpus is a collection of 570k human-written English phrase pairs that have been manually annotated for balanced categorization using enume, contradictory, and neutral tags, allowing natural language inference to be supported (NLI). It will serve as a standard for evaluating textual representation systems, particularly those produced by representation

learning approaches, as well as a resource for constructing any type of NLP model

- SquAD from [8]. SQuAD (Stanford Question Answering Dataset) is a reading comprehension dataset in which a model answers questions using data from a text from which all answers are obtained. Not only does the model need to be able to discover the answer to the question from the relevant text, but it also needs to be able to choose none rather than guessing blindly when there is no comparable answer.
- CoNLL-2003 from [9]. CoNLL-2003 is the most common publicly available dataset of named entities. It's created by IPS articles in four different languages (Spanish, Dutch, English and German) and focuses on 4 entities: PER (Person), LOC (Location), ORG (Organization) and MISC (Other, including all other types of entities). In our work we mainly use the English dataset.

4.3 Result

From Table 1, we can see that our model achieves good results, our method outperforms the two comparison baselines by 0.3 and 0.106 percentage points respectively. We attribute the growth of our results to our multi-task learning, through which we introduced the NLI task as well as the traditional MRC task, and by learning with these two tasks our target NER dataset learned about these two domains, compensating for the weak generalization ability caused by the small number of specific samples in the original NER task.

Table 1: Result of our method on CoNLL-2003, which we mainly take F1-score as our benchmark.

Method Name	F1-score
BERT-TAGGER	92.8
BERT-MRC	93.04
OUR-Method	93.146

Also why the NLI task can be helpful for the target task is analyzed. The basic goal of NLI is to establish a semantic relationship between two sentences (Premise, Hypothesis) or two words so that the model can concentrate on semantic understanding. And we can concidered that the correct answer is an entailment to the evidence sentence, whereas the wrong answer is not, and our answer is simultaneously part of the whole sentence (Span-based QA). We therefore believe that the ability of linguistic inference enables the model to obtain the correct predicted answer from the sentence. So we can apply the inference learning capability of the NLI task to the Span-based QA problem. Our application method here is to combine the NLI task with the MRC task for multi-task learning to facilitate the model to apply the extra-domain knowledge learned in the NLI task to the MRC task, and to enhance the expressive power of the MRC task, we also introduce the traditional large MRC dataset SquAD to enhance the model's intra-domain knowledge, and the two are combined to enhance the overall model's expressiveness to better solve the problem in our target dataset.

5 CONCLUSIONS

We propose in this paper to introduce NLI knowledge into the NER task converted to MRC data form by multi-task learning, and to introduce large in-domain MRC datasets (SquAD) to bring in in-domain knowledge into the NER task in MRC data form by the same multi-task learning approach. Our approach introduces both extra-domain and intra-domain knowledge into the target dataset by multi-task learning, thus improving the performance of the target dataset. Our experiments demonstrate the effectiveness of our approach, and we try to give a preliminary conclusion to explain why our approach can achieve certain results. The approach we adopt here to introduce an out-of-domain in-domain dataset for multi-task learning can be generalized to various tasks and is not limited to solving the NER model.

REFERENCES

[1] Aguilar G, Maharjan S, López-Monroy A P, et al. A multi-task approach for named entity recognition in social media data [J]. arXiv preprint arXiv:1906.04135, 2019.

[2] Bowman S R, Angeli G, Potts C, et al. The SNLI corpus[J]. 2015.

[3] Camburu O M, Rocktäschel T, Lukasiewicz T, et al. e-snli: Natural language inference with natural language explanations [J]. Advances in Neural Information Processing Systems, 2018, 31.

[4] Delvin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding [J]. arXiv preprint arXiv:1810.04805, 2018.

[5] Khot T, Sabharwal A, Clark P. Scitail: A textual entailment dataset from science question answering[C]//Thirty-Second AAAI Conference on Artificial Intelligence. 2018.

[6] Li X, Feng J, Meng Y, et al. A unified MRC framework for named entity recognition[J]. arXiv preprint arXiv:1910.11476, 2019.

[7] Nooralahzadeh F, Bekoulis G, Bjerva J, et al. Zero-shot cross-lingual transfer with meta learning[J]. arXiv preprint arXiv:2003.02739, 2020.

[8] Rajpurkar P, Jia R, Liang P. Know what you don't know: Unanswerable questions for SQuAD[J]. arXiv preprint arXiv:1806.03822, 2018.

[9] Sang E F, De Meulder F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition[J]. arXiv preprint cs/0306050, 2003.

[10] Trivedi H, Kwon H, Khot T, et al. Repurposing entailment for multi-hop question answering tasks[J]. arXiv preprint arXiv:1904.09380, 2019.

[11] Wang A, Singh A, Michael J, et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding[J]. arXiv preprint arXiv:1804.07461, 2018.

[12] Zhao S, Liu T, Zhao S, et al. A neural multi-task learning framework to jointly model medical named entity recognition and normalization[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 817-824.