General Named Entity Recognition: Benchmark Between Advanced And Vanilla Architecture

Junyi Guo
Department of Biostatistics
Harvard University
Boston, MA 02115
junyiguo@hsph.harvard.edu

Yuxin Xu
Department of Biostatistics
Harvard University
Boston, MA 02115
yuxinxu@hsph.harvard.edu

Named entity recognition (NER) describes the task of identifying if proper nouns that represent something or someone [10] in the corpora belong to generic name entity classes like proper name and location, or domain-specific name entity classes like enzyme, that are useful information to human. NER problems are first brought to people's attention at the sixth Message Understanding Conference (MUC-6) in 1996 [5] under the increasing need of using machines to extract information from text. Named entity recognition serves not only as a significant method of information extraction, but also as the foundation of other natural language processing topics of great interests, such as text summarization and knowledge-base construction [7].

Traditionally, NER projects are done using rule-based extraction, unsupervised clustering and feature-based supervised learning. Due to the rise of neural networks in the recent decade, using deep learning on NER problems becomes increasingly popular. Past work in this field involves word-level representation, character-level representation and hybrid representation using convolutional neural network [3], recurrent neural network [6], recursive neural network [8], neural language model [11] and transformer [9].

Electronic Health Record has been one of the major contributing factors to physicians heavy workload. In a survey [4] conducted on "over 3,700 physicians in nearly every specialty, work setting and region", the most popular opinion physicians have towards EHR is complaint on the extra workload EHR brings about. Since both of us come from a health data science background and hope to apply machine learning to the field of healthcare and medical care, constructing an artificial intelligence that will automatically recognize the speech, summarize the speech text, and put relevant key words in patient's EHR could possibly save a lot of time for the physicians and streamline their work. However, since none of us is experienced enough to do speech recognition or text summarization, and since this class is a machine learning class, we decide to focus on named entity recognition for this project as a building block for the future.

In this project, we aim to investigate how machine can learn not only the meaning of words but also their detailed contextual information in the entire sentence. We want to build and compare different models that can properly tag the information of each word in a given sentence. In order to achieve our goal, we need a carefully tagged text dataset and some NLP-suitable architectures for the training purposes.

We will use the Annotated Corpus for Named Entity Recognition dataset we found on Kaggle [1]. This dataset is extracted from the Groningen Meaning Bank (GMB) dataset, and contains taggings for words in multiple sentences. The entire dataset contains 1048575 word entries with tagging

Preprint. Under review.

information and 47959 different sentences in total. The first column represents the sentence id, the second column represents the specific words, the third column represents the part-of-speech tagging, and the forth column represents the detailed information about entities, for example geographical entity and time indicator. In this project, we are going to use the last column as our target variable and tune our model to be able to learn detailed information about entities including location information and person information.

For model constructions, we would like to build and compare three different model with different complexity and investigate their performances. The first model we are going to construct will be a vanilla Bidirectional LSTM model. LSTM model is one of the most commonly used RNN architectures that is suitable for learning language and temporal information, and bidirectional LSTM enhanced the model ability in learning the order and contextual information. Thus, we would like to use the Bidirectional LSTM model as our baseline model for our Named Entity Recognition task.

In addition to our baseline model,we would like to use another model that is more advance and provides better capability in language learning. Transformers came to our mind. Transformers-based models are well-know for their capability in parallelization as well as outstanding performances in multiple Natrual Language Processing tasks. Thus, we decided to train a BERT model for our particular task. However, training a transformers-based model from scratch is time-consuming and computationally intensive. Thus, we decide to utilize a pre-trained BERT model in our task. For the model training, We plan to follow the guidance provided by Sterbak in his web post Named entity recognition with Bert [12]. We will first follow the guidance for prepossessing our tagged data in order to prepare the input for BERT model. Then, we planned to apply BERT tokenizer as well as BERT model pre-trained on token classification task from Huggingface.

Finally, we would like to challenge ourselves by including a state-of-the-art model that is designed and fine-tuned with the Named Entity Recognition task. We found several state-of-the-art NER models from the Named Entity Recognition leaderboard on CoNLL 2003 (English) dataset [2], which is one of the most standard and commonly used NER dataset. We would like to pick the LUKE model or the ACE model from the leaderboard, since those models achieve the top performances in the CoNLL 2003 dataset. According to Yamada et al., LUKE is an advance model with pretrained contextualized representations of words and entities based on the bidirectional transformer [14]. According to Wang et al., ACE Automated Concatenation of Embeddings (ACE) helps automate the process of finding better concatenations of embeddings for structured prediction tasks [13]. Both architectures tries to improve the final prediction results by finding and applying word embeddings and feature representations for the text data we hope these state-of-the-art models can help us achieve better results. We will use the pre-trained model provided on the Github pages of corresponding papers and try to fine-tune the model on our selected dataset.

We will need GPU power to train and fine-tune our models, since training NLP models can take up a lot of time using CPU. Our first choice of coding platform will be Google Colab, since Colab grants us some free access to GPU resources.

There are two types of evaluation for named entity recognition problems: exact-match evaluation and relaxed-match evaluation. On top of giving the named entity the correct class, exact-match evaluation requires the model to detect the correct boundary for named entities to be considered as successful, whereas relaxed-match evaluation only requires overlap between the correct named entity boundary and the predicted named entity boundary. In this project, we will evaluate the outcome of out model using both evaluations in our project to reflect how well the model performs in terms of detecting the correct boundary for named entities and categorizing them in the correct classes respectively. Under each of the two kinds of evaluation, we will provide commonly-used metrics such as accuracy, macro-averaged F-score and micro-averaged F-score.

Our project can be divided to several stages. In the first stage, we will focus on preprocessing our dataset including cleaning and tokenizing the textual data. We plan to finish this part by the beginning of April. The second stage would be applying and fine-tuning the NLP model. We expect this stage to be time-consuming and we plan to finish this part by the beginning of May. Then, we are going to spend the next few days analyzing the result and preparing for the final delivery. During the project, we anticipate that the most difficult parts would be text processing, tagging data that are suitable for our model and fine-tuning our model. We will try to resolved those potential challenges by finding previous works and papers targeting similar tasks.

References

[1] Annotated corpus for named entity recognition. URL https://www.kaggle.com/abhinavwalia95/entity-annotated-corpus#ner_dataset.csv.

[2] Named entity recognition on conll 2003 (english). URL https://paperswithcode.com/sota/named-entity-recognition-ner-on-conll-2003.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch, 2011.

[4] C. Gerry. Physician workload survey 2018, 2018. URL https://locumstory.com/spotlight/physician-workload-survey-2018/.

[5] R. Grishman and B. Sundheim. Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, page 466–471, USA, 1996. Association for Computational Linguistics. doi: 10.3115/992628.992709. URL https://doi.org/10.3115/992628.992709.

[6] Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging, 2015.

[7] J. Li, A. Sun, J. Han, and C. Li. A survey on deep learning for named entity recognition, 2020.

[8] P.-H. Li, R.-P. Dong, Y.-S. Wang, J.-C. Chou, and W.-Y. Ma. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2664–2669, Copenhagen, Denmark, Sept. 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1282. URL https://www.aclweb.org/anthology/D17-1282.

[9] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li. Dice loss for data-imbalanced nlp tasks, 2020.

[10] G. Petasis, A. Cucchiarelli, P. Velardi, G. Paliouras, V. Karkaletsis, and C. D. Spyropoulos. Automatic adaptation of proper noun dictionaries through cooperation of machine learning and probabilistic methods. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 128–135, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132263. doi: 10.1145/345508.345563. URL https://doi.org/10.1145/345508.345563.

[11] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power. Semi-supervised sequence tagging with bidirectional language models, 2017.

[12] T. Sterbak. Named entity recognition with bert. URL https://www.depends-on-the-definition.com/named-entity-recognition-with-bert/.

[13] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, and K. Tu. Automated concatenation of embeddings for structured prediction. *arXiv preprint arXiv:2010.05006*, 2020.

[14] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto. Luke: Deep contextualized entity representations with entity-aware self-attention. In *EMNLP*, 2020.