Named Entity Recognition on Queries

Jianwen Liu
School of Informatics, Computing and Engineering
Indiana University Bloomington
Bloomington, IN 47408, USA
Jl147@iu.edu

Abstract

Query is a common form of text which is widely used. Named entity recognition on query is a hard task. In this paper, we built a word plus char level neural network for named entity recognition on queries to capture the information of both words and characters of queries and compared it with the baseline Bi-LSTM-CRF model. The result showed that word and char NN model does not have an advantage for short queries corpus compared with the enhancement on formal text. The reason is that informal syntax and shortness of queries destructed the sequential pattern in text. The LSTM model of characters brought up more chaos to the word sequence, decreasing the performance of word plus character NN model.

1. Introduction

Named entity recognition is a classical task in Natural language processing (NLP). It is a subtask of information extraction that seeks to locate and classify the named entity mentioned in text. Traditionally, named entity recognition have predefined categories, such as location, person, product and time, etc. Recently, named entity recognition has been a heated topic both in academia and industry.

In Collins dictionary, a query is a question, especially one that you ask an organization, publication, or expert. The most common use of queries is the search query in search. When we asked a question to the search blank, it is a query parsed by the search engine. In the industry, many AI assistants like Siri and Alexa are becoming important product offered by IT firms. A main work of these AI assistant is to deal with queries generated in daily conversations with customers. This requires deep understanding to the queries for machine. Therefore, the understanding of queries has become crucial to the development of AI assistant system (Liu et al 2013). Fortunately, the development of NLP opens a door to the automation processing on queries.

As a classic task of NLP, named entity recognition laid the foundation of information extraction, leading to more efficient language processing application, such as web search. For

instance, for traditional search, if the search engine returns millions of documents to each search query, it will be a very inefficient search. However, with the help of named entity recognition, search engine can pre classify the documents by entities and tags, then return those are most relative with the parsing result of search queries.

Even though most search queries contain named entity (Guo et al, 2009) and theoretically named entity can help speeding up the search process, there are many features for queries that leads problem to practical implementation on named entity recognition on queries. There are mainly two features included, first, queries are short. Compared with other corpus like novel or speech, queries are usually short. They are questions asked under some circumstance instead of long and organized corpus. Another feature of queries is that they are messy or informal. Since queries are just short questions asked by people, they are less prepared. An analogy to query is tweet. There are many misspelling, incorrect capitalization and wrong punctuation in tweets. Similarly, queries also are informal texts. When people propose queries to the search engine or personal assistant, it is obvious that people will not spend much time editing or polishing those queries.

The features of query make applying named entity recognition on it different to on traditional corpus. In this paper, we applied the word and character level named entity recognition on queries and analysis its result. The second section is literature review, the third section is design of experiments and algorithms, the fourth section includes results and analysis and the final section is conclusion.

2. Literature Review

Named entity recognition had been a long discussed NLP task for many years. The term “named entity” dates back the 1990s on Sixth Message Understanding Conference (Grishman and Sundheim, 1996).

Researchers had tried many approaches to experiment named entity recognition on corpus. The first class of approach is supervised learning, for instance, Bikel et al. (1998) firstly applied Hidden Markov Model to for named entity recognition. They even propose a rule to recognize the data for named entity. Zhou and Su (2002) also utilized Hidden Markov Model on named entity recognition and they gave a bigger number of named entities and reach a high model performance. On the bass of Hidden Markov Model, Malouf (2002) used Maximum Entropy to do named entity recognition and compared the results of it with that of Hidden Markov Model. Sekine (1998) experimented on decision trees for named entity recognition on Japanese.

McCallum and Li (2003) utilized conditional random field (CRF) for named entity recognition. Actually, CRF becomes a standard model for the named entity recognition for recent research.

Another approach for named entity recognition is unsupervised learning. Clustering is the main method in the unsupervised learning class. The advantage of clustering on named entity is that we don not need a huge amount of data to train the model.

Inverse document frequency is a frequently used index for infer topics of the document. Zhang and Elhadad (2013) used it combined with syntactical features for Named entity recognition. They applied it on two types of data and it turned out that both corpus have good prediction accuracy. Etzioni et al. (2005) used an unsupervised information oriented approach (mutual information) for named entities classification. They defined a space of features which corresponds to each entity.

The research above-mentioned all relates to traditional corpus. However, the performance for traditional named entity recognition is not convincing on queries (Strauss, 2016). This is because the majority of named entity recognition approach is train on formal text, particularly, most on newswire text. Even though those approach have a good effect on traditional corpus, it is not guaranteed that they will have comparable performance on informal data, such as queries or tweets.

Besides the word level NN model or char level NN model, a newly proposed model in named entity recognition is the word plus char model (Ma and Hovy, 2016). This model tried to capture the sequential information of both word and character level. It turns out that this combined NN model have a better performance on formal text (Chiu and Nichols, 2015). However, applying this model for named entity recognition on short queries is rare. In this paper, we investigated the performance (training complexity and F score) of this model on short queries and compared it to the baseline word level NN model and infer the insight of application of different models on named entity recognition on queries.

3. Design of Experiments and Algorithms

3.1 Data

In this paper, we used the MIT restaurant corpus[1] for dataset. MIT restaurant corpus is collected by MIT Computer Science and Artificial Intelligence Lab, spoken language system (SLS). SLS group collected a bunch of data in which MIT restaurant corpus is included. This

[1] Source: https://groups.csail.mit.edu/sls/downloads/restaurant/restauranttrain.bio

dataset is semantically tagged training and test corpus with BIO format for Named entities label. They are spoken queries and the whole training dataset is 742K.

3.2 Algorithm

The algorithm we proposed to solve the named entity recognition on queries task is word + char level neural networks architecture. The algorithm includes two main parts:

1) Bi-directional Long short-term memory

Bi-directional Long short-term (Bi-LSTM) is an advanced architecture in Recurrent neural network framework. The best part of Bi-LSTM is that it not only captures the sequential information of queries while avoid gradient explosion as LSTM model, but also use forward and backward in formation for tokens in the text. For our model, we applied Bi-LSTM on two layers. The first layer is on the character level. We applied the Bi-LSTM to get the vector representation of characters of queries. Secondly, we used the Bi-LSTM on the combined word representation. On this layer, Bi-LSTM is to capture the whole vector representation of words.

2) Conditional Random Field

Conditional Random Field (CRF) is a typical model for named entity recognition. CRF is part of sequential models. The chain structure of CRF predicts the label of sequence of texts. CRF is a graphical model which depicts a conditional distribution of observed and output variables. Now CRF is popular in many NLP tasks, such as POS tag, named entity recognition and parsing. We applied the the CRF layer on the top of the second LSTM layer and try to predict the final named entity labels.

The architecture is shown in Figure 1.

3.3 Implementation

We firstly use pandas for data ETL. We load the data and converted into data frame. Fortunately, the dataset is already tokenized by SLS group, which means we save the step of tokenization in NLP. Then we built a function to extract the token and corresponding named entity labels in the data and organized them by sentence.

The next step is to vectorize those tokens and labels. We built a dictionary to count the frequencies of tokens and labels, transferring them to numbers and convert them to numpy arrays for use. Since we also used the characters as another features, we preprocess the text and extract the characters, transferred it into vector and used numpy array to represent it. The steps for vectorize character are similar to that of vectorize word.

Finally, we used Keras and Keras_contrib to build the main structure for deep learning model. The Keras is used to build Bi-LSTM layer and the other layers like input, output embedding and dropout layer. The Keras_contrib is only used to construct the conditional random field layer. We construct this layer only before the output.

Finally, we used scikit learn for train test split and check the metrics data of the model. We use precision, recall and F1 score as metrics.

For comparison, we also built the word level neural network (BiLSTM+CRF) as baseline. As a typical neural network model for named entity recognition, word level NN model has been proved to have good performance on most kinds of text. Here we use this model as a baseline and compare its performance with the word plus character model.

4. Results and Analysis

The train loss is shown as in figure 2. We can find out that the training epochs is long for our model. Exactly, we set the epoch of training epochs to 1000. As shown in the the figure, the train loss decreases drastically before 200 epochs, and it slowly decreases, after 800 epochs, the curve is basically stabilized.

The reason that we need a longer than average epoch is that we have longer batch size and more complex neural network than vanilla Bi-LSTM-CRF model. Exactly, we construct two LSTM layer in the model. In addition, by deep learning theory, longer epochs are required to achieve same accuracy when batch size is larger.

Now let us investigate the the metric table about the model (Table 1). First of all, the average f score is only 0.29 for all labels. In addition, the average precision is 0.31 and average recall is 0.31. This may seems very disappointing at first glance. However, it turns out that named entity recognition on informal text falls far behind the formal text. For example, the baseline (Bi-LSTM-CRF) model on tweets has only 0.30 f score (Strauss, 2016). Therefore, even 0.29 F1 score is not a good result, but it is still acceptable.

Second, when we dive into the the metrics on different categories, there are some more interesting facts. The amenity and location have the highest metrics score among 8 categories listed. This show that our model has a better prediction power on amenity and location in queries than other named entities. On the other hand, named entities like hours, ratings and price only have small metrics score, which means the model does not behave well on these named entities. Another interesting fact is that the recall for named entity cuisine is as high as 0.57, while other

named entities are far behind this score on recall. This means cuisine has a higher percentage of relevant results which is correctly classified than others.

We now compared the result of word plus character NN model and word level model from Figure 3 and Table 2. We have several findings as follows: 1) the training time for word plus character model is longer than the baseline model. The reason is that as a fancier model, word plus char model has more layers and more vectors, which leads to more parameters to be trained. Two layer of Bi-LSTM makes the model time costly. 2) the F score for word plus char model is also smaller than the baseline word level NN model (29% to 52%). This result contradicts to the performance on formal data like newswire or books. The main reason is that the sequential information for characters in short queries is more chaotic than that of formal text, which makes the word plus character NN model fails to predict accurately. The informal syntax of characters actually negates the sequential pattern of word captured by LSTM model, instead of enhancing them like in formal text.

5. Conclusion

We built a word-char level NN model and applied the model on named entity recognition on short queries. From the experiment result, we have conclusions as follows:

First, the model performance of our model on named entity recognition on queries is acceptable, even though they are not high. By check precision, recall and F1 score, we found out basically all the metrics are about 0.35. These results are still smaller than that on formal text. The reasons are as follows: 1) the queries are short. This feature leads to the bad performance on LSTM layer. Since we use LSTM layer to try to capture the sequential information in sentences, the shortness of queries obstructs this consistency in text and may fail our attempt. 2) the queries violate formal syntactical rules. As people who propose queries often gave syntactically wrong sentences, the sequential relation of named entities may not obey the formal syntactical rules. This may destruct our CRF layer performance and further model performance.

Second, according to running time of our algorithm, it is slower than the common Bi-LSTM-CRF model on word level. There are several potential reasons. Firstly, we build a model including two LSTM layer and CRF layer. This means we have more parameters train compared with common NN models. The training on parameters is costly. Another reason that our model is slow is that we concatenate the representation of character vector and representation of word vector as input to the second layer Bi-LSTM. This concatenation built a longer vector than plain word vectors, which takes more time to train.

Appendix

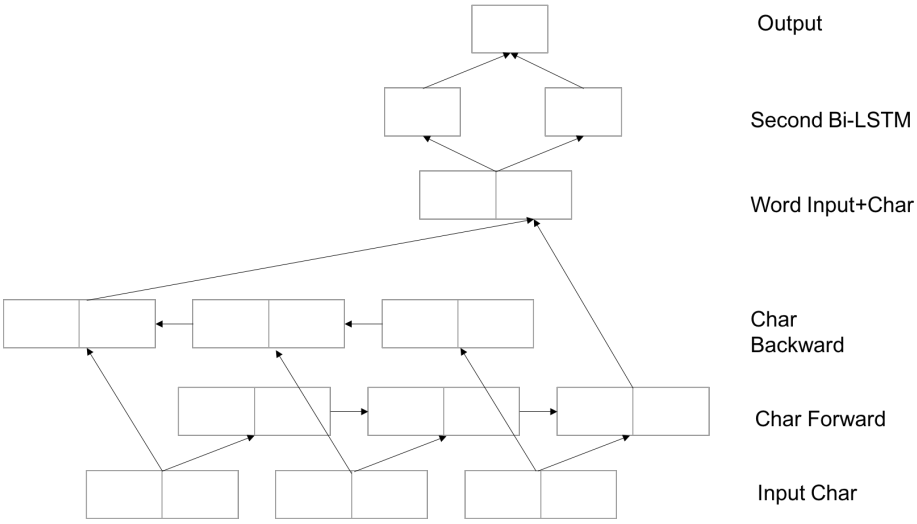

Figure 1 The word + char NN model architecture

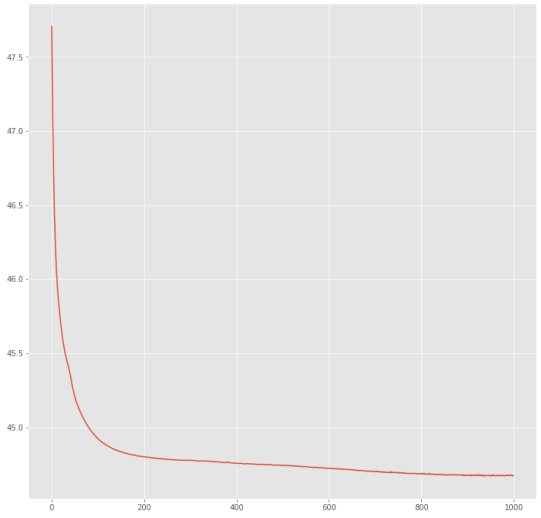

Figure 2 Train loss for word + char level NN model

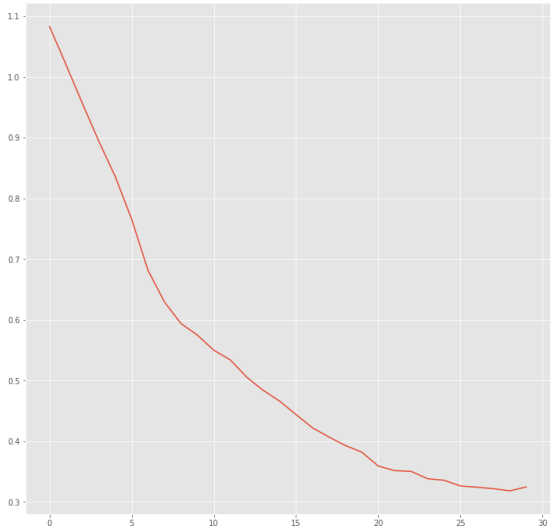

Figure 3 Train loss for word level NN model

Table 1 Metrics for word + char level NN model

	Precison	recall	F score	Support
Amenity	0.31	0.39	0.35	167
Rating	0.29	0.16	0.21	49
Location	0.31	0.31	0.31	236
Restaurant_Name	0.25	0.19	0.22	124
Hours	0.15	0.05	0.08	55
Price	0.20	0.23	0.22	43
Cuisine	0.36	0.57	0.45	164
Dish	0.50	0.13	0.21	93
Avg/total	0.31	0.31	0.29	931

Table 2 Metrics for word level NN model

	Precison	recall	F score	Support
Amenity	0.46	0.49	0.47	167
Rating	0.46	0.43	0.44	49
Location	0.70	0.69	0.69	236
Restaurant_Name	0.40	0.35	0.37	124
Hours	0.42	0.43	0.43	55
Price	0.53	0.37	0.44	43
Cuisine	0.50	0.64	0.56	164
Dish	0.50	0.30	0.38	93
Avg/total	0.52	0.52	0.52	931

References

[1] Bikel, Daniel M., et al. "Nymble: a high-performance learning name-finder." arXiv preprint cmp-lg/9803003 (1998).

[2] Chiu, Jason PC, and Eric Nichols. "Named entity recognition with bidirectional LSTM-CNNs." Transactions of the Association for Computational Linguistics 4 (2016): 357-370.

[3] Etzioni, Oren, et al. "Unsupervised named-entity extraction from the web: An experimental study." Artificial intelligence165.1 (2005): 91-134.

[4] Guo, Honglei, et al. "Domain adaptation with latent semantic association for named entity recognition." Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2009.

[5] Grishman, Ralph, and Beth Sundheim. "Message understanding conference-6: A brief history." COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics. Vol. 1. 1996.

[6] Liu, Xiaohua, et al. "Named entity recognition for tweets." ACM Transactions on Intelligent Systems and Technology (TIST) 4.1 (2013): 3.
[7] Ma, Xuezhe, and Eduard Hovy. "End-to-end sequence labeling via bi-directional lstm-cnns-crf." arXiv preprint arXiv:1603.01354 (2016).
[8] Malouf, Robert. "A comparison of algorithms for maximum entropy parameter estimation." proceedings of the 6th conference on Natural language learning-Volume 20. Association for Computational Linguistics, 2002.
[9] McCallum, Andrew, and Wei Li. "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons." Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. Association for Computational Linguistics, 2003.
[10] Sekine, Satoshi. "NYU: Description of the Japanese NE system used for MET-2." Proc. of the Seventh Message Understanding Conference (MUC-7. 1998.
[11] Strauss, Benjamin, et al. "Results of the wnut16 named entity recognition shared task." Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT). 2016.
[12] Zhang, Shaodian, and Noémie Elhadad. "Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts." Journal of biomedical informatics 46.6 (2013): 1088-1098.
[13] Zhou, GuoDong, and Jian Su. "Named entity recognition using an HMM-based chunk tagger." proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2002.