Named-Entity Recognition Model on Chemical-Protein Dataset

Training and testing NER model with Spacy

SEOKHWAN SONG

Iowa State University

Through text mining chemical-protein interactions, it will be able to detect relations between chemical compounds and genes [1]. In this paper, we train and test NER(Named-Entity Recognition) model on a chemical-protein dataset with Spacy. The methodology is introduced and through some experiments, the results such as precision and error are discussed for better performance.

CCS CONCEPTS • Named-Entity Recognition • Text mining • Chemical-protein

Additional Keywords and Phrases: Machine Learing, Python, Spacy

ACM Reference Format:

SeokHwan Song. 2018. Training and Testing NER Model on Chemical-Protein Dataset: In Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY. ACM, New York, NY, USA, 10 pages. NOTE: This block will be automatically generated when manuscripts are processed after acceptance.

1 INTRODUCTION

Named entity recognition (NER) is one of the challenging learning processes because of a lack of the amount of available train data. However, most languages can be used and the range is very wide [2]. Therefore, in this paper, we even use the dataset about chemical-protein to detect the relations between chemicals and genes. The dataset has text information about PubMed abstracts, manually annotated chemical and gene/protein mentions [1]. It also provides offsets where the entity mentions are located and their types. Through NER model, the machine will detect what an entity mention's type is with the mention.

The natural language procession library called, Spacy is used for the NER model in python, and data preprocessing is created and some experiments are conducted with two models, the different numbers of iterations, and three different sizes of batches.

2 METHODOLOGY

To train and test the dataset, data preprocess and modeling are applied.

2.1 Data Preprocess

As long as, Spacy model is used, the dataset should be preprocessed in a certain format. There are train datasets and test datasets. Both datasets have two datasets for 'entities' and 'abstracts'. The entity datasets have information about entities such as mentions and offsets. The abstract datasets have information about abstracts such as the actual abstracts and titles.

There are some steps to do preprocess. Firstly, when the datasets are imported with pandas, the chemical compound, ‘NA’ is changed into NaN, so all ‘NA’ has to be changed to str(NA). Moreover, the offsets provided are based on the whole texts including titles and abstracts, so merging them is necessary. To train the text, each entity has to be connected with a sentence which the entity is located in. Therefore, the merged abstracts and titles have to be tokenized and connected to the entity mentions. NLTK tokenizer is used to conduct the task. Furthermore, some entity mentions overlap with each other like ‘CD86’ and ‘CD86 receptor’, and when these entities are used to train with one sentence, it will make a problem with the Spacy model. Therefore, longer entity mentions are removed. Lastly, some sentences do not have any entity mentions on it and some entity mentions have empty features, therefore, they are removed.

2.2 Model

The library Spacy NER model applies an embedding strategy, called ‘Bloom’ with using sub-work features. Moreover, convolution layers are also applied with residual connections, and this system also normalizes layers and uses non-linearity. Through the system, it will give you better performing models with balanced efficiency and accuracy [3].

Two different models, pre-existing and blanc Spacy models are used with several steps. First of all, the model uses a function to shuffle the examples randomly. During the modeling, there are some loops to train for the number of iterations, and call batches to get data in the batches. [4]

3 EXPERIMENT

Firstly, two different models, the pre-existing and the blanc Spacy model are compared. Secondly, models with different sizes of batches and different numbers of iterations are compared. The 20, 30, and 40 of iterations and batches, (1.0, 4.0, 1.001), (1.0, 10.0, 1.5), and (4.0, 32.0,1.001), are applied.

4 ERROR ANALYSIS

4.1 Scores

The values of precision, recall, and F scores are compared, and F scores are calculated by labels as well.

4.2 Tables

Firstly, the pre-existing Spacy model and the blank Spacy model are compared. All scores of the blank model are higher than the pre-existing model, so we can say that using the blank model performs better. Since the blank model performs better than the pre-existing model, the blank model is used with different parameters. On the blank model, 30 iterations and batch size (1.0, 4.0, 1.001) are applied.

Table 1: Scores of Train and Test with pre-existing Spacy model and blank Spacy model

Train (Pre-existing model)	Test (Pre-existing model)	Train (Blank model)	Test (Blank model)
Model Precision	Model Precision	Model Precision	Model Precision
92.6038	77.3681	96.0053	78.2201
Model Recall	Model Recall	Model Recall	Model Recall
92.6321	71.9930	96.8872	74.5934
Model F Score	Model F Score	Model F Score	Model F Score

Train (Pre-existing model)	Test (Pre-existing model)	Train (Blank model)	Test (Blank model)
92.6630	74.5838	96.4442	76.3637
Label ‘GENE’ F Score	Label ‘GENE’ F Score	Label ‘GENE’ F Score	Label ‘GENE’ F Score
91.0669	72.1138	95.8032	74.1087
Label ‘CHEMICAL’ F Score	Label ‘CHEMICAL’ F Score	Label ‘CHEMICAL’ F Score	Label ‘CHEMICAL’ F Score
94.0414	76.6561	97.0105	78.3260

Secondly, we also compared models with different batch sizes and iterations. When only batch sizes are changed, the scores of training get better, but the performances with test dataset are worse. With the bigger number iterations, which is 45, the scores with the train dataset dramatically increase but the scores with the test dataset decrease and the model takes longer. With the smaller number iterations, which is 20, it takes shorter than the model has 30 iterations but the model with 30 iterations performs better.

Table 2: Scores of Train and Test with blank Spacy model with different number iteration and batch size

Train (Batch size: 4.0, 32.0, 1.001)	Test (Batch size: 4.0, 32.0, 1.001)	Train (Iterations: 20)	Test (Iterations: 20)
Model Precision	Model Precision	Model Precision	Model Precision
99.1992	77.6197	97.4736	77.7150
Model Recall	Model Recall	Model Recall	Model Recall
99.6478	75.0850	98.4293	73.9419
Model F Score	Model F Score	Model F Score	Model F Score
99.4230	76.3313	97.9491	75.7816
Label ‘GENE’ F Score	Label ‘GENE’ F Score	Label ‘GENE’ F Score	Label ‘GENE’ F Score
99.3460	74.3240	97.5058	73.8436
Label ‘CHEMICAL’ F Score	Label ‘CHEMICAL’ F Score	Label ‘CHEMICAL’ F Score	Label ‘CHEMICAL’ F Score
99.4918	78.1341	98.3438	77.5420

5 DISCUSSION

With Spacy, training NER model on chemical-protein dataset performs well. Through some experiments with different parameters, the performances of the model change a little bit but not dramatically. With a different way to preprocess the dataset or a bigger amount of training set will help this model to improve the accuracy. Moreover, setting a different value of dropout rates is not applied in this paper, but it would help to improve the performance as well [3].

ACKNOWLEDGMENTS

I had some discussions with Yun Qi Bang.

REFERENCES

[1] The BioCreAtIvE (Critical Assessment of Information Extraction systems in Biology). Retrieved from https://biocreative.bioinformatics.udel.edu/tasks/biocreative-vi/track-5/

[2] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. arXiv:1603.01360. Retrieved from https://arxiv.org/abs/1603.01360

[3] Spacy. Retrieved from https://spacy.io/models

[4] Machine Learning Plus. Let’s Data Science. Retrieved from https://www.machinelearningplus.com/nlp/training-custom-ner-model-in-spacy/