Named Entity Recognition using Word Embeddings and Part-of-Speech Embeddings

Anonymous ACL submission

Abstract

In this work I present several experiments carried out in the field of Named Entity Recognition — NER, focusing on an approach based solely on the representation of words without the help of gazetteers. The model uses two sources of information, one is the representation of words through pre-trained word embeddings and the second one is the representation of the same sentences through Part of Speech embeddings. Reaching an average of 92 % between validation and test set.

1 Introduction

This report shows the techniques I used to deal with the Named Entity Recognition — NER task. Named entities are sentences that contain the names of persons, organizations and locations. The task require to assign each word one of the following {PERSON, ORGANIZATION, LOCATION, O} class used to distinguish the different types of entities. The O tag is used for tokens recognized as non-entity.

2 Dataset and Evaluation

This section will illustrate information relating to the dataset and the metrics used for performance evaluation.

Dataset The dataset used in this task contains sentences and labels in English. The dataset has already been divided into three portions: train, dev and test. The dataset was supplied pre-tokenized, to avoid bias with the different tokenization methods and evaluate the task on the same level. The portion of the train is made up of $100,000$ sentences, the dev is made up of $14,434$ sentences and the test are made up of $15,474$ sentences. In the table 3 are reported statistics of named entities per data file.

Evaluation The task is a multiclass classification to be more precise we have four unbalanced classes. The metrics taken into account for the NER tasks the F1-Score, the precision and the recall all and the metrics in the macro version.

3 Pre-processing

Despite several attempts to use preprocessing techniques such as stemming and lemmatization, I found the use of pos tagging particularly useful, as will be explained later in the section 4.2. I have kept capitalization unchanged, to avoid generating ambiguity with 'will' the verb and the name 'Will'. In this task every single token must be classified, for this reason it was not possible to remove words (for example stop words) or to use other tokenizers capable of removing punctuation. A lot of text cleaning work has been solved with the application of the pos tagger able to highlight punctuation marks, articles and conjunctions.

4 Neural Network Architecture

In this section I show the main components of the neural network used.

4.1 Word embeddings

The use of word embeddings allows you to create a meaningful representation of words in a large vector space. It is shown that the use of pre-trained word embeddings is better in most cases than a randomly initialized embeddings. For this reason, after several tests, the model presented uses a pre-trained vector by Google with a vocabulary of 3 millions of words called *Google-news300D* (Chiu and Nichols, 2016), where the words are represented in a 300-dimensional space. Probably Google trained embeddings outperformed other pre-trained embeddings for the number of words in common between the training vocabulary and that of Google.

4.2 Part of Speech embeddings

In addition to the representation of words by word embeddings, as demonstrated in other studies, POS tagging can be a trusted friend to solve the problem of the Named Entity Recognition. One of the greatest advantages brought by the use of a pos tagger is obtained when the model meets an out of vocabulary word and therefore is forced to consider it as an unknown word by losing all the information. This is because a name not present in the vocabulary is treated like any unknown word, and we can only rely on the context to be able to understand if that unknown word could be an entity and what type of entity or not. The pos tag of a word provides important information for determining the use of the word and those in context. For example, in POS tagging an adjective is more likely to be followed by a noun than a verb (Ma and Hovy, 2016). The proposed model uses the pos tagger offered by spaCy (Honnibal and Montani, 2017). To do this, therefore, I used an embeddings layer of size 300. The inputs of this layer are the words belonging to a phrase on which POS Tagging has been applied.

4.3 Bidirectional Long Short-Term Memory

Along with word embeddings, another constant of the NER models (Luo et al., 2018; Chiu and Nichols, 2016) is the use of LSTMs and BiLSTM. Since their advent, they have become the point of reference for many natural language processing problems. The use of a bidirectional LSTM allows to consider the context from left to right and from the opposite side.

4.4 Conditional Random Field — CRF

The vast majority of state-of-the-art models in the field of Named Entity Recognition, use a combination of LSTM and a CRF layer. This is because CRF helps improve the model's performance when the current prediction is affected by the previous ones. For this reason I have tested several combinations of models with the CRF layer.

5 Experiments

During my experiments I started from a basic model to which I then incrementally added further components. In the experiments I used *Cross Entropy* as a loss function and *Adam* as optimizer.

Word Embeddings + BiLSTM The first model used in my experiments was a 100-dimensional non-pre-trained word embeddings model and two BiLSTMs ending with a multi-layer neural network using it as a baseline.

Google News 300d + BiLSTM Subsequently, as previously mentioned, I used the pre-trained by Google where each word has representation in a 300-dimensional vector space.

Google News 300d + BiLSTM + CRF Then I added a layer of CRF, changing the loss function using the *Negative Log-Likeihood* — NLL Loss.

(Google News 300d + BiLSTM) + (POS + BiLSTM) Finally I moved on to a multi-input model which additionally takes as input a vector containing the pos tagging of the sentences. This second input is used within a dedicated 300-dimensional embeddings whose output is passed within 3 BiLSTM and then within a linear layer and then linked to the output of the 3 BiLSTM of the word embeddings. This model was trained with 3 epochs. In table 4 I have proposed two versions of this model with and without the CRF layer.

6 Results

Table 4 shows a comparison of the different models proposed during the experiments. The table shows a baseline which was provided by the task organizers. All the models shown in it are in the best hyperparameter configuration that I have found for them. As you can see from table 4, the model that has obtained the highest performance is composed of a multi-input neural network with word and pos embeddings, separately using three bidirectional LSTM layers and these networks join in the layers of dense linear network with a dropout with 40% probability. Table 1 shows the F1 scores for each individual class and table 2 shows the precision, recall and F1 metrics. A particular aspect is to note the increase in performance obtained using the Conditional Random Field in the model with pre-trained word embeddings, leading the model to reach a higher score of 7 - 8% confirming the usefulness of the CRF as shown in the state models of art, particular instead in the case of the multi-input model with pos tagging where the model without CRF reaches a score very little higher so as to make irrelevant the work of the CRF when in reality I would have expected a higher score since NER is a sequence labeling task. In the **confusion matrix** shown in the figures [1, 2, 3, 4] tables, you can see that the person entity was easier to recognize than to identify organizations and locations.

References

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

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7.

Ling Luo, Zhihao Yang, Pei Yang, Yin Zhang, Lei Wang, Hongfei Lin, and Jian Wang. 2018. An attention-based bilstm-crf approach to document-level chemical named entity recognition. *Bioinformatics*, 34(8):1381–1388.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

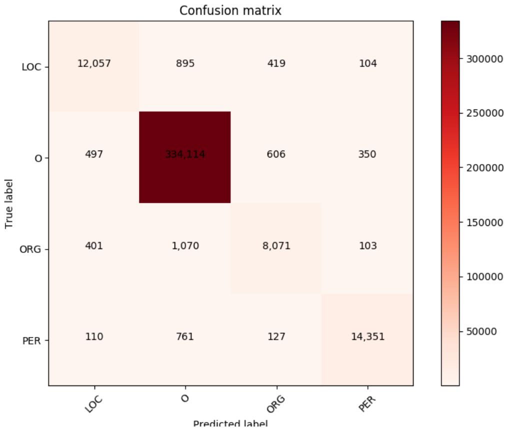

Figure 1: Dev - Confusion Matrix

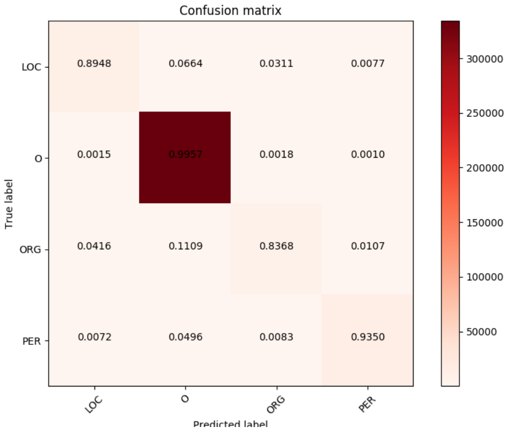

Figure 2: Dev - Confusion Matrix - Normalized

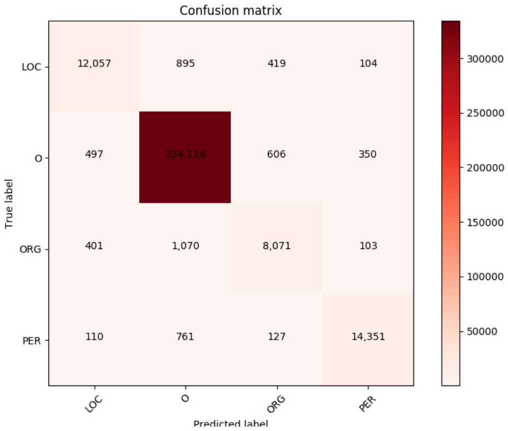

Figure 3: Test - Confusion Matrix

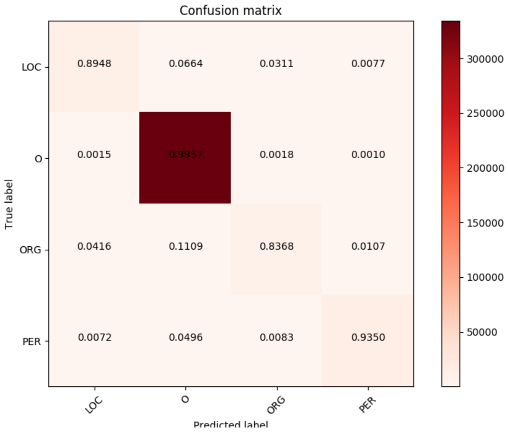

Figure 4: Test - Confusion Matrix Normalized

Tag	F1 - DEV	F1 - TEST
O	99.19%	99.37%
PERSON	94.23%	94.86%
LOCATION	90.68%	90.85%
ORGANIZATION	84.55%	85.55%
ALL TAG	92.19%	92.66%

Table 1: POS emb + W2V emb Score

Metric	**DEV**	**TEST**
Precision	93.43%	93.81%
Recall	91.02%	91.56%
F1	92.20%	92.66%

Table 2: Docker Score.

Entity	**Train**	**Dev**	**Test**
O	2,177,423	315,809	335,567
PERSON	100,409	14,396	15,349
LOCATION	84,937	12,359	13,475
ORGANIZATION	61,988	9,043	9,645

Table 3: Dataset statistics - tag distribution

Model	**F1 - DEV**	**F1 - TEST**
Baseline	24.88%	24.97%
Word Emb (100d) + BiLSTM	78.02%	78.73%
Word2vec + BiLSTM	80.25%	81.04%
Word2vec + BiLSTM + CRF	88.12%	89.76%
(POS emb + BiLSTM) + (W2V + BiLSTM) + CRF	92.11%	92.37%
(POS emb + BiLSTM) + (W2V + BiLSTM)	92.20%	92.66%

Table 4: Model comparison.