NLU : Assignement - III

anirbanb@iisc.ac.in

1 Problem Statement

The task is to build an NER system for diseases and treatments. The input of the code will be a set of tokenized sentences and the output will be a label for each token in the sentence. Labels can be D, T or O signifying disease, treatment or other.

2 Solution Approach

To solve the task of building a **Named Entity Recognition** (NER) system, I considered a type of discriminative undirected probabilistic graphical model, namely, **Conditional Random Field(CRF)**. The reason behind this choice of model is CRFs fall into the sequence modeling family and unlike a discrete classifier which predicts a label for a single sample without considering "neighboring" samples, a CRF can take *context* into account. It is often used for labeling or parsing of sequential data, such as *natural language processing* or *biological sequences*.

2.1 Tools

I have used CRF implementation provided by `sklearn-crfsuite` to design my NER model. It is thin a CRFsuite (python-crfsuite) wrapper which provides scikit-learn-compatible CRF estimator. The training algorithm for CRF model is L-BFGS with Elastic Net (L1 + L2) regularization.

To generate the word embedding vectors, `Word2Vec` library has been utilized. `NLTK` toolkit's pos_tag API gave the required Parts of Speech of a word. Both of these are used in feature modelling for CRF.

2.2 Feature

As a simple baseline, **word identity**, **word suffix**, **word shape** are used. Some information from nearby words are also considered. Start/end-of-sentence words are taken into account as features.

I add some more features to get better results. **POS-tag** is one such feature. I also include **word-vector embedding** as a feature. **Most-similar-word** of the current word is another such feature.

Apart from those mentioned above, whether a word is capitalized and whether it is a titular word - considered as features.

2.3 Evaluation Metric

The given data is highly imbalanced in nature. Most of the words fall in category O. A very low percentage of words are from category D and T. The following table shows the data distribution:

D	T	O
4889	3821	55810

Table 1: Word counts for each type of label

Hence, accuracy is not a good measure for this task. *F1-score* (takes care of both precision and recall) is a more appropriate evaluation metric.

2.4 Methodology

A 10-fold cross validation is done. I train on 9 folds, use the 10^{th} as test set. Repeat this 10 times with different folds as test set. This is a robust option.

3 Experimental Results

I perform alternative experiment by incrementally adding sets of features. The final goal is to identify the most useful features (and the value of each feature) for this task. In addition to quantitative results, I also look at specific examples and try to qualitatively understand value of each feature by noticing which examples each feature helps in.

3.1 Results : Different Feature Space

The following tabular formatted output presents precision, recall and F1-score for various features considered. The first table depicts results considering only baseline features. Next table on-wards, features like POS-tag, word embedding and similar words are added one-by-one.

3.1.1 Baseline

	Precision	Recall	F1-score
D	0.797	0.626	0.701
O	0.938	0.975	0.956
T	0.724	0.499	0.591

Table 2: Precision, Recall and F1-score for each label. Feature set : Baseline (word identity, word suffix)

3.1.2 Baseline + POS tagging

	Precision	Recall	F1-score
D	0.801	0.648	0.716
O	0.940	0.975	0.957
T	0.726	0.508	0.597

Table 3: Precision, Recall and F1-score for each label. Feature set : Baseline and POS-tag

3.1.3 Baseline + POS + Embedding

	Precision	Recall	F1-score
D	0.792	0.676	0.729
O	0.945	0.972	0.959
T	0.731	0.555	0.631

Table 4: Precision, Recall and F1-score for each label. Feature set : Baseline along with POS-tag and word embeddings

3.1.4 Baseline + POS + Embedding + Similar Word

	Precision	Recall	F1-score
D	0.788	0.681	0.731
O	0.946	0.970	0.958
T	0.714	0.570	0.634

Table 5: Precision, Recall and F1-score for each label. Feature set : Baseline along with POS-tag, embeddings and most similar word

3.2 Results: Visualization

In this section, I present the evaluation in a visual form. The following bar graphs provides a better understanding on how different features are affecting the f1-score for three labels (D, O and T).

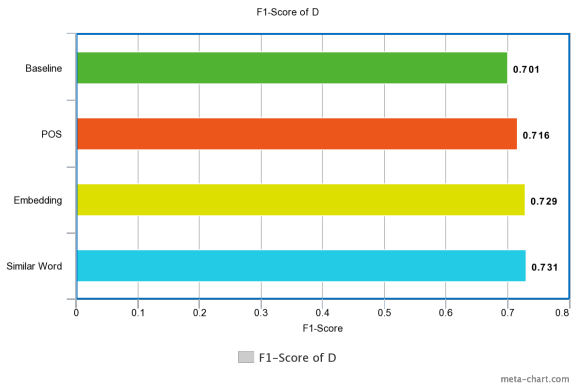

Figure 1: Comaprison of F1-score for label - **D**

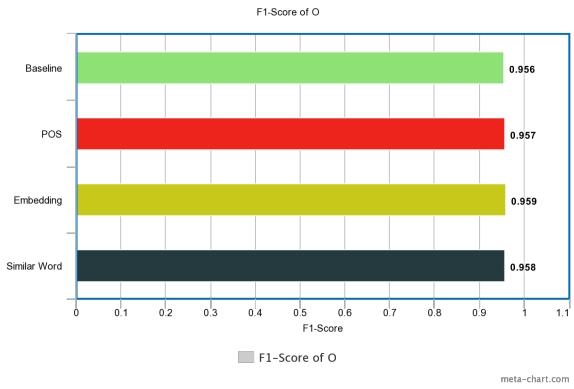

Figure 2: Comaprison of F1-score for label - **O**

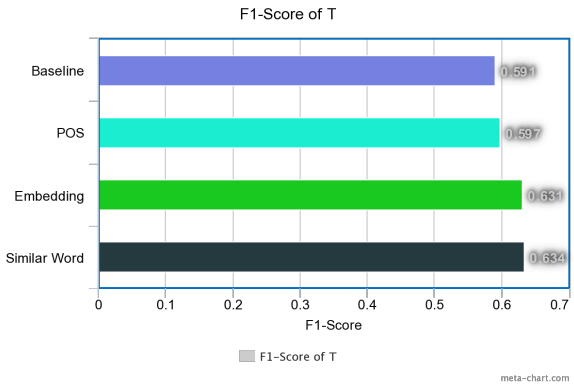

Figure 3: Comaprison of F1-score for label - **T**

3.3 Discussion

3.3.1 Final Model

It can be observed from Figure 1 and 3, as we have added features, the f1-score value increases. Hence, we finalize the model corresponding to Table 5 where the feature set includes baseline features along with pos-tag, word embedding and most similar word feature. This gives best results in terms of our evaluation metric.

For label-D, we can see POS-tag and word embedding gives significant increase in f1-score whereas in case of label-T, adding word embedding as feature provides much better results. We don't find any significant improvement for label-O probably because the data-set is highly imbalanced and the f1-score is quite high.

3.3.2 Hyperparameter Optimization

To improve quality, hyperparameter tuning is done. I try to select regularization parameters using randomized search and 3-fold cross-validation.

In our task the hyperparameters are $c1$ and $c2$. The params that gives best results are : $c1 : 0.1746, c2 : 0.04084$

3.3.3 Top Features

Here is some observations on few top-positive and top-negative features which have been most influential for our model :

- **+6.250923 D word.lower():infertility -** the model remembered names of some entities - maybe it is overfit, or maybe our features are not adequate, or maybe remembering is indeed helpful
- **-2.128486 O +1:word.lower():patient -** model learns that if next word was "patient" then the token is likely a part of label-O