Text Mining Hand-in Assignment 2

Elin Benja Dijkstra (s2696096)

November 10th 2019

1 Introduction

The field of automatically identifying biomedical entities mentioned in patents and linking them to existing knowledge basis is an important topic in information extraction from text. For this assignment we have a data set containing citings from patents. We want to label the entities in the documents. In order to achieve this, we will train Conditional Random Fields (CRF) to perform the sequence labelling task.

2 Data Exploration

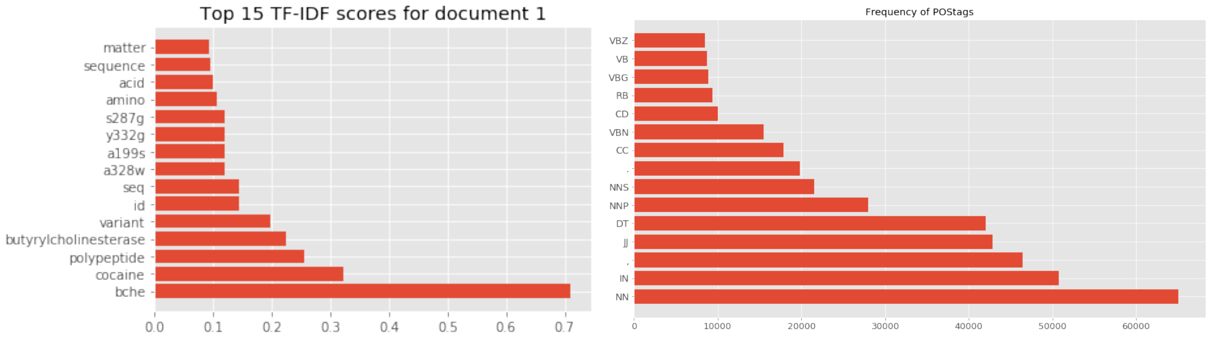
Let us first take a look at the data. The full dataset consists of 22 BIO-labelled files. The documents consist of lines formatted as (word, POStag, label). POS stands for part of speech and identifies the type of word (e.g. verb, noun etc.). Each word gets a label: (B) for the beginning and (I) for the inside of each entity type, and (O) for the tokens outside any entity. The documents are all of different length. We also see some inconsistencies in the data where we are missing labels for some on the words.

Document Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Size	3468	29588	5857	4526	16242	3295	2109	18073	11990	5226	33903	26200	510	11933	40735	83701	32188	4202	10332	15081	52615	27178
Missing labels	1	17	0	2	5	8	1	1	1	1	8	5	0	1	17	28	3	3	1	2	13	5

Table 1: Description of dataset content

To get some insight into the most important words, the TF-IDF values are calculated. The results for Document 1. are shown in Fig 1a. In Document 1. "bche" is the term with the highest TF-IDF value. BCHE is a gene connected to the production of butyrylcholinesterase (4^{th} highest value) in the liver.

It is also interesting to look at the frequency of the POStags that occur the corpus. Again, the top 15 most frequent POS tags are displayed in 1b. NN is the most common POS tag in the corpus which belong to "Noun, singular or mass".



(a) Top 15 TF-IDF of document 1

(b) Frequency of the POStags

Figure 1: Data exploration on words and POStags

3 Methods

3.1 Pre-processing

Before we begin building our model, we need to make sure the data is structured the way we need it to be. Our dataset consists of 22 files. Since we want to perform cross validation where the documents only occurs in either the training set or the test set, we load the documents individually. As mentioned in Section 2, there are some inconsistencies in the data which will also be removed during this step by removing these lines. We do need to keep in mind that this effects the sequence. However, since we have seen in 1 that the number of missing labels is fairly small, we accept the resulting inaccuracies.

3.2 Cross-Validation

To get a better estimate of the performance of our model, cross-validation is implemented. It should be avoided that parts of a document appear in both the training and the test set to prevent overfitting. Therefore GroupKFolds is used for both splitting the data set in a training and test set initially and the hyperparametersearch. Five fold cross validation is implemented. This entails that for five different combinations of documents for the training and test set evaluation and hyperparameter search is done of which we subsequently take the average. Unfortunately, due to time restrictions the hyperparameter search is restricted which could result in fluctuations in the optimal parameters found.

3.3 Conditional Random Fields

Conditional Random Fields (from hereon: CRF) can be seen as undirected graphical models. It optimizes sequences as a whole by including features and predicted labels in future timesteps. It predicts the current label by extracting features of the current timestep and applying the weights associated with these features. These features are learned by the model. It also includes previous labels.

3.4 Features

In the tutorial, some standard features are already implemented such as the last two and three letters of the word, and a context of -1 and +1. Hoping it will improve the performance, the feature set is expanded. First, we add a larger context of -2 and +2. Additionally, we look at the length of the current word and the words in the context. In biology patents, long words could point towards genes or other scientific names. We also check if the words contain symbols. For example is the word is hyphenated. The three different features are evaluated separately.

4 Results

We run the 5-fold Group cross-validation and hyper parameter search on the features described in 3.4. We test each feature invidually. The resulting measures are shown in Table 2.

Feature Set	**Result GridSearch with highest F1 Score**	**Weighted Precision**	**Weighted Recall**	**F1-score**
Baseline	'c1': 0.2001 'c2': 0.1092	0.8864	0.8152	0.8482
Larger context (+2, -2)	'c1': 0.1637 'c2': 0.0147	0.8832	0.8376	0.8496
Contains symbol	'c1': 0.0661 'c2': 0.0014	0.8804	0.8112	0.8438
Word length	'c1': 0.0845 'c2': 0.0360	0.8852	0.8122	0.8462

Table 2: Results of the CRF and hyperparameter search.

None of the features have a large effect on the precision of F1-score. Enlarging the context to -2 and +2 has some positive effect on the recall. When looking at the individual results per fold, there are high fluctuations for each feature set. The parameters for the fold with the highest F1-score are included. As mentioned before, this is possibly due to the restricted number of iterations for the hyperparameter search.

5 Discussion and Conclusion

When looking at the results we see that a larger context has a positive effect on the performance. However, including word length or a "contain symbol" boolean does not result in any significant improvement. To improve our model further, we could choose to expand the features or combine features that have a positive impact on the performance when implemented individually. Additionally, the hyperparameter search can be expanded. However, for the sake of runtime, the number of iterations is kept low.