PAyatsOValls at SemEval-2019 Task [9]: [Detection and Interaction of Drug-Drug from Biomedical texts]

Pere Ayats
pereeayats@gmail.com

Olga Valls
ovalls@gmail.com

Abstract

The DDIExtraction 2013 task concerned the recognition of drugs and extraction of drug-drug interactions that appear in biomedical literature. In this report our solutions will be addressed in order to understand the process followed to complete the different tasks.

1 Introduction

The *SEM conference (Lexical and Computational Semantics) holds yearly different workshops about evaluations of computational semantic analysis systems. The evaluations are intended to explore the nature of meaning in language. For the SemEval-2013 workshop, its task 9 concerned the recognition of drugs and extraction of drug-drug interactions.

It was divided in two different subtasks that had different possible approaches:

1. The recognition and classification of drug names.
2. The extraction and classification of their interactions.

In the following pages we will see how we decided to tackle both problems.

2 NER and DDI Corpus

The original corpus consists on a Training folder and two test folders, one for the NER and the other for the DDI. Each of the folders contains XMLs from the DrugBank and from MediLine. Those subfolders have been mixed into one common folder per training data and one common folder for test data. For the validation data 10% of the training data has been used.

3 Task 9.1: Recognition and classification of pharmacological substances.

The approach used for this task is to train a CRF (Conditional Random Field) feed with several features extracted from each of the tokens that composes each of the sentences of the XML files.

For the current token, the previous and the next, some features has been tested. The ones that have given the best results are the following:

- form: the token itself
- formlower: the token itself in lowercase
- suf3: last 3 characters of the token
- suf4: last 4 characters of the token
- isUpper: is the token uppercase?
- isTitle: is the first character of the token in uppercase?
- isDigit: is the last character a digit?
- hasSymbol: has the token any of the following characters? \, (,), digits, , , -, +
- inDron: is the token inside the drug list from the Drug Ontology?

A Drug Ontology from Bio Portal has been used to create a list of drugs to check whether a token is a drug or not.

In the creation of the features, the tokens has been lemmatize to create a lemma feature, although it has been seen no improving in results using that feature.

Some of the features that have not improve the results are, for the current token, the previous and the next:

- suf5: last 5 characters of the token
- iniCons: does the token starts with a consonant?
- iniVowel: does the token starts with a vowel?
- has2cons: has the token 2 consonants together?
- has3cons: has the token 3 consonants together?
- has2vowels: has the token 2 vowels together?
- has3vowels: has the token 3 vowels together?
- has3Suffix: are the last 3 characters in the suffixes list?
- has4Suffix: are the last 4 characters in the suffixes list?
- has5Suffix: are the last 5 characters in the suffixes list?
- lastUpper: is the last character of the token in uppercase?
- lastDigit: is the last character of the token a digit?
- postag: postag of the token
- lemma: lemma of the token
- combinations of bigrams and trigrams
- wordfreq: frequency of the token (also for bigrams and trigrams)

3.1 Experiments

Starting with the code provided, new features has been added to check whether the results improved. The final feature set is the one that provides the best result, previously explained.

3.2 Results

The average results obtained with the features selected are shown in Figure 1:

Features	Precision	Recall	F1
Selected	0.95	0.59	**0.65**
PosTag / Lemma	0.87	0.52	0.60
All features	0.86	0.38	0.49

Figure 1: Results of CRF using different features

As we can see, the metrics used to evaluate the different results are precision, recall and the F1 score. Precision is the fraction of relevant instances among the retrieved instances. Recall is the fraction of relevant instances that have been retrieved over the total amount of relevant instances. Finally, the F1 score is the harmonic average of precision and recall.

Contrary to what was supposed, adding pos-tag, lemma or bigrams as features, does not improve the results, as well as using a huge bunch of features doesn't help either, as the model can un-learn instead of learn, which is what we are interested in.

4 Task 9.2: Extraction of drug-drug interactions.

Two different approaches have been used for this task. The first one is a Convolutional Neural Network (CNN). The second one is a Naive Bayes model using pre-trained word embeddings.

The classification of each drug-drug interaction is done according to one of the following four types: advise, effect, mechanism, general interaction. In the case of no interaction a new class null is created.

4.1 Convolutional Neural Network

The XML files has been pre-processed so that, for the train and the test sets, the names of the drugs are changed for DrugX and DrugY, as the interest of each sentence, for each pair of drugs, is the context in which the drugs are in, not the drugs themselves. Some drugs are more than one token long, and some pre-process has been done so that the whole drug name is substituted by DrugX or DrugY. The modified sentences have been stored in a list of sentences, where each sentence is a list of words composing that sentence.

To be able to feed the CNN with the training sentences as input, a vocabulary has to be created, where and integer is assigned to each different word that composes it. In addition, all the sentences need to have the same length, so a padding is done, to fill the shorter sentences with zeros. A 10% of the largest sentence's length is added to the maximum length of the sentence to cover the case where a test sentence is larger that the calculated maximum length of the training set.

The architecture for the Convolutional Neural Network is shown in Figure 2:

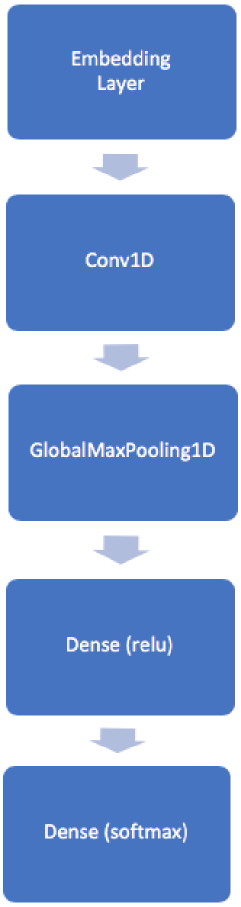

Figure 2: CNN Architecture

4.1.1 Hyper-parameters

The hyper-parameters used to train the Network are the following:

- max_words = vocabulary words
- embedding_dims = 300
- filters = 128
- kernel_size = 5
- hidden_dims = 128
- batch_size = 64
- epochs = 10

4.1.2 Results

The average results obtained are shown in figure 3:

DDI Type	Precision	Recall	F1
mechanism	0.4897	0.5497	0.5179
effect	0.5174	0.4944	0.5057
advise	0.5381	0.5747	0.5558
int	0.8182	0.1875	0.3051
Overall	0.5909	0.4516	**0.5119**

Figure 3: Results using CNN

4.2 Naive Bayes

In order to compare the results achieved with the CNN, a Naive Bayes model has also been used. The XML files for the training set and the test set are pre-processed parsing the sentences and eliminating stop words.

A reduced version of the GloVe embedding model is loaded to use in further steps. In doing so, we reduce computing time while only taking into account relevant words.

The set of features used in this model are only two. The first one is the relevant verb for the drug-drug interaction in question. To compute it, we search for all the verbs in the sentence using pos-tagging. After that, we choose the one that is closer to one of the drugs and we convert it to a vector. The second feature is the average embedding of the sentence containing the interaction.

4.2.1 Results

Since we only have two features, the experimentation was simple. We tried them individually and combined. The results are shown in figure 4:

Features	Precision	Recall	F1
Verb	0.119	0.4736	0.1901
Sentence	0.1379	0.3912	0.2039
Verb+Sentence	0.1532	0.5169	**0.2363**

Figure 4: Results using Naive Bayes with different features

The best result is obtained when using the combination of both features with a F1 score of

0.2363. The decomposed evaluation is shown in figure 5:

DDI Type	Precision	Recall	F1
mechanism	0.1717	0.5132	0.2573
effect	0.1926	0.475	0.274
advise	0.1643	0.5792	0.256
int	0.0842	0.5	0.1441
Overall	0.1532	0.5169	**0.2363**

Figure 5: Results using NB by type of interaction

5 Conclusions

After performing different approaches to solve both tasks, we can say that the key step for these tasks and most Machine Learning problems is to find the best combination of features and model that suits our initial goal.

For the first task, CRF was used. The interesting property of CRF is that it takes into account the contextual information when doing the prediction which helps a lot if the task in question is some kind of variation of Name Entity Recognition.

For the second task, CNN was the way to go. Considering the results obtained, there is a lot of room for improvement. One approach that could be taken is using a biomedic pre-trained word embedding model instead of using a general-use one.

Some other approaches could be done (like different types of Neural Networks or combinations of them), but the aim of the work was not trying a lot of approaches, but find some that provided acceptable results.

6 Future Work

Taking the three approaches for the two tasks as a basis, some experiments can be done, trying to find better or complementary features, for the cases of CRF and Naive Bayes, as well as mofify the hyperparameters or add more layers in the Convolutional Neural Network.

7 Github Repository

In https://github.com/ovalls/ahlt you can find the scripts and results for the NE Recognition and DD Interaction:

- NER_AHLT.py: code for the Name Entity Recognition, where the extraction of the features is done.
- train-crf.py: CRF.
- DDI_AHLT.py: code for the Embeddings and CNN for Drug-Drug Interaction.
- DDI_AHLT_NB.py: code for the Embeddings and Naive Bayes for Drug-Drug Interaction.
- split_train.py: script to create the validation data with the 10% of the training dataset.
- dron_clean.csv: list of the Drugs from the Drug Ontology from Bio Portal.
- data: folder with the training, validation and test datasets

References

[1] ClearTK. *Feature Extraction Tutorial.* [https://cleartk.github.io/cleartk/docs/tutorial/feature_extraction.html]

[2] Austin G. Walters. *Convolutional Neural Networks (CNN) to Classify Sentences.* [https://austingwalters.com/convolutional-neural-networks-cnn-to-classify-sentences/]

[3] Biomedical natural language processing. *Natural Language Processing Laboratory (NLPLab).* [http://bio.nlplab.org/#word-vectors]

[4] Jeffrey Pennington, Richard Socher, Christopher D. Manning. *GloVe: Global Vectors for Word Representation.* [https://nlp.stanford.edu/projects/glove/]