Silver Standard Machine Learning for Protein-Cell Death Interactions Text Mining in PubMed Abstracts and Articles

Hannes Berntsson
Computer Science Student, Faculty of Engineering
Lund University
Ole Römers väg 3
223 63, Lund, Sweden
dat15hbe@student.lu.se

June 10, 2019

Abstract

Relation extraction for bio-medical text mining is an essential part in processing the huge amounts of work published in the field of bio-medicine. This paper is part of a larger project in its very initial phase. The project aims to create a tool for bio-medical text mining that can map out and answer what and how different proteins/genes affect lysosomes and cell death. I present a couple of baseline machine learning models for binary relation extraction for this purpose. Results on training and testing were done on the BioInfer binarized gold standard corpus along with a silver standard corpus produced by the other teams in the project using PubMed data. The two models demonstrate an F-score of **58%** and **71%** respectively when trained and tested on tenfold cross validation. A similar F-score of **66%** is also shown for the silver standard corpus by one of the models. There is an un-adressed but discussed issue with training on randomly generated examples of "no relation" for a silver standard corpus. After training a model on the generated silver standard corpus the model is used to predict relations throughout a handful of entity-tagged medical abstracts from PubMed. The results show a satisfactory high amount of *no relation* predictions proving that the approach for the project has actual practical potential. There are also issues highlighted with among other things scaling (memory wise) binarized relations.

1 Introduction

1.1 The Main Project

There are currently over 15-million published articles in the bio-medical field, around 800,000 of these articles are published on the subject of cell death and lysosomes. Filtering and processing these articles is often decades of work just to form relevant hypotheses on the subject. This project has the end purpose of automating the process of locating which proteins/genes effect cell death/lysosomes (notably in chains of effects between several proteins to an effect on cell death). The initial iteration of the project is divided into three sub-projects with different teams:

1. **Identify and tag proteins and other entities** throughout PubMed articles and send the tagged corpus to the second sub-project.

   A. P. Sjövall & E. Holmström

2. **Tag relations using strict grammatical rules** giving a data-set with very low recall but near 100% precision which is the sent to this sub-project.

   O. Nordengren & V. Lundqvist

3. **Train a machine learning relation extractor** using the binarized Bio-Infer corpus and more importantly the data-set from the second sub-project. Apply the trained model on the entity-tagged PubMed data to extract relations with a much higher recall but lowered precision.

   The sub-project this article describes.

Hypothetically a silver standard training set would allow for all examples being specifically on protein to protein and protein to cell death/lysosomes relations from the actual corpus the model is to predict on. This could make the model more accurate for those specific relations by minimizing any wrong predictions caused by training on other examples of relations. It could also likely produce better overall results to train on a silver standard corpus compared to any existing gold standard one because it would be comparable to cross-corpus testing which is shown to produce much lower scores (Airola et al., 2008). The goals for this sub-project was to build a baseline machine learning model, use it to predict new relations in the PubMed corpus and highlight any issues and improvements for future iterations of the project.

We chose to only look at interactions between pairs of entities (binary relations) bound by sentences. Looking at binary relations is the most prominent approach when it comes to relations in information extraction (IE) (Pyysalo et al., 2008). Binary relations as an approach has also been proven in practice (Alex et al., 2008) although it is shown that IE might not completely remove the effort needed to dig through huge bodies of work. A lot has also been done by the NLP community the last decade to produce more refined models that go beyond binary relations and treat it more as a Named-Entity-Recognition (NER) problem (Björne et al., 2009; Björne and Salakoski, 2015). Looking solely at relations existing within a sentence is also very much prominent in state of the art bio-medical IE and protein-protein-interaction (PPI) (Peng and Lu, 2017; Peng et al., 2018), this despite the obvious issues caused by co-references and similar concepts meaning some relations can't be fully comprehended within a single sentence.

1.2 This Sub-Project

Because of time constraints, limited experience and the short term goal I did not aim to re-create any state of the art models for binary relation extraction. Instead a baseline kernel based SVM solution was tested along with a simple bigram/trigram bag-of-word (BoW) deep learning model was employed. The better performing model was then to be trained and tested on the silver standard corpus produced by Nordengren and Lundqvist and finally used to predict relations troughout the entities tagged in the PubMed abstracts by *Sjövall and Holmström*. The hope is that these tags can be used to map out a large part of PPI instances (along with interactions with cell death/lysosomes) as well as connect them for a tool that bio-medical scientists can use to find potentially interesting hypotheses. Important to repeat and note is that the implementations in these three sub-projects are just the very inital proof-of-concept implementations and that future iterations will replace and build on our results.

The BioInfer binarized gold standard corpus (Pyysalo et al., 2007) was chosen to work off of while the first iteration of the silver standard corpus was produced and to use as a way to test and potentially help train the models. BioInfer was particularly attractive as a corpus as it has been highly used in the NLP community in the last decade. Additionally it allows future iterations to look at bigger than binary relation complexes as well as full relation tagging (type, direction and trigger) in a NER fashion. It also allowed for the binary relations to be tagged as not only "positive" or "false" but rather "no-interaction", "positive interaction" and "negative interaction" to allow for mappings such as "Protein A induces cell death" or "Protein B prevents Protein A production". This kind of mapping is key to be able to fully map how studied proteins actually effect cell death and lysosomes. The silver standard corpus was produced in the same manner. Important to note is that "no-interaction" examples are not given in either corpus but the BioInfer corpus is fully tagged which means that "no-interaction" are *truly* any combination of entities not tagged to have one. The silver standard corpus on the other hand contains many thousands of un-tagged combinations of entities that may or may not be interacting.

2 Methods

The examples are defined as one **sentence**, two tagged **entities** where order matters for the relation and a type of **relation** either *positive*, *negative* or *no relation*. The models train on and predict the relation type of the examples.

Any and all Natural Language Processing (NLP) tags and features are produced using the ScispaCy,

(Neumann et al., 2019) *en_core_sci_md* models.

Statistics of the BioInfer corpus and silver standard corpus (SSC) can be seen in Tabel 1 where the validation sets consist of randomly picked examples from the corpora.

Precision, Recall and F-Scores are calculated on the weighted average over classes in randomized 10-cross validation according to the test/validation sized specified in Tabel 1. The confusion tables were also produced by taking averages from the same cross validation runs.

The SVM model was fitted using full training set batches. 25 epochs with a batch size of 20 was used to fit the data to the BoW Neural Net Model.

2.1 Support Vector Machine with NLP Tags

This model consists of **tokens** from the sentence and their corresponding **Part-of-Speech** (PoS) tags and the syntactic **dependency** (Dep) in order accordingly:

First Entity: One token before the entity and the following three tokens.

Second Entity: Three tokens before the entity and one token following.

The features are vectorized using the *scikit learn* (sklearn) DictVectorizer[1]. The classifier used to fit the features is a c-support vector machine from sklearn[2]. Running with the parameters: `kernel='poly', gamma=1, C=1, degree=2`. Meaning that the data is separated using a non-linear hyper-plane, gamma determines how perfectly it tries to fit the data, C governs how much error is accepted to allow a smooth fit, degree is simply the polynomial degree of the kernel. These values were chosen mainly by testing the performance and adjusting accordingly.

2.2 Deep Learning with BoW

This model is summed up as a simple 3 layered dense neural net training on entity replaced sentences using a BoW vectorization with the 10000 most common bigrams and trigrams.

2.2.1 Entity Replacement BoW

The two tagged entities in each example get replaced with the tokens "ENTITY1" and "ENTITY2". An example from BioInfer where "*alpha-catenin*" is the first entity and "*beta-catenin*T-cell factor*DNA complex*" is the second:

"*alpha-catenin inhibits beta-catenin signaling by preventing formation of a beta-catenin*T-cell factor*DNA complex.*"

would be converted into:

"*ENTITY1 inhibits beta-catenin signaling by preventing formation of a ENTITY2.*"

This in turn makes the most common bigrams/trigrams for the BoW consist of bigrams/trigrams such as: "*ENTITY1 inhibits*", "*blocked by ENTITY1*", "*ENTITY2 was inhibited*", "*ENTITY1 induces ENTITY2*" etc.

The BoW vectorization on the 10000 most common bigrams and trigrams is in turn produced using the sklearn countVectorizer [3].

2.2.2 Neural Net and Model

The net itself is built using the Keras machine learning API [4]. The very simple net consists of three dense layers: The input layer uses *rectified linear unit* activation (Hara et al., 2015) and an output dimension of 100, the middle layer uses the softmax activation function (normalized exponential function) and 100 as output dimension. The output layer uses sigmoid activation (Sibi et al., 2013) for the multi-class output of dimension size three.

Optimization is performed using Adam (Kingma and Ba, 2015) (*first-order gradient-based optimization of stochastic objective functions*, as opposed to stochastic gradient descent).

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.DictVectorizer.html`

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`

Corpus	**Test Ex.**	**Val. Ex.**	Tot. No Relation	Tot. Pos	Tot. Neg	Baseline Guess
BioInfer	4476	791	2495	2612	160	49,59%
SSC	7123	1258	3451	3208	1722	41,18%

Table 1: Corpora Statistics (with randomly generated *no relation* examples)

Pred.\True	no	Pos.	Neg.
no	**260**	108	5
Pos.	88	**300**	7
Neg.	5	8	**9**

Table 3: Confusion matrix of the BoW neural net model predictions on **BioInfer**.

Pred.\True	no	Pos.	Neg.
no	**473**	126	77
Pos.	121	**311**	40
Neg.	70	46	**143**

Table 4: Confusion matrix of the BoW neural net model predictions on **SSC**.

3 Results

3.1 Models

The NLP Feature SVM model ran early on in the project and produced an F-score of **58%**, seen in Table 1. As this model was neglected early on very few metrics were generated for it in general and it was never used on the SSC.

The BoW Neural Net model ran on both the BioInfer corpus and the generated SSC, in both cases 10 fold random cross validation was used to generate the metrics and confusion tables seen in Tables 2, 3 and 4, displaying an average F-score of **71%** and **66%** respectively.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

[4] https://keras.io/getting-started/functional-api-guide/

no interaction	**positive**	**negative**
100462	29	13

Table 5: Predictions on entity-tagged PubMed data.

3.2 PubMed Predictions

For the experimental predictions on the PubMed data the BoW Neural Net model was trained on the full SSC (not divided for validation). The predictions were made on 100,504 examples of tagged entity pairs in sentences generated from PubMed abstracts. As each sentence can contain up to and over 25 tagged entities 100,000 examples doesn't come from very many abstracts or sentences. A file of around 15,000 abstracts with tagged entities gives over 27 million entity pair examples.

The statistics of the predictions on the PubMed data can be seen in Table 5.

4 Conclusions

4.1 Model Performance

The performance of the NLP feature SVM model was studied much less as focus was shifted quickly to the BoW neural net approach meaning it stands with a lot less metrics. Nevertheless it showed an F-Score of **58,46%** on the binarized BioInfer corpus which shows it trains for the task notably but not very well seeing as baseline guessing for the corpus sits at **49,59%**. Mainly this points toward that a SVM model potentially will need more finesse to get satisfactory results for this task compared to a deep neural net learning approach.

While it is very important to not that they are not directly comparable (targets, binarization and over-

	BioInfer			SSC		
	Recall	**Prec.**	**F**	**Recall**	**Prec.**	**F**
NLP Feature SVM	*	*	**58,45**	*	*	*
BoW Neural Net	70,70	71,02	**70,64**	65,89	66,09	**65,90**

Table 2: Metrics of the two models on BioInfer and the Silver Standard Corpus. *The SVM model wasn't as thoroughly tested as it showed less promise early on in the project.

all definitions of the problem differ); the BoW Neural net model showed results on the BioInfer corpus (Table 2) close to what other much more complex models have presented in the past with an F-score of **71%**, compared to eg.: **77%** (Björne et al., 2009), **64%** (Airola et al., 2008).

Performance on the SSC was similarily high with an F-score at **66%** especially considering it has a better (lower) baseline guess (**41%**) compared to the BioInfer corpus (**50%**). This shows that the SSC is possible to train on with a similar difficulty as an accepted gold standard for the same task.

4.2 Application

Using the model (trained on the SSC) to predict on the PubMed abstracts gave: 29 examples as *Positive* and 13 as *Negative* examples while tagging 100462 examples as *no relation*. This is a lot of examples tagged as *no relation* but seeing as 100,000 examples come from only a handful of abstracts it's actually what should be expected. The distribution of the results in Table 5 themselves are very promising for this approach being a possible help in mining biomedical texts.

A large part of why this approach is so attractive is that it could be used to perform NLP taks on any other big corpora without good gold standard data connected to them. Obviously a gold standard corpus based on PubMed articles for our specific task would be preferable but that is much more expensive and many times not an option for an experimental NLP machine learning project.

4.3 Issues

The biggest problem with this approach, to which I do not present a solution, is the problem with of how to generate examples of "*no relation*". As the silver standard training set is generated with a tiny recall score (most examples of actual relations are not included) the generation of random "*no relation*" examples will give a set of examples that actually contain many examples of relations. In other words the actual training set with the generated *no relation* examples doesn't actually have the high precision sought after for this method to be fully effective.

As shown in the statistics in Table 1 and the confusion matrix in Table 4 BioInfer has nearly no examples of *negative* relations. This doesn't only make it very hard to train for that classification but it also questions how well that tag is defined and used in the BioInfer corpus in relation to how this project have defined and interpreted it.

Another possible issue, especially when looking at binary relations from tagged entitites is that the amount of examples scales pretty badly. Processing 15,000 abstracts would mean handling over 20 million examples and seeing as there are nearly 800,000 articles on cell death alone that quickly means a lot of processing just to divide the data enough that memory wouldn't be an issue.

4.4 Improvements

The most obvious improvement for future iterations of this project is making more sophisticated models. The closest thing to being done for a future iteration is a neural net model using word-embeddings, tagging of the entities and a reconstruction of the sentences by following the shortest path in the dependency graph of the sentences.

It would likely be beneficial to eventually move away from the binary definition of the task and follow the NER/graph approach of Björne et al. (2009 & 2015) and their project. A very possible approach would be to try and build on and modify their solution and implementation.

Creating an ensemble of an SVM and a neural net could potentially improve the learning capabilities of any future models. Trying out ensemble with models trained on gold standard corpora could potentially boost the easier examples as well.

If the models used to predict the relations is properly paired with the first sub-project by *A. P. Sjövall & E. Holmström* (the entity tagging) it would allow for the prediction to be run and made on a completely unprocessed plain-text of abstracts and articles making it much easier to employ and integrated into other tools and applications.

The problem of having to handle a very large amount of examples would be solved if the models and overall design of the solution switched to a NER-tagging-like solution that simply processes the plain text, finding entities and trying to find any actual relations as it goes. Compared to actually holding hundreds of altered versions of the same sentence throughout the whole prediction process, across thousands of sentences, there would be much

smaller memory issues.

Acknowledgments

I want to thank Sonja Aits for starting, organizing and mentoring this project and wish her the best of luck if she continues with it. I also thank Pierre Nugues for his mentoring and supervision throughout. Biggest of thanks to Markus Klang for his constant help and advise these two months. For an enjoyable project I thank all the other student project members and wish them luck with the rest of their studies.

Finally I must thank the Faculty of Engineering and Lund University for holding this project course and enabling this work.

References

A. Airola, S. Pyysalo, J. Björne, T. Pahikkala, F. Ginter, and T. Salakoski. 2008. All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *Proc. BioNLP 08 ACL Workshop*, 9(Suppl 11):S2.

B. Alex, C. Grover, B. Haddow, M. Kabadjov, E. Klein, Roebuck S. Matthews, M., R. Tobin, and X. Wang. 2008. Assisted curation: Does text mining really help? *Biocomputing '08*, pages 556–567.

J. Björne and T. Salakoski. 2015. Tees 2.2: Biomedical event extraction for diverse corpora. *BMC Bioinformatics 16*.

J. Björne, F. Ginter, J. Heimonen, S. Pyysalo, and T. Salakoski. 2009. Learning to extract biological event and relation graphs. *NODALIDA 2009 Conference Proceedings*, pages 18–25.

K. Hara, D. Saito, and H. Shouno. 2015. Analysis of function of rectified linear unit used in deep learning. *IJCNN*, 2015.

D. P. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*, San Diego, 2015.

M. Neumann, D. King, I. Beltagy, and W. Ammar. 2019. Scispacy: Fast and robust models for biomedical natural language processing. *arXiv:1902.07669*.

Y. Peng and Z. Lu. 2017. Deep learning for extracting protein-protein interactions from biomedical literature. *W17-2304*, BioNLP 2017:29–38.

Y. Peng, A. Rose, R. Kavuluru, and Z. Lu. 2018. Extracting chemicalprotein relations with ensembles of svm and deep learning models. *Database*, Volume 2018.

S. Pyysalo, F. Ginter, J. Heimonen, J. Björne, J. Jarvinen J. Boberg, and T. Salakoski. 2007. Bioinfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8:50.

S. Pyysalo, A. Airola, J. Heimonen, J. Björne, F. Ginter, and T. Salakoski. 2008. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics 9*, 3:1–11.

P. Sibi, S. A. Jones, and P. Siddarth. 2013. Analysis of different activation functions using back propagation neural networks. *Journal of Theoretical and Applied Information Technology*, 47(3):1264–1268.