Detection of Biological Entities

Emil Aminy
Lund University

Petter Berntsson
Lund University

Abstract

Named Entity Recognition (NER) in bio-medical texts provides an efficient way for researchers to comb through huge amounts of published works to help accelerate research about cell death. This paper is a sub-project to a larger project. In the larger scope the goal is to tag bio-medical entities, map out relations between these entities and how these relations work. This paper focuses on the tagging of proteins. This paper presents 2 different methods of NER - Machine Learning and Dictionary-based tagging. This paper also presents 2 ways of improvement by combining these methods, by their union and intersection. The purpose of this is to gain better recall and precision respectively.

1 Introduction

Within the bio-medical fields of science there exists an enormous amount of information in the form of scientific articles, with more being produced every day. These articles can contain vital information that could possibly lead to major breakthroughs, however the massive quantity of information makes it nearly impossible for a human to effectively do so. Even when focusing on a specific subject within the field the number of articles is too large for any human to feasibly locate, read and understand.

However, sifting through large amounts of data and creating relations is something that computers excel at. This paper is part of a project which aims at creating a tool that will aid humans with just that. A tool that could aid researches in finding related articles which would allow researchers to spend more of their on other things. To accomplish this natural language processing can be used.

1.1 Natural Language Processing

Natural language processing, or NLP, is a field at the conjunction of computer science and linguistics which focuses on the processing and analysing of natural language data by computers. One task NLP can be used for is called information extraction (IE), where NLP is used to automatically extract data from unstructured data. For the purposes of creating relations between scientific articles the interesting methods of IE are named entity recognition and relation extraction.

Named entity recognition (NER) is the task of identifying and classifying entities in a text. Within the context of this paper an entity is a protein, disease or medical compound.

Relation extraction (RE) is the task of detecting semantic relations between two entities in a text. A relation in this case could perhaps be *protein A inhibits disease B*, where inhibits would be the relation that connects the two entities.

Once these two tasks are accomplished, a database of articles with identified entities and relations can be created. This database can be further used to create a network of relations between entities found in multiple articles, creating chains of entities that could possibly be related. For example, if one article talks of how *A causes B* and a second article talks of how *B counteracts C*, then perhaps there exists a relation between *A* and *C* which could be studied further by human researchers.

Since this is quite a large undertaking, this paper focuses solely on the NER of proteins. Two methods will be used in conjunction to identify proteins, one utilising a dictionary and one a machine learning model.

1.2 Dictionary

The dictionary method of identifying protein names consists of collecting as many protein names and synonyms that can then be used to search for matches in an article text. This method relies heavily on the quality of the dictionary, collecting a comprehensive list of protein names is therefore a large part of this method. In this paper, two protein databases were utilised to build the dictionary.

The first database used was UniprotKB/Swiss-Prot, which is a manually curated database of approximately 500 000 proteins maintained by UniProt. The proteins within come from multiple species (Uniprot, 2018).

The second database was HGNC, short for HUGO Gene Nomenclature Committee (HUGO is short for Human Genome Organisation), is a fully curated database of human proteins, no other species are included. Due to being limited to only humans it is relatively small with only approximately 40,000 proteins (HGNC, 2019).

1.3 Machine Learning

Machine learning (ML) is a field within computer science which combines algorithms and statistics to create systems that can perform tasks without specific instructions. In this paper a sub-type of ML algorithms referred to as semi-supervised learning algorithms will be utilised.

Semi-supervised learning (SSL) is a combination of supervised learning (SL) and unsupervised learning (UL). All three algorithm types require training before use, where they are shown large quantities of data which they then analyse to build their statistical models. The difference between them is that SL receives training data which is annotated, i.e. contains an input and a desired output. Its training data essentially comes with the correct answers. This is something UL has to make do without. It only receive input data and has to find patterns or structures within the data by itself (Zhu, 2009).

This difference might seem trivial but is actually quite impactful. Creating annotated training data is a time consuming task which requires a human to manually review the input data. Being able to forgo it saves time, though it might also result in unpredicted results since the model can't be instructed what to output (Zhu, 2009).

SSL combines these two algorithm types to reduce the need for large amounts of annotated data while still allowing the models to be taught what to look for. One way of implementing SSL is to train models on massive amounts of unannotated data to create a model. If done properly, this model should have a clear understanding of its data but no real goal for this understanding. This is called pre-training and requires a lot of computational power (Zhu, 2009).

Once pre-trained, the model can receive some further training on a comparatively small amount of annotated data which specialises the pre-trained model on specific tasks related to the data. This is called fine-tuning. Since the pre-training and the fine-tuning does not have to happen on the same computer, pre-trained models can be uploaded to the internet and used by anyone to fine-tune on specific tasks (Zhu, 2009).

In this paper, a pre-trained model called BioBERT will be utilised. This model is itself based on another pre-trained model named BERT. BERT is en English language model published by Google which was trained on the entirety of English Wikipedia. BERT was then further pre-trained by a team of researchers from Korea University and Naver Corp. on biomedical corpora consisting of over 20 billion words to create BioBERT (Devlin et al., 2018; Lee et al., 2019).

2 Method

2.1 Dictionary

As mentioned, the dictionary approach will utilise two databases to create a dictionary of protein names, synonyms and abbreviations. The nature of these databases introduce some complications that had to be dealt with. First of all, a single protein can have multiple names, some may even have tens of names. This has to be handled and made sure that all protein synonyms refer back to the same protein entity. Unique identifiers were given to each protein entity and all synonyms were connected to said identifier.

Furthermore, the fact that two separate databases also needed to be handled. Luckily each database had their own system of unique identifiers and cross-referenced to each other, making the merging of SwissProt and HGNC proteins easier. Some overlap existed though where protein synonyms could be listed in both databases. Duplicate names had to therefore be handled as well.

The complete dictionary, even after being merged properly, still contained a lot of ambiguity. This was caused by the fact that the same proteins from different species in some cases have identical names. Additionally, a lot of ambiguity existed between protein names and English words. This was however expected. The purpose of the two pronged approach of the NER is to have the combined results augment the separate results. How this works is discussed later in the paper.

Once complete, the dictionary could be used to search for words in texts. This was done with a dictionary tagger written by Marcus Klang. It takes the dictionary and the text as input and then finds all matches in the text. It does this following two principles; longest match and dominant right match (Klang, 2019).

The longest match principle states that the longest matching word should take priority over shorter matches. For example, a dictionary containing *cat* and *locate* would both match the word *locate* since it the word *cat* can be found in *locate*. The principle simply states that the longer word, i.e. *locate*, should take priority. It is a common text-matching principle and usually yields better results.

The dominant right match principle is a way to handle match overlap. If you have the sentence *one two three* with two matches, *one two* and *two three*, you can see that they both overlap on the word *two*. Longest match does not apply here since it's no longer a sub-string. Dominant right states that the rightmost match takes priority. This is an arbitrary choice and could have also been dominant left, it's simply there to handle overlap.

With these two principle Klang's dictionary tagger was able to run the dictionary on the test from the JNLPBA and yield results (Huang et al., 2019).

2.2 BioBERT

BioBERT, as discussed, is an open-source pretrained ML model specialised on English texts within the bio-medical fields. It is freely available on GitHub and allows for fine-tuning which can further specialise it. Since the goal was to only identify proteins, the model was fine-tuned on the training and validation set from the JNLPBA corpus for 24 hours on an Intel i7 3770k CPU (3.5 GHz) (Huang et al., 2019).

Since the BioBERT is designed to understand words in a context, the goal was for it to better understand ambiguous words, being able to identify what species a protein belongs to or whether an ambiguous words i a regular English word or a protein. Additionally, the hope was for BioBERT to be able to identify new proteins names, which it had not encountered before in training nor could be found in the dictionary.

Once fine-tuned BioBERT was applied on the test set from the JNLPBA corpus which yielded result (Huang et al., 2019).

2.3 Combined results

The purpose of the two method approach was to create a combined result that would hopefully improve on the separate results. The idea behind it was that the dictionary approach would yield results with high recall but low precision while the BioBERT approach would yield mediocre recall and high precision. The two results could then be combined to create a union and an intersection of the matches found. The union would therefore be a combination of all matches found either by the dictionary or BioBERT or both, while the intersection would only contain matches both methods agreed on. The goal being that the precision of the intersection and the recall of the union would both respectively be higher than the individual results.

3 Results

3.1 Dictionary Tagger

The dictionary approach resulted in a recall of 30%, precision of 13% and then the harmonic mean (F1-score) of 18. The dictionary approach used the databases of HGNC and Uniprot. NCBI was not included. Results are represented in the table below.

3.2 BioBERT

The ML approach yielded a recall of 71%, precision 61% and F1-score of 66. The complete results are represented below, where BioBERT results are represented.

	Dict	ML	∪	∩
Recall(%)	30	71	75	44
Precision(%)	13	61	27	68
F1-Score	18	66	40	53

The following table presents the difference between the baseline machine learning approach, with the union and intersection approach:

	∪	∩
Recall	+4	-27
Precision	-34	+7
F1-Score	-26	-13

4 Conclusion

4.1 Performance of Union and Intersection

The dictionary approach acts as a supplement to the machine learning approach. The union and intersection method both give a notable improvement with their respective strengths. The union gives a better recall while lowering precision. Likewise the intersection improves the precision and reduces the recall. Both of these results were expected.

While at first glance, they may seem worse than the ML results they do actually have a use. For instance, the union shows that the recall has increased. This means that it is catching more of the entities found in the text. It does so by lowering its standards for classifying a word as an entity, but it does so in a manner that allows it to find a larger percentage of the total population. If it were to be further worked on and achieve slightly higher recall percentages, it could be a very efficient tool to use as a rough first filter. This could for instance be used to filter away entire sentences that do not contain any entities whatsoever allowing for the creation of more entity dense texts. Texts that could be very useful for testing the capability of future ML models.

Similarly, the intersection results have a higher precision, meaning they have increased their standards for what could be considered an entity, at the cost of missing a higher percentage of them. Again, this result, with some more work done to it, could create a very precise algorithm that could be utilised to generate silver standard training data for ML models. This could be an incredible asset that would allow for more varied training data which in turn would result in better conditions for training a more accurate ML model.

Overall the results do show the merits of a two method approach. There is a clear use for algorithms that specialise in either having high precision or high recall, and the results do point towards this being the case with the two methods. However, it does also show that a pure ML approach does achieve quite high results by itself and should perhaps receive some more time and development.

4.2 Issues

The project was constrained by time and experience. The issues brought up were not a question of *if* they could be solved, there are definitely ways, but rather a question of *is there time*? The main issue relating to machine learning was the computational power available. A local super-cluster was made available, but issues relating to parallelization and foreboding trial runs made it so that fine-tuning ran on PCs, specifically on older CPUs. The model therefor was not fine-tuned efficiently or for very long, not even on GPUs.

Similarly, the dictionary approach was constrained by computational power. As mentioned, two databases were used in the dictionary. Originally there were intentions of using a third database, called NCBI, which was 2 orders of magnitude larger than the other two dictionaries combined. However, it turned out the NCBI database was too large to efficiently merge in with the other two dictionaries due to there not being enough RAM on available hardware. Modifications were made to the merging algorithms, however this resulted in the merging requiring taking too long. The processing of the NCBI database required either too much RAM, or took too long when using less RAM.

4.3 Improvements

The poor results of the dictionary clearly show the need for a larger dictionary. Since the NCBI database was not included, only speculations can be made about the hypothetical results. As discussed previously, the purpose of the dictionary was to get a large recall which could then be used with the more precise ML method.

If the recall in the dictionary tagging were to, there would be more for the intersection to intersect with. With a greater intersection, the more precise ML method would have more results to compare with which in turn would probably improve precision.

With a greater union, the likelihood of totally covering previously partially matched entities would be greater. Recall in the union would improve. Precision might decline, since a larger set of lower precision matching could outweigh the higher precision matching garnered from unions made over entities. It all depends on the change of ratio between the intersection and union when the dictionary size increases.

Considering the short fine-tuning of the BioBERT model, results were satisfactory. The project should be viewed as a pilot trial, rather than a definitive result. The model can be fine-tuned more effectively, for example by utilising GPU's, and for longer (in terms of epochs). Combining a more adeptly trained model with a larger dictionary would definitely produce higher metrics, and given enough time would be very doable. Especially since BioBERT has been shown to have significantly higher results when trained properly (Lee et al., 2019).

References

J. Devlin, M. Chan, K. Lee, and K. Toutanova. 2018. ”open sourcing bert: State-of-the-art pre-training for natural language processing”. [Online]. Available: https://arxiv.org/pdf/1810.04805.pdf. [Accessed: 16- Jan- 2020].

HGNC. 2019. ”frequently asked questions”. [Online]. Available: https://www.genenames.org/help/faq/. [Accessed: 16- Jan- 2020].

M. Huang, P. Lai, R. Tsai, and W. Hsu. 2019. Revised JNLPBA corpus: A revised version of biomedical NER corpus for relationextractiontask”. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1901/1901.10219.pdf [Accessed: 16-Jan-2020].

M. Klang. 2019. ”dictionary-tagger”. [Online]. Available: https://github.com/Aitslab/BioNLP/tree/master/marcus/dictionarytagger. [Accessed: 16- Jan- 2020].

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. ”biobert: a pre-trained biomedical language representation model for biomedical text mining”. [Online]. Available: https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/btz682/5566506. [Accessed: 16- Jan- 2020].

Uniprot. 2018. ”uniprotkb”. [Online]. Available: https://www.uniprot.org/help/uniprotkb. [Accessed: 16- Jan- 2020].

X. Zhu. 2009. ”semi-supervised learning”. [Online]. Available: http://pages.cs.wisc.edu/~jerryzhu/pub/SSL_EoML.pdf. [Accessed: 16- Jan- 2020].