Recognition of genetic mutations in text using Deep Learning

Pedro Matos  
DETI/IEETA, Universidade de Aveiro  
Aveiro, Portugal

Sérgio Matos  
DETI/IEETA, Universidade de Aveiro  
Aveiro, Portugal

ABSTRACT
Knowledge about genetic mutations and their impact on the organism is continuously being produced and communicated through scientific publications. This information is then collected by curated databases and integrated in structured knowledge resources, facilitating its discovery and reuse. To aid this work, information extraction methods are increasingly being integrated in the database curation pipelines.

This work describes an information extraction method based on deep neural networks for the recognition of mutation mentions in literature abstracts. When applied to the tmVar dataset, the character-based model reached an F-measure of 0.874. This result was achieved without the use of knowledge resources or any handcrafted features.

CCS CONCEPTS
• Computing methodologies → Machine learning; Supervised learning by classification;

KEYWORDS
Deep Learning, NER, Neural Networks, Bi-LSTM-CRF, Artificial Intelligence

1 INTRODUCTION

The construction and updating of knowledge resources is an utmost requirement in the biomedical field, given the large number of interconnected entities such as diseases, chemicals, genes, or genetic mutations, and the numerous scientific articles where this information is described. Information extraction methods that identify mentions of these entities and their relations in large text corpora are therefore vital for aiding this work, and various tools have been proposed with this objective [10].

Deep learning is a sub-area of automatic learning that attempts to model complex structures in the data through the application of different neural network architectures with multiple layers of processing. These methods have been successfully applied in areas ranging from image recognition and classification, natural language processing, and bioinformatics. In this work, we explored methods for named-entity recognition (NER) in text using techniques of deep learning to identify genetic mutations. The objective of this work is to create a NER system that can learn from the corpora available without relying on hand-crafted features.

Recurrent Neural Networks (RNNs) are a family of neural networks which specialize in processing sequences. These neural networks use parameter sharing to apply the model to sequences of different lengths and generalize across them. Sharing the weights across all words in the sentence enables the network to learn the rules of the text only once and with fewer parameters [3]. In RNNs, the connections from the input and from the hidden layers of the previous word are parameterized and are optimized during the training phase.

By combining two RNNs, one which moves forward from the start of the sequence and another which moves backwards from the end, a bidirectional RNN is obtained. This structure enables the joint RNN to combined dependencies from previous and following inputs. Such is the strategy employed by Mavromatis for disease extraction [6].

Most modern RNN implementations use the Long Short-Term Memory (LSTM) cell [4] to compute a neuron’s output and in the biomedical domain this is no exception [1, 5, 9]. These cells are probably one of the most significant developments in RNNs [3].

RNNs can also be combined with other machine learning methods. A powerful combination is the Bi-LSTM-CRF which is a bidirectional LSTM RNN with a CRF as the output layer [1, 5]. This hybrid system besides capturing dependencies through the hidden layers also captures label dependencies at the CRF layer. unanse2017 recurrent show that pre-trained word embedding can improve the performance of this hybrid system [9].

2 METHODS

The system consists of a neural network made of Bi-LSTM-CRF layers that identifies the genetic mutations present in the inputs given. We compared two text encoding models: one with character level embeddings and another with word level embeddings.

2.1 Dataset

To train and evaluate our methods, we used the tmVar corpus [11], which consists of 334 documents containing 967 annotations for training and 166 documents with 464 annotations for testing. Fig. 1 shows an example sentence from this corpus.

2.2 Pre-processing

Mutations have a very distinct way of being represented, as happens with most biomedical literature, which leads to being very difficult
A novel missense mutation Asp506Gly in Exon 13 of the F11 gene in an asymptomatic Korean woman with mild factor XI deficiency.

Figure 1: Example sentence from the tmVar corpus (PubMed ID 22016685). The highlighted word corresponds to a mutation mention.

to create hand-crafted rules to detect them. This difficulty also influences the pre-processing necessary to treat the data. The pre-processing done in this work can be divided in two types: one for each model. To the character model was necessary to convert all the corpus into sentences of characters and label each one of them by the BIO labelling system where B is for a character at the beginning of the entity, I for a character that is inside the entity and O for a character hat is the last of the entity.

The word model is more complex because it is necessary to use a tokenizer to convert the corpus into tokens. The tokenizer used was a version of the OSCAR4 tokenizer, changed to work with the tmVar dataset. The labelling system for a word model is the SOBIE system, where the BIO tags are the same as the character model, but S stands for a token that is the whole of an entity (a “singleton”) and the E marks the token responsible for the end of the entity.

2.3 Models
There are a lot of Deep Learning architectures such as CNNs and RNNs, but among the most used for text recognition and classification are Long short-term memory (LSTM) units. RNNs are networks with loops, allowing information to persists. In another words, this networks have “memory”. Yet, they can only retain recent information which can be a serious problem when it is necessary to have past context to predict what is desired. Since LSTMs have a long term memory they can retain information for longer periods and overcome standard RNNs problems.

2.3.1 Character Model. The character model consists of a bidirectional LSTM network with a CRF layer (BI-LSTM-CRF). In fact, there are 3 BI-LSTM layers and one CRF on top of the last one. This model contains two configurations but both have the same architecture as neural network. The character model was trained with a mini-batch approach, in which sentences with similar length are grouped, and with a maximum sequence length approach. Figure 2 shows the character base network diagram.

2.3.2 Word Model. The word model uses token that can be words or part of words which lead us to think that it could have less BI-LSTM layers, but the results showed a big difference in performance when training with 3 BI-LSTM layers like the character model. As said before, the word model needs the use of a tokenizer to pre-process the data and convert the corpus into tokens. After that it is necessary to use embeddings that represent the tokens we have. Tokens that are related are mapped to nearby points which leads the model to understand that they mean the same or that they share a relation. This model was trained using two embeddings models: the BioNPLab word2vec model [7] and the GloVe model [8].

2.4 Word Embedding
Vector Space Models (VSM) represent words in a continuous vector space where semantically similar words are mapped to nearby points (embedded nearby each other). The different approaches that leverage this principle can be divided into two categories: count-based methods and predictive methods. Count-based methods compute the statistics of how often some word co-occurs with its neighbor words in a large text corpus, and then map these count-statistics down to a small, dense vector for each word. Predictive models directly try to predict a word from its neighbors in terms of learned small, dense embedding vectors.

GloVe is a count-based method that focuses on the statistics of word occurrences in a corpus. It learns the vectors by essentially doing dimensionality reduction on the co-occurrence counts matrix. First it constructs a large matrix of (words x context) co-occurrence information. For each word/row it is counted how frequently the word is saw in some context/column in a large corpus.

Word2Vec is a particularly computationally-efficient predictive model for learning word embeddings from raw text. It comes in two flavors, the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model. Algorithmically, these models are similar, except that CBOW predicts target words (e.g. ‘mat’) from source context words (‘the cat sits on the’), while the skip-gram does the inverse and predicts source context-words from the target words. This inversion might seem like an arbitrary choice, but statically it has the effect that CBOW smooths over a lot of the distributional information (by treating an entire context as one observation).

The BioNPLab provides trained embedding models where the vectors were induced from PubMed and PMC texts and created using the word2vec tool. In this case, the model contains more word embeddings related to the biomedical corpora, which is helpful for our dataset that is made of biomedical documents. It also contains around 5M embeddings while the GloVe model used contains around 400K embeddings, but even the biggest GloVe model contains only 2.2M embeddings.

3 RESULTS
Both the character and word model were trained with the Keras framework [2]. As said before the character model was trained with a mini-batch approach, in which sentences with similar length are grouped, and with a maximum sequence length approach. These approaches achieve in a close range, yet with better performance for the mini-batch approach.

The mini-batch approach groups together the sentences that have the same length and feeds them to the network at the same time. In that case, the batch size will vary in accordance to the number of sentences that have the same length. After each epoch being trained, the model is tested for the test dataset and the best epoch is saved as the model to be used. To use the max sequence length approach, it is necessary to pad the sentences to the maximum sequence length. With this approach it is not necessary to group the sentences together and it is possible to use the Keras arguments to define the epochs and the batch sizes. Yet, since our task is to label all the entities of the sentences and most of them are non-mutation entities it was not possible to use Keras callbacks to save the best epoch (the best epoch is found by the test performance) and the last
epoch trained will correspond to the final model. This approach also requires more computational power and more time to train the network since the sentences are bigger because of the padding.

The word model performance is quite low in comparison to the character model. We tested with both word embeddings model here presented, yet the best results were clearly achieved with the BioNPLab model, not only for the number of embeddings (around 5M) but also because the embeddings were induced from biomedical corpus. The only approach presented in this model was a mini-batch approach and the architecture of the model was exactly the same as the character model: three layers of BI-LSTM with a CRF layer on top. The results presented in Table 1 show a clear difference between the character and word models. The mini-batch approach obtained the best results in the character model and was also the best performance. The differences between mini-batch and max-sequence in the character model may be due to the mini-batch approach being able to save the best epoch. The word model requires tokenization to pre-process the data and our tokenizer may not be the more adequate to this task. Other problem may lay in the fact that not all tokens contain a representation in the word embedding models. The BioNPL model contains around 5M tokens, yet in our dataset it leaves 1437 tokens without representation.

The results achieved with the best character-based model are comparable to those achieved with the tmVar tool before applying post-processing rules, as shown in Table 1. Additionally, the tmVar model makes use of six types of features, as described in the paper, and of an annotation scheme composed of 11 labels instead of the more common three (B, I, O). This encoding is based on domain knowledge and allows distinguishing different mutation types at the same type as improving the recognition performance.

### Table 1: Results for both models.

| Model             | Precision | Recall | F-Measure |
|-------------------|-----------|--------|-----------|
| tmVar             | 0.914     | 0.914  | 0.914     |
| w/o post-processing | 0.920     | 0.837  | 0.877     |
| Char-Mini-Batch   | 0.881     | 0.866  | 0.874     |
| Char-Max-Sq       | 0.844     | 0.868  | 0.856     |
| Words-BioNPL      | 0.672     | 0.770  | 0.718     |
| Words-GloVe       | 0.564     | 0.526  | 0.544     |

4 CONCLUSIONS

The objective of this work was to create a NER system that could perform a sequence labelling task in the tmVar dataset. To perform this task, we used two Deep Learning models and compared the performance of both. We created two models, one with character and another with word embeddings. The results obtained show that it is possible to achieve good performance without the presence of hand-crafted rules. The tmVar system produced a F-measure of 0.914 with CRF methods and the help of post-processing methods. In this work, we relied only in a neural network to understand the dataset and achieved as best result 0.874 of F-measure.

Our best result was achieved in the character model and we believe that such results are because of the biomedical corpus being morphologically richer than a standard corpus. The mini-batch approach is the suggested one because it allows to save the best epoch and trains in lower time. The fact that the word model could not find embeddings for all the tokens could be determinant in the lower results presented by the word model, but it is possible to see that the model using the BioNPL embeddings (trained on biomedical corpora) achieved higher results than the GloVe embeddings (trained on standard texts). Another problem for the word model could be the tokenization that is a very important aspect in the creation of any word model. Remains to be seen if another tokenizer could increase the performance of this model.

The proposed model has two important advantages, namely not requiring specially designed features and not requiring any special tokenization, as is the case with tmVar and other probabilistic sequence tagging models. This makes it specially suited to identifying mutations and other entities such as chemicals or proteins for which tokenization is usually an important factor. Applying this model to such entities is a relevant follow-up work, as is the use...
of a richer annotation scheme as used in tmVar. It would also be interesting to study if applying post-processing rules could further improve these results.

ACKNOWLEDGMENTS

This work was partially supported by ERDF - European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme, and by national funds through FCT - Foundation for Science and Technology Portugal, under project ref.: PTDC/EEI-ESS/6815/2014.

REFERENCES

[1] Raghavendra Chalapathy, Ehsan Zare Borzeshi, and Massimo Piccardi. 2016. An Investigation of Recurrent Neural Architectures for Drug Name Recognition. arXiv preprint arXiv:1609.07585 (2016).
[2] François Chollet et al. 2015. Keras. https://keras.io.
[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.
[4] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 1735–1780.
[5] Chen Lyu, Bo Chen, Yafeng Ren, and Donghong Ji. 2017. Long short-term memory RNN for biomedical named entity recognition. BMC bioinformatics 18, 1 (2017), 462.
[6] George Mavromatis. [n. d.]. Biomedical Named Entity Recognition Using Neural Networks. [n. d.].
[7] SPFGH Moen and Tapio Salakoski2 Sophia Ananiadou. 2013. Distributional semantics resources for biomedical text processing. In Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan. 39–43.
[8] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 1532–1543.
[9] Iñigo Jauregi Unanue, Ehsan Zare Borzeshi, and Massimo Piccardi. 2017. Recurrent neural networks with specialized word embeddings for health-domain named-entity recognition. Journal of biomedical informatics 76 (2017), 102–109.
[10] Qinghua Wang, Shabbor S Abdul, Lara Almeida, Sophia Ananiadou, Yalbi I Balderas-Martinez, Riza Batista-Navarro, David Campos, Lucy Chilton, Hui-Jou Chou, Gabriela Contreras, et al. 2016. Overview of the interactive task in BioCreative V. Database 2016 (2016).
[11] Chih-Hsuan Wei, Bethany R Harris, Hung-Yu Kao, and Zhiyong Lu. 2013. tmVar: a text mining approach for extracting sequence variants in biomedical literature. Bioinformatics 29, 11 (2013), 1433–1439.