Disease Named Entity Normalization Using Pairwise Learning To Rank and Deep Learning

Thanh Ngân Nguyễn, Trang Minh Nguyễn, Thanh Hải Đặng*

Faculty of Information Technology,
VNU University of Engineering and Technology,
E3 Building, 144 Xuan Thuy Street, Cau Giay, Hanoi, Vietnam
**hai.dang@vnu.edu.vn*

Abstract

Because chemicals and diseases play an important role in our life, the scientists find out the way of extracting information in biomedical literature to analysis the relation between them. We introduce the system which learns the similarities between mention and concept names by pairwise learning to rank approach and use a special type of neural network - Siamese Network - to combine semantic informations and morphological informations. We tested our model using two benchmark golden corpora, namely the BioCreative V Chemical Disease Relation (BC5 CDR) corpus and the NCBI Disease corpus. We used the disease and chemical vocabularies distributed by the Comparative Toxicogenomics Database project (CTD) as the reference database. Experimental evaluation on the BC5 Disease and NCBI Disease corpus yield the F1 scores of 85.57% and 72.55%, respectively. This result is significantly higher than the baseline model, especially on the BC5 Disease corpus. When comparing with a state-of-the-art model on NCBI corpus, their result exceeds ours due to their adoption of larger vocabulary size and smaller number of test mentions.

Keywords: *named entity normalization, deep learning, convolutional neural networks, pairwise learning to rank.*

1. Introduction

Biomedical information extraction is one of the top interests of biomedical researchers as well as data scientists in recent years since it directly helps improve modern human health, reveal many groundbreaking discoveries, and reduce the cost of research in this field. According to [3], a drug takes around 14 years and two billion dollars on average to be developed. Even though, 95 percent of potential drugs could not pass the development phase. Chemical safety and toxicity studies can be hugely improved if the adverse drug reactions between chemicals and diseases can be recognized. Moreover, recognizing and normalizing biomedical entity are important for discover new, significant relations between chemicals and diseases which do not occur together in the same published article [4].

Named Entity Normalization (NEN) is one of the most important parts of information

extraction, especially for biomedical research and clinical application, which determines the location of specific data and then extracts structured information from unstructured text. In term of biomedical field, Disease Named Entity Normalization is the linking of disease named entities in textual documents to their identifiers that are pre-defined in existing lexicons [5].

The NEN task has a lot of challenges: 1) ambiguity - same entity mention may has several identifiers, 2) variation - multiple naming schemes exist for only one disease identifier [6]. Moreover, disease names are named in many ways, depending on anatomical locations, symptoms, treatment, etc. Besides, disease entity mentions are also frequently long and complex, so they are written as abbreviation. In many studies, people use rule-based algorithm to solve this problem but it is not effective in handling all disease terminology in biomedical literature.

The goal of this research is to build a model for normalization of disease entities mentioned in biomedical textual documents. In this paper, we propose a model using pairwise learning to rank approach and deep learning.

2. Related Work

Biomedical named entity normalization has received a lot of research attention recently. One of high-performing approach is published in the journal Bioinformatics owing to DNorm [5]. DNorm also use pairwise learning to rank approach but they optimize loss function via stochastic gradient descent instead of applying neural network as our method. Their dataset is the NCBI disease corpus with 593 training data, 100 development data and 100 test data. The advantage of DNorm approach is using the set of annotations. For any given mention *m*, model choose some candidates which are relevant or not and decide a suitable entity to work with base on score ranking. The model does not need iterate through all names and the result is an even better [5].

Another state-of-the-art system is CNN-based ranking (2017) [6]. The system consists of two tasks: candidate generation and candidate ranking. The first step uses same based-rule in [7] to generate candidates. It separates the entities into three groups: an entity mention exactly matches an entity in the database, an entity mention exactly matches an entity in the database after changing morphology and an entity mention does not belong to the two categories above, but a part of entity mention appears in database. The second task is ranking candidates based on similarity. To compute the similarities of mention and candidate pairs, they implement a Convolution Neural Network. Their method consists of two main parts: semantic representation and ranking based on similarity. The CNN-based ranking biomedical entity normalization system get the higher result when extracting morphological information.

3. Method

The overview of our models is illustrated in Figure 1. We proposed a NEN pipeline model that consists three parts:

(i) Preprocessing module for stemming and abbreviation resolution.
(ii) Dictionary matching module for speeding up the inference phase.
(iii) Convolutional Neural Networks based model for learning to map text mentions.

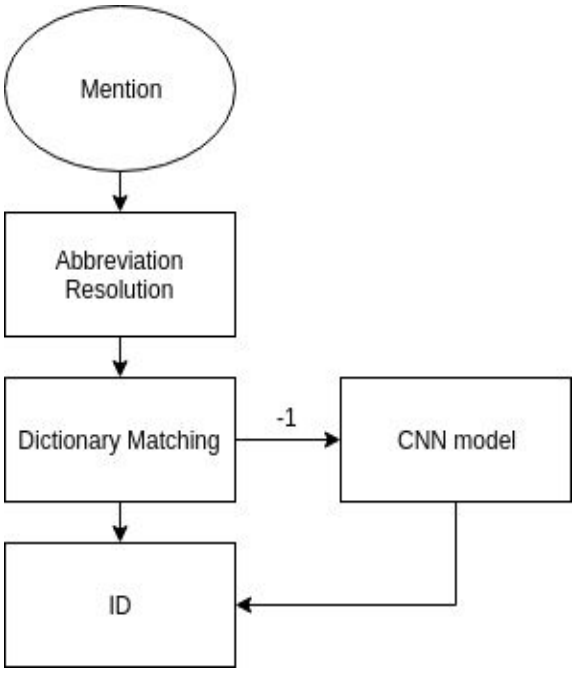

Figure 1. The disease named entity normalization pipeline.

3.1 Abbreviation resolution

In biomedical literature, there are a lot of long disease entities, and they are often referred to using acronyms and other shorthand. Unfortunately, there is no rule to get the full form of their names because in different documents one word can show different meanings and different words can have the same meaning. Abbreviation definition identification based on automatic precision estimates [8]. For example, “AD” is the shorthand of both “Alzheimer's Disease” and “Attachment Disorder”.

For disease entities, we use Abbreviation Plus Pseudo-Precision (Ab3P) tool[1] which is an abbreviation definition detector. Ab3p can be used to identify abbreviations in documents and return the list of replacement words with probability. For instance, if entity “PFS” appeared in document, Ab3P would detect it and return the result “PFS|progression-free survival|0.999408” in which “PFS” is the abbreviation name, “progression-free survival” is the full name, “0.999408” is the probability of an abbreviation name with a full name.

3.2 Dictionary matching

To make it more convenient in accessing any disease named entity identifiers, we create a dictionary with keys are names after preprocessing and values are corresponding mention identifiers. In preprocessing step, We convert names to lowercase, remove punctual characters and stem for each word to reach higher accuracy. With stemming, we use Snowball tool[2] and remove all white spaces among tokens. However, dictionary matching cannot normalize several mentions since it has no context information. Even though dictionary matching method has some disadvantages, it gives a high precision result because the concept assigned to a mention by dictionary matching is very likely to be the right concept. If there is no such concept to be found, dictionary matching will not predict.

[1] https://github.com/ncbi-nlp/Ab3P
[2]http://www.nltk.org/_modules/nltk/stem/snowball.html

3.3 Convolution Neural Network

Candidate Generation

The candidate generation prepares the input for the task of ranking pairs mentions and concept names.

For each mention *m* in the vocabulary, we create:

- A set of relevant concept names (*mention(n^+)*) which are names that have the same identity as *m*.
- A set of irrelevant concept names (*mention(n^-)*) which are names that do not have the same identity as *m* but morphologically look like *m*.

Each training examples include *mention m*, *positive name* n^+, *negative name* n^-, with $n^+ \in mention(n^+)$, $n^- \in mention(n^-)$. A name is represented as two lists: a word list and a character list which brings semantic information and morphological information.

Pairwise Learning To Rank

Learning to rank for information retrieval is a task to automatically construct a ranking model using training data, such that the model can sort new objects according to their degrees of relevance, preference, or importance [5]. There are three common approaches to learn to rank algorithms: pointwise, pairwise and listwise. However, in our research, we use pairwise learning to rank approach. The pairwise approach does not focus on accurately predicting the relevance degree of each document; instead, it cares about the relative order between two documents [5].

Let $M=\{m_1, m_2, ..., m_n\}$ is a set of mentions from the corpus where *n* is the number of mentions, *mention(n^+)* is set of relevant concepts names, *mention(n^-)* is set of irrelevant concepts names. With each $m_i \in M$, we find a pair of names (n^+, n^-) where n^+ is a positive name, n^- is a negative name, $n^+ \in mention(n^+)$, $n^- \in mention(n^-)$. We measure the distance between mention m_i with name based on score function, $score(m_i, n^+)$ and $score(m_i, n^-)$, score are computed as any distance function, e.g Euclidean. Suppose that the higher score is, the more relevant the pair (*mention, name*) are. Therefore, the loss function aims to maximize $score(m_i, n^+)$ and minimize $score(m_i, n^-)$.

Pairwise learning to rank is based on similarity between mention and concept names stored in lexicons. Mentions and concept names are represented as embeddings:

(i) For word embeddings, a word list is used to look up the corresponding vectors pre-trained by a word embeddings model.

(ii) For character embedding, we initialize uniformly distributed arrays and train them as parameters of model.

Siamese Network

Word embeddings bring the semantic information and character embeddings learn morphological information. Therefore, we develop two Siamese Networks which take word embeddings and character embeddings as inputs and then we combine the two output vectors into one vector that encompasses semantic and morphological information. The last layer implements the contrastive loss as the score function [10]. The exact loss function is:

$$= \tfrac{1}{2}(1-Y)(D_W(m_{embedding} - n^{n-}_{embedding}))^2$$

$$+ \tfrac{1}{2}Y(max(0,\ \varepsilon + D_W(m_{embedding} - n^{n+}_{embedding})))^2$$

In which, Y is 1 if n is *positive name* $n+$, Y is 0 if n is *negative name* $n-$; D_w is any distance function (i.e. Euclidean); ε is the margin; m *embedding*, n^+ *embedding*, n^- *embedding* is embedding of mention, positive name, negative name, respectively.

Figure 2 below shows the main architecture of the model.

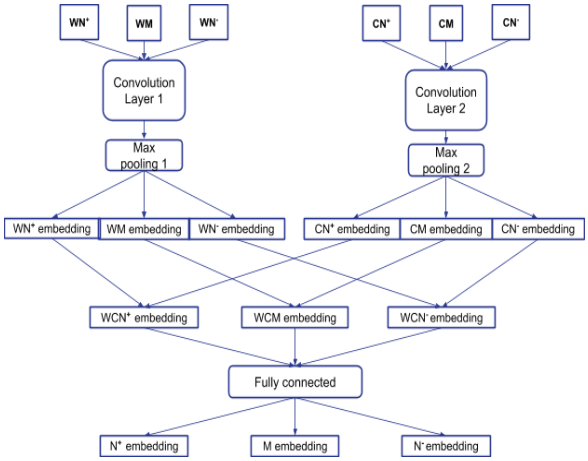

Figure 2. Using siamese networks to combine word embeddings and character embeddings.

4. Experiments and Results

4.1 Datasets

For the named entity normalization, we used the BioCreative V Chemical Disease Relation (BC5 CDR) corpus, the NCBI Disease corpus, and the Comparative Toxicogenomics Database project (CTD)[3] (see Table 1).

[3] http://ctdbase.org

Table 1. Information about the corpora used for training and evaluating the model.

Corpus	Subset	Articles	Mentions	Unique
BC5 CDR	Train	500	4182	1965
	Dev	500	4244	1865
	Test	500	4424	1988
NCBI	Train	593	5145	1710
	Dev	100	787	368
	Test	100	960	427

In BC5 CDR corpus, we found that the average disease mentions length is 1.62 with the longest term of 15 tokens ("colorectal, breast, pancreaticobiliary, gastric, renal cell and head and neck cancers"). Besides, in the NCBI Disease corpus, the average disease mention length is 2.051 tokens, with 13 tokens for the longest ones ("rare, sex-linked recessive, dysmyelinating disease of the central nervous system").

4.2 Experiments and results

Baseline model

The baseline model we use in this report is a traditional dictionary matching method. The algorithm of this method is simple. Given a database, which is, in this case, all concept names in the CTD vocabulary plus the training data of a corpus, and a list of all entity mentions, the baseline model will use the mentions' text to look up in the database and return corresponding identifier of those mentions.

Model settings

We implement the CNN model using the TensorFlow library[4]. Table 2 summarises

[4] TensorFlow is an Open Source Software Library for Machine Intelligence: https://www.tensorflow.org

the hyper-parameters tuned on the development set of BC5 Disease corpus.

Table 2. Hyper-parameter settings of Convolutional Neural Network model.

Hyper-parameter		Value
Embedding dimension	Word embedding	200
	Character embedding	25
Convolutional layer 1	Number of filters	64
	Number of units in fully connected layer	256
	Filter size	(1, 200)
Convolutional layer 2	Number of filters	32
	Number of units in fully connected layer	128
	Filter size	(4, 25)
Max pooling	Size	(3, 3)
Fully connected layer	Number of units	384
Dropout	Rate	0.5
Mini-batch	Size	64

Because the word embedding is based on frequencies of co-occurring adjacent tokens, the pre-trained word embeddings trained on a large-scale data should be more effective than that on a small-scale one. Our word embeddings was the 200-dimensional vector as provided by [11] which used the word2vec skip-gram implementation [12] on all PubMed abstracts and PMC full texts (6 million distinct words). The character embeddings are initialized randomly using Glorot uniform initialization algorithm [13] but instead of fixing the embedding matrix, the loss which comes from supervised training process is also used to make an update to character embeddings. In another word, character embeddings are also adequately tuned during model's optimization by back-propagating gradients. We also apply early stopping [14] and dropout [15] to reduce the effect of overfitting. Early stopping is based on performance on development set which is randomly sampled from the training data using the rate of 30 percent.

Evaluation metrics

We use the same evaluation metrics as DNorm [5] in this report. The exact location and the number of occurrences of each mention are ignored, only the set of disease concepts found within each abstract is used to evaluate. Given the set of identifiers annotated in the golden test set of a corpus and the set of identifiers returned by the model, the number of true positives is the intersection of these two sets. We calculate the standard Precision, Recall and harmonic F1 score and average them using micro-averaged.

Experimental results

After being trained on the training dataset, the model is tested using the testset of each corpus and produced the results reported in Table 3 and Table 4. On the BC5 Disease and NCBI corpus, our model could yield the F1 scores of 85.57% and 72.55%, respectively. As a comparison, the model of Cho et al. (2017) [18] has F1 of 78.8% on the NCBI corpus [2]. We noted that their performance score, however, is reported from 843 test mentions while ours from 960. Further, they adopted a vocabulary larger than that used in ours.

Table 3. Evaluation results of the baseline and our proposed model on two different test sets.

Model	**Dataset**	**P**	**R**	**F1**
Baseline	BC5 Disease testset	82.23	80.78	81.50
	NCBI testset	**74.43**	68.05	71.10
Proposed model	BC5 Disease testset	**83.83**	**87.37**	**85.57**
	NCBI testset	68.88	**76.63**	**72.55**
Cho et al., 2017	NCBI testset	**70.60**	**89.10**	**78.80**

Table 4. Dictionary matching results on the test set

	Match right	Match wrong	Not match	Matching accuracy
BC5 CDR testset	3535	194	701	94.79
NCBI testset	653	93	231	87.53

4.3 Discussion

Each part of the proposed pipeline plays a specific role in the overall performance of the entire NEN process. This pipeline shows a potential of being a standing alone NEN tool as well as a compatible NEN module for a NER system. Although our model outperforms the baseline model, it still has a lot of potential improvements which could be done in order to become the new state-of-the-art model in this field of study.

Currently, the model is able to correctly assign IDs for mentions that:

- Match exactly to one of the concept names.
- Slightly differ from one of the concept names such as being plural forms or having short prefix.

However, a mention might be completely different from all of the concept names that share its ID. In that case, the model finds it hard to map the mention to the concept since the embedding of that mention would not be close to any of the concept names' embeddings.

5. Conclusions and future work

5.1 Conclusions

In conclusion, based on [5, 6], we developed a named entity normalization pipeline using pairwise learning to rank and deep learning which are flexible and compatible with the upstream NER module and any downstream process. We tested our model using two well-known golden corpora namely the BC5 CDR corpus and the NCBI Disease corpus demonstrated the model efficacy. Each part of model (Abbreviation Resolution, Dictionary Matching, CNN based model) has been proved to contribute at improving the model performance and the model are compared with others approaches and show comparable results.

5.2 Future Work

Since there are limitations of the model as discussed in the 4.3. Discussion section, the focus of the future works will be solving the overfitting problem and finding a better way to generate training candidates which are data being fed into the CNN model. After dealing with those challenges, we will extend our work to normalize other entity types such as chemical, gene, protein, etc. To improve the overall performance, we will develop a joint model of named entity recognition and named entity normalization as it being suggested to be more effective than the pipeline approach [16, 17].

Acknowledgments

We would like to sincerely thank Dr. Nguyễn Bá Đạt (Faculty of Information Technology, VNU University of Engineering and Technology, Hà Nội) for the useful, constructive comments and the approval for publication of our work as a technical report.

References

[1] Li, Jiao, Yueping Sun, R. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu, "Annotating chemicals, diseases, and their interactions in biomedical literature", *Proceedings of the fifth BioCreative challenge evaluation workshop*, 2015, pp. 173-182.

[2] Doğan, Rezarta Islamaj, Robert Leaman, and Zhiyong Lu, "NCBI disease corpus: a resource for disease name recognition and concept normalization", *Journal of biomedical informatics*, 2014, vol. 47, pp. 1-10.

[3] Wei, Chih-Hsuan, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Jiao Li, Thomas C. Wiegers, and Zhiyong Lu, "Overview of the BioCreative V chemical disease relation (CDR) task", *Proceedings of the fifth BioCreative challenge evaluation workshop*, 2015, pp. 154-166, Spain: Sevilla.

[4] Chun, Hong-Woo, Yoshimasa Tsuruoka, Jin-Dong Kim, Rie Shiba, Naoki Nagata, Teruyoshi Hishiki, and Jun'ichi Tsujii, "Extraction of gene-disease relations from Medline using domain dictionaries and machine learning", *Biocomputing 2006*, 2006, pp. 4-15.

[5] Leaman, Robert, Rezarta Islamaj Doğan, and Zhiyong Lu, "DNorm: disease name normalization with pairwise learning to rank", *Bioinformatics 29*, 2013, no. 22, pp. 2909-2917.

[6] Li, Haodi, Qingcai Chen, Buzhou Tang, Xiaolong Wang, Hua Xu, Baohua Wang, and Dong Huang, "CNN-based ranking for biomedical entity normalization", *BMC bioinformatics*, 2017, no. 11, vol. 18, pp. 385.

[7] D'Souza, Jennifer, and Vincent Ng, "Sieve-based entity linking for the biomedical domain", *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015, vol. 2, pp. 297-302.

[8] Sohn, Sunghwan, Donald C. Comeau, Won Kim, and W. John Wilbur, "Abbreviation definition identification based on automatic precision estimates", *BMC bioinformatics*, 2008, no. 1, vol. 9, pp. 402.

[9] Liu, Tie-Yan, "Learning to rank for information retrieval", *Foundations and Trends® in Information Retrieval*, 2009, no. 3, vol. 3, pp. 225-331.

[10] Hadsell, Raia, Sumit Chopra, and Yann LeCun, "Dimensionality reduction by learning an invariant mapping", *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, 2006, vol. 2, pp. 1735-1742, IEEE.

[11] Pyysalo, S., F. Ginter, and H. Moen, "Distributional semantics resources for biomedical text processing", *LBM 2013*, pp. 39-44.

[12] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean, "Distributed representations of words and phrases and their compositionality", *Advances in neural information processing systems*, 2013, pp. 3111-3119.

[13] Glorot, Xavier, and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks", *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249-256.

[14] Caruana, Rich, Steve Lawrence, and C. Lee Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping", *Advances in neural information processing systems*, 2001, pp. 402-408.

[15] Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *The Journal of Machine Learning Research*, 2014, no. 1, vol. 15, pp. 1929-1958.

[16] Le, Hoang-Quynh, Mai-Vu Tran, Thanh Hai Dang, and Nigel Collier, "The UET-CAM system in the BioCreAtIvE V CDR task", *Fifth BioCreative challenge evaluation workshop*, 2015, pp. 208-213.

[17] Leaman, Robert, and Zhiyong Lu, "TaggerOne: joint named entity recognition and normalization with semi-Markov Models", *Bioinformatics 32*, 2016, no. 18, pp. 2839-2846.

[18] Hyejin Cho, Wonjun Choi and Hyunju Le, A method for named entity normalization in biomedical articles: application to diseases and plants. *BMC Bioinformatics*, 2017.