Capstone Project
BioCreative VII Track 2

Chemical Named Entity Recognition in PubMed Articles

Wenmo Sun
School of Information, The University of Arizona, Tucson, AZ 85721, USA

Fall 2021

Table of Content

Introduction

One of the most challenging applications of NER is considered as biomedical-related NER which aims to identify and classify entities, such as chemical names, proteins, diseases, etc. In the meantime, the complexity of the problems and practical applications make the research in this field very meaningful for the public good. The study of biomedical NER can be dated back to the early 2000s with the surge of NER research[1]. During the past two decades, the approaches to solve biomedical NER problems have been evolving with the development of technologies. This project is part of the BioCreative VII Track 2 – Full-text Chemical Identification and Indexing in PubMed Articles. During this semester, I explored the data augmentation approach to improve the performance of BioBERT on this task. The results showed that substitution of chemical entities did not help improve the performance. This finding agreed with a recent paper that attempted to solve the same problem.

Background

Named Entity Recognition (NER) is a Natural Language Processing (NLP) task that involves processing and identifying interested words or expressions in a text.[2] NER has shown very diverse applications, which include information extraction/retrieval, question-answering, machine translation, text summarization, information mining, etc.[3] The application of automatic NER systems can drastically reduce the costly manual extraction.

The automatic identification of chemical names has been very challenging due to various reasons. Firstly, it requires domain knowledge; Secondly, the expression of chemical compounds (such as drugs names, generic names, empirical names, abbreviations, etc.) can hugely vary in different settings. The exact same compound may have multiple different names and can be written in different forms by different nomenclature. Thirdly, chemical names can often be found as unstructured data which can be images or scans of handwriting. Last but not the least, chemical NER shares some common challenges as other NER systems, such as language factors, ambiguity in the text, annotation of training data, etc.[3]

Regardless of the issues and challenges, chemical NER has advanced significantly over the past decades. Many different approaches were used to improve existing approaches and methods. They are generally be categorized as being based on dictionaries, rules, machine learning, and even deep learning.[4][5] With the surge of deep learning techniques[6], there have been many new approaches added to the field. This paper intends to survey the current and past progress of NER systems designed for chemical identification tasks. It is organized as follows. In Findings and Discussion, Traditional Approaches is a section that introduces the past efforts in developing chemical NER systems. Deep Learning Approaches covers the recent achievement in chemical NER with the use of various deep learning techniques. Hopefully, this paper can fill the gap as the existing reviews in this field have not touched on deep learning techniques.

A typical chemical NER system is built through data pre-processing, features processing, applying NER techniques, and solving recognition mistakes or normalization. As for the NER techniques, they include dictionary-based approaches, rule-based approaches, and machine learning approaches, there are deep learning approaches.

To fully understand the performance of NER tasks, it is important to know a few keywords and terms. The most common metrics of NER tasks are precision, recall, and F-1 score. They are widely used and accepted to evaluate NER systems. Precision is the fraction of relevant instances among the retrieved instances. Recall is the fraction of relevant instances that were retrieved. In addition to precision and recall, F-1 score is another import metric seen almost in every NER task. F-1 score is the harmonic mean of precision and recall. It is widely used in machine learning and natural language processing for measuring performance. It is considered more useful since it takes both false positives and false negatives into account, especially when there is uneven class distribution.

$$Precision = \frac{true\ positive}{true\ positive + false\ positive}$$

$$Recall = \frac{true\ positive}{true\ positive + false\ negative}$$

$$F1 = \frac{2 \times\ Precision\ \times\ Recall}{Precision + Recall}$$

Traditional Approaches

Dictionary-based approaches were straightforward to understand. It requires a dictionary of chemical names and therefore allows the identification of chemical names in free texts. While it may achieve very high precisions, but the drawback, low recall, can be very bothersome. Therefore, it is very difficult for dictionary-based systems to achieve high F1 scores. [7][8] Dictionary-based methods are considered simple and precise when the exact entries are provided. But the drawback is low recall. It is believed that dictionary-based approaches cannot handle unseen entities and ambiguous contexts effectively.[9] Meanwhile, the ongoing maintenance of the dictionaries is very costly and time-consuming. [10][11]

Another approach in this category is based on hand-crafted rules to extract name entities in the text. It is known that rule-based systems are often genre and task-dependent. The rules are constructed based on patterns of the words or the context of the words. Therefore, they are lacking portability and are hard to adapt to new types of entities. [12] In consequence, rule-based methods are usually used together with dictionaries to improve precision and recall.

Machine Learning Approaches

Several Machine Learning algorithms have been used for chemical NER development. Conditional Random Fields (CRFs) were seen in many chemical NER tasks.[13]–[16] The basic idea of CRFs is that the label of current words depends only on the tag of one previous word. Much research used to center on the Hidden Markov Model (HMM) which is a generative type of sequence-based model. They are considered simple and quick.[17]–[19] However, HMM models cannot identify the relationships between neighboring tokens. Maximum Entropy Markov Model (MEMM) was proposed to address the issues of HMM using a conditional model.[8], [20] It is believed that CRFs have several advantages over HMM and MEMM models. CRFs use undirected graphs to avoid label bias problems that come with MEMM and other Conditional Markov Models. Therefore, CRFs have become more popular than the other two approaches. In addition to the above approaches, Support Vector Machines (SVMs), Naïve Bayes, Decision Tree, and other ML-based approaches were also seen in chemical NER tasks. They were reported to achieve comparable results in some of the tasks.[21]

There have been a handful of reviews on the traditional ML-based models. The traditional approaches often can be seen to have F-1 scores from 50% -70% which are good but not ideal. To get more details, readers can refer to the cited references and explore more if interested. Since this paper will focus more on the current advance, the following section covers deep learning-based approaches.

Hybrid Models

Since each of the above approaches has its own advantages and weaknesses, it is often seen that multiple approaches were used together to take advantage of each approach. For instance, CRFs and dictionary-based approaches were used in ChemSpot and achieved a 68.1% F-1 score in their task. More recently, a BiLSTM-CRF model is reported to be helpful in solving Biomedical NER problems and achieved a 73.50% F-1 score in their task.[22] Some of the hybrid models can even achieve over 90% F-1 scores on chemical NER tasks.[23], [24] This is not surprising as deep learning approaches overcame the shortcomings of the early methods and showed great success in solving NLP problems.

Deep Learning Approaches

Deep Learning (DL) is a subfield of Machine Learning, and it is inspired by the structure and function of human brains which is called artificial neural networks. With the help of DL, chemical NER systems have been shown to outperform previous NER models.[25]–[27] Because of the benefits of DL systems, they have been predominantly used in chemical NER research in recent days.

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) that can handle order dependence in sequence prediction problems. It has been used in chemical NER tasks. In the CollaboNet model reported in 2019, the F1-score of the chemical NER task was achieved at 92.72%.[28] This is an improvement of a few previous LSTM-based chemical NER models. When being evaluated on the same dataset, BC5CDR-chem, three previous LSTM-based models were at 89%, 90%, and 93.14%.[8], [27], [29] In 2020, there was a hybrid model, DTranNER, which achieved a 93.44% F-1 score on the chemical NER task. They used two separate deep learning networks which provided distinctive contextual clues to enhance their model.[30]

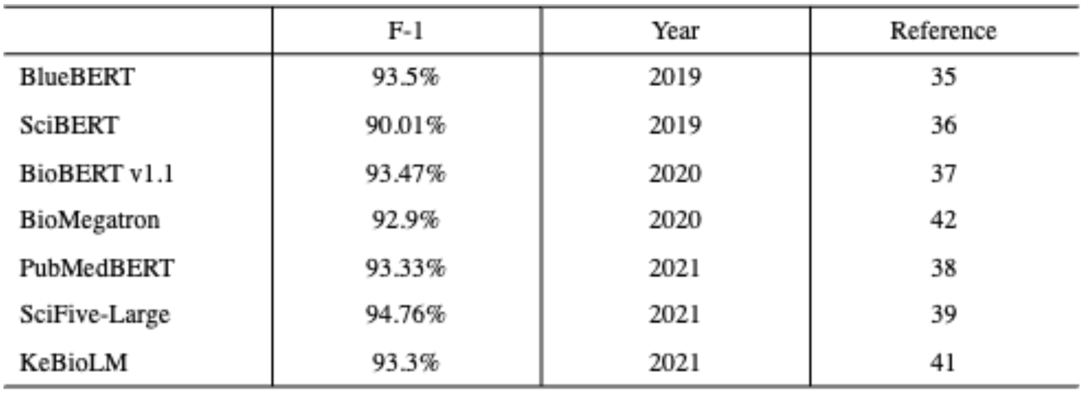
In addition to RNN, a convolutional neural network (CNN) was also helpful in solving chemical NER tasks. CNN was first used in combination with LSTM models for chemical NER in 2018. They found that CNN has the advantage of reducing training complexity as compared to only using LSTM models.[31] There is also other research on implementing CNN in chemical NER tasks.[32]–[34]

Another big name in the DL approach family is BERT. BERT is the Bidirectional Encoder Representations from Transformers created in 2018.[6] In contrast to the models that read the text in sequential orders, BERT read the text at once and this character allows the model to learn the context of a word based on all of its surroundings. In that sense, the BERT-based model should be very suitable for chemical NER tasks.

The first application of BERT in chemical NER tasks was in 2019. The model is known as Biomedical Language Understanding Evaluation (Blue) BERT. The BERT model was pretrained on PubMed abstracts and clinical notes and fine-tuned with ELMo. The F-1 score it achieve on BC5CDR-chem task was 93.5%.[35] Another BERT-based model, SciBERT, was released around the same time. The model used unsupervised pretraining on a large multi-domain corpus of scientific publications to improve performance on downstream tasks. The model was evaluated on BC5CDR dataset as well. The best F-1 score for NER task was 90.01%. [36] The above two models were also evaluated against BioBERT model. In 2020, BioBERT v1.1 was released with a 0.62% F-1 score improvement in chemical NER tasks. The BERT-based model was trained on PubMed abstracts, PMC full-text articles, and BooksCorpus. The additional corpora of different sizes were believed to improve the performance.[37] Researchers observed that mixed-domain pretraining didn't help to improve the performance. To solve problems in chemical NER, domain-specific pretraining is the key. Therefore, the model can benefit from in-domain vocabularies and language models generated from in-domain data. When being evaluated on BC5CDR-chem data, PubMedBERT achieved an F-1 score of 93.33%.[38]
A more recent text-to-text transfer transformer (T5) model, SciFive-Large, is the best-reported system on chemical NER task (BC5CDR). It outperformed BioBERT and other BERT-based models. The F-1 score on BC5CDR reached 94.76% which is the highest score reported. [39] T5 approach overcomes the limitation of BERT which only produces a single prediction for a given input by outputting a string of text for each input.[40]

In addition to the above BERT-based models, there are many more models that have done a great job on chemical NER tasks. For instance, KeBioLM model is a PubMedBERT based model trained with knowledge-enhanced language models. The F-1 score of the chemical NER task evaluated on BC5CDR is reported to be 93.3% which is the same as PubMedBERT.[41] BioMegatron model also took advantage of domain language models and it achieve an F-1 score of 92.9% for chemical NER task (BC5CDR).[42]

Table 1. Comparison of BERT/Transformer-based chemical NER models on BC5CDR dataset.



	F-1	Year	Reference
BlueBERT	93.5%	2019	35
SciBERT	90.01%	2019	36
BioBERT v1.1	93.47%	2020	37
BioMegatron	92.9%	2020	42
PubMedBERT	93.33%	2021	38
SciFive-Large	94.76%	2021	39
KeBioLM	93.3%	2021	41

BC5CDR which has been mentioned multiple times in the above paragraphs is a corpus used in BioCreative Challenge V. BioCreative is a challenge started in 2004 which allows the researcher to share their knowledge and skills in text mining and information extraction systems applied to the biological domain.{Citation} There has been a great achievement in the past as evidenced by the popularity of their datasets. This project is part of the BioCreative VII, and the corpus used for the task is BC7T2-NLMChem-corpus_v2.BioC. It is a corpus consisting of 150 full-text PubMed articles with chemical entity annotations from human experts. There are around 5000 unique chemical names, mapped to around 2000 MeSH identifiers.

Medical Subject Headings (MeSH) thesaurus is a controlled and hierarchically organized vocabulary produced by the National Library of Medicine. It is used for many biomedical NLP tasks, and it contains chemical names. The MeSH data used for this project is 2021 MeSH.

Procedure and Findings

Preparation

At the beginning of the semester, I worked on setting up HPC so that I can run the NER baseline model utilizing the computing resources. To set up the HPC properly, I built a singularity image file from remote. The recipe file I used for the singularity image file was obtained from https://github.com/clulab/hpc-ml/. The base BioBERT was downloaded from https://github.com/dmis-lab/biobert. The BioBERT v1.1 model was obtained from https://huggingface.co/dmis-lab/biobert-v1.1/tree/main. The datasets for this task were downloaded from BioCreative at https://ftp.ncbi.nlm.nih.gov/pub/lu/BC7-NLM-Chem-track/.

Experiment and Discussion

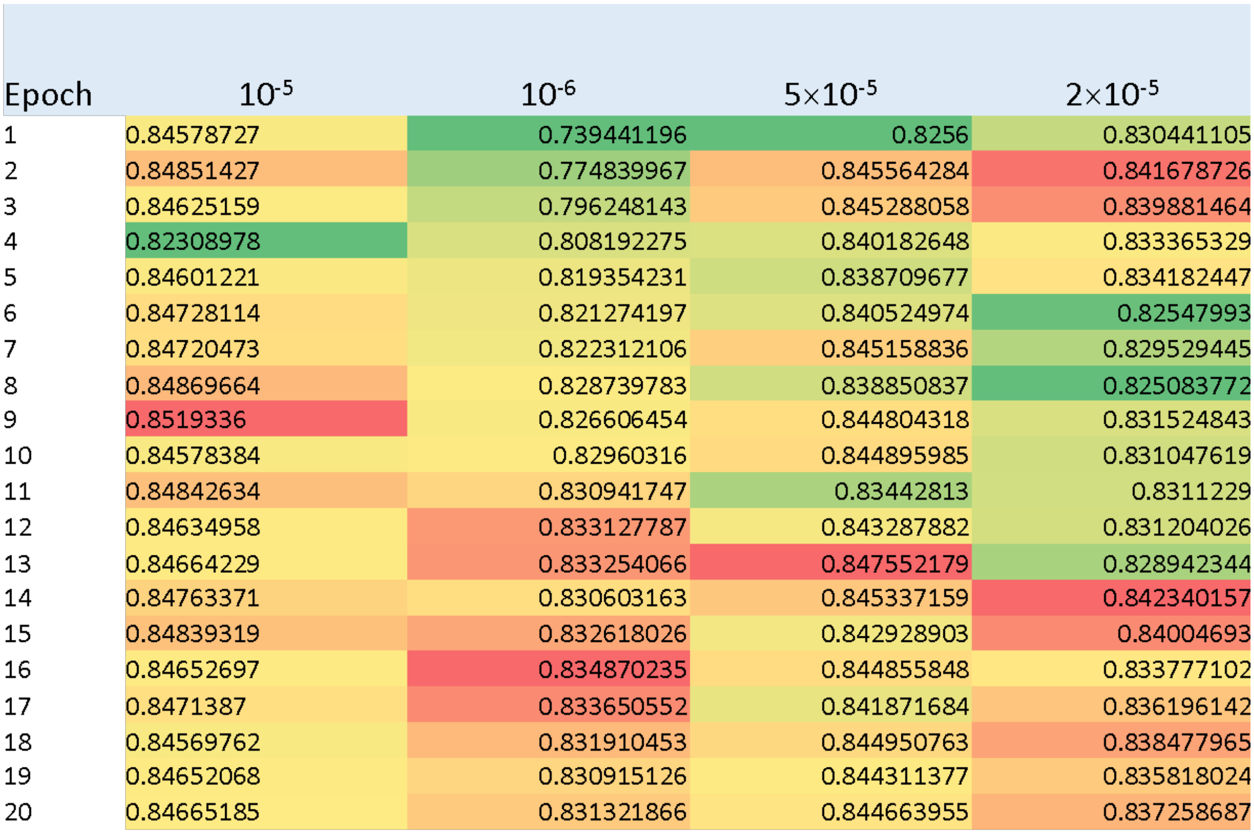
- Learning rate determination

After getting familiar with the process of submitting tasks to HPC, I started to run the baseline model, BioBERT (v1.1), using the provided training data (converted from BC7T2-NLMChem-corpus_v2.BioC.xml). An example slurm file used for running the task is shown below. As it shows, the task was submitted to the puma cluster using the windfall queue. It requested 4 GB memory and 1 GPU. The estimated running time was 4 hours. The container used was 0927-1.sif. The base model used was BioBERT v1.1.

```
#!/bin/bash

#SBATCH --job-name=BioBERT-0927-1-puma
#SBATCH --mail-user=wmsun@email.arizona.edu
#SBATCH --partition=windfall
#SBATCH --ntasks=1
#SBATCH --nodes=1
#SBATCH --mem=4gb
#SBATCH --time=04:00:00
#SBATCH --gres=gpu:1


cd ~/BioBERT
export HF_HOME=cache
export CUDA_VISIBLE_DEVICES=0

 singularity exec --nv 0927-1.sif python3.7 -u run_ner.py \
 --model_name_or_path dmis-lab/biobert-v1.1 \
 --task_name ner \
 --train_file BC7T2-NLMChem-corpus_v2.BioC.train.json \
 --validation_file BC7T2-NLMChem-corpus_v2.BioC.dev.json \
 --output_dir model/model-0927-1-puma/ \
 --num_train_epochs=20.0 \
 --save_steps=10000 \
 --save_total_limit=3 \
 --evaluation_strategy=epoch \
 --learning_rate=1e-06 \
 --per_device_train_batch_size=8 \
 --do_train \
 --do_eval
```

The default learning rate of the program was set to 10^{-5} as displayed in the above slurm file. To study if different learning rates would produce better results, learning rates at 10^{-6}, $5\cdot10^{-5}$, $2\cdot10^{-5}$ were also used. The corresponding results are listed in Table 2. It was clear that the learning rate at 10^{-5} performed the best. Therefore, for the following tasks, the learning rate was kept at 10^{-5}.

Table 2. Summary of model performance at different learning rates.



Epoch	10^{-5}	10^{-6}	5×10^{-5}	2×10^{-5}
1	0.84578727	0.739441196	0.8256	0.830441105
2	0.84851427	0.774839967	0.845564284	0.841678726
3	0.84625159	0.796248143	0.845288058	0.839881464
4	0.82308978	0.808192275	0.840182648	0.833365329
5	0.84601221	0.819354231	0.838709677	0.834182447
6	0.84728114	0.821274197	0.840524974	0.82547993
7	0.84720473	0.822312106	0.845158836	0.829529445
8	0.84869664	0.828739783	0.838850837	0.825083772
9	0.8519336	0.826606454	0.844804318	0.831524843
10	0.84578384	0.82960316	0.844895985	0.831047619
11	0.84842634	0.830941747	0.83442813	0.8311229
12	0.84634958	0.833127787	0.843287882	0.831204026
13	0.84664229	0.833254066	0.847552179	0.828942344
14	0.84763371	0.830603163	0.845337159	0.842340157
15	0.84839319	0.832618026	0.842928903	0.84004693
16	0.84652697	0.834870235	0.844855848	0.833777102
17	0.8471387	0.833650552	0.841871684	0.836196142
18	0.84569762	0.831910453	0.844950763	0.838477965
19	0.84652068	0.830915126	0.844311377	0.835818024
20	0.84665185	0.831321866	0.844663955	0.837258687

- Error analysis

The next step was to perform an error analysis of the existing model produced by training with the training data. There were 1900 error cases out of 11183 test cases. The major issue found was the model had difficulties identifying entities that are in abbreviation forms. For instance, an entity ‘SYBR green’ is a chemical entity but the model failed to identify it. Based on this observation and discussion with the team, I determined to perform data augmentation for the next step.

Prior to data augmentation, I also trained the model again with 61089 chemical entities that have abbreviations (4 or more uppercase letters) and 17872 chemical entities that have abbreviations (2 or more uppercase letters). The F-1 scores dropped to 0.1 and 0.03 respectively. This has shown that training the model with standalone chemical entities was not helpful at all. Therefore, I proceeded to do data augmentation.

- Data augmentation

The plan for data augmentation was to replace chemical entities in the training data with randomly picked chemical names with abbreviations (from MeSH). Then, the new training data can be used to train the model and the performance of the model was expected to improve as this step purposefully teach the model to learn about abbreviations in the PubMed article contexts. The code used to do data augmentation was written in Python and can be found at https://github.com/clulab/chemnorm. The most challenging part of the process was to correctly update the spans of the tokens. There were multiple different situations when it comes to updating spans. For instance, when the length of the tokens is different, after inserting the new spans of the current token, all the subsequent token spans should be updated. Further, when there are spaces or symbols such as dash lines, the spans will be different depending on various conditions.

The original training data was converted into python dictionaries for easier processing. They are ‘ner_tags’, ‘tokens’, and ‘spans’. In the ‘ner_tag’, there are IOB labels assigned by the human annotators. Inside ‘tokens’, there are tokens. For each of the tokens in ‘tokens’, there are corresponding spans in ‘spans’.

The data augmentation of the original training record was implemented in separate steps as there are several keys in each of the records. Therefore, to replace the chemical entities in the records, I first randomly picked chemical entities from MeSH. Then, the new chemical entities replace the original ones in the ‘tokens’. Since there are new tokens introduced, the spans and tags need to be updated. To update the tags, the replacement needs to be tokenized and assigned IOB tags. Then, the old tags were removed, and the new tags were inserted. Similarly, to update the tokens, old tokens were removed, and new tokens were added. These two steps were straightforward. However, when it came to updating the spans, there were several trivial factors need to be considered. Several challenges include updating one span will require the update of all the subsequent spans; spaces between tokens and symbols need to be considered because spaces can be absent or present in between them; the size of the spans list will change, etc. After rounds of adjustment and debugging, I was able to realize the wanted function which is to substitute the old chemical entities with new ones. To be confident to use the newly generated data, a roundtripping test was needed.

The program I wrote for data augmentation performed well in roundtripping tests, given the complexity of the task. As shown in Table 3, over 90% of the records passed the test and I continue to use the 8460 records for producing additional training data. After completing the data augmentation, I performed training with three training datasets (4230, 8640, 16920 records).

Table 3. Results of roundtripping tests.

Records	Contain chemical entities	Matched	Not matched
0~7059	3184	2924	260
7061~18910	6213	5536	677
	9397	8460	937

- Training with new data

The results of the additional training showed that the augmented data caused the performance of the model to drop (Figure 1). I was not sure if the document id in the training data would affect the performance. In the data augmentation process, I chose to update the document id by 1 for each record even though they might come from the same document. If the document id plays a role in the training process, inconsistency in the document id will affect the performance of the model. Therefore, I prepared another dataset and made sure the sentences from the same article share the same document id. The dataset was used to train the model and it (red triangle in Figure 1) did not show a significant difference from the previous training.

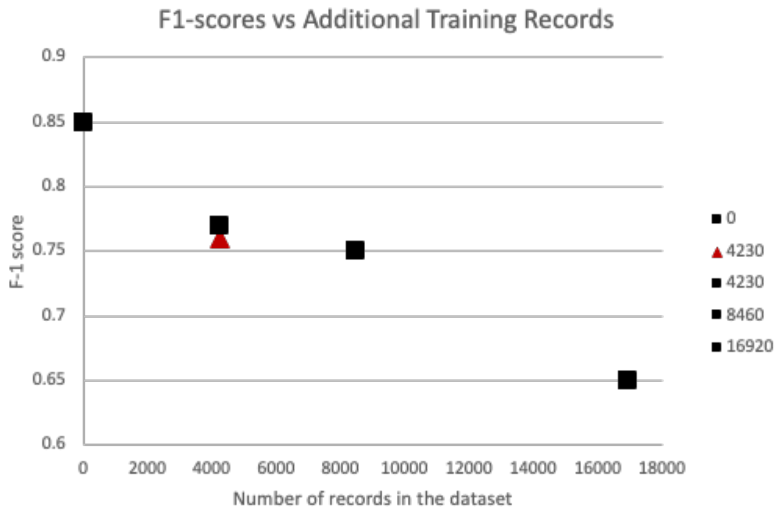


Figure 1. Performance of the model using additional training records.

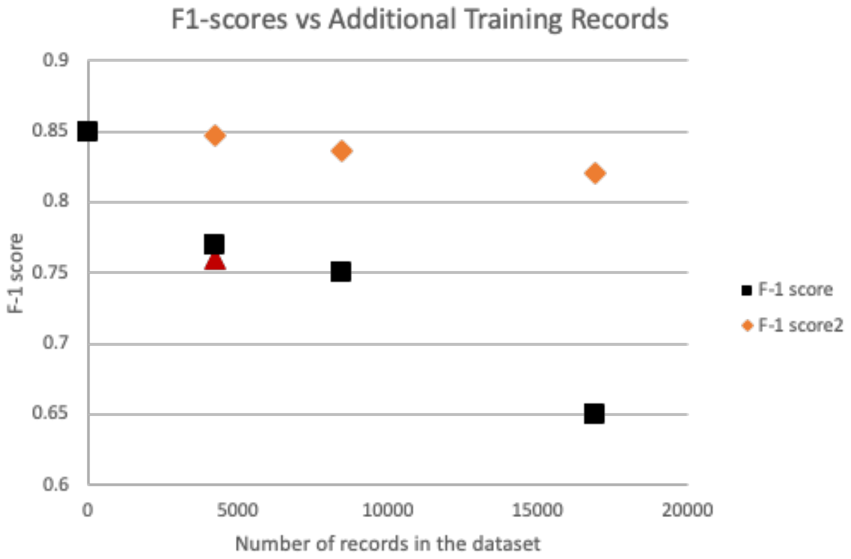


Figure 2. Performance of the model using additional training records which have consistent document ID’s.

To fully confirm if the current data augmentation would work for improving the model, I also trained each of the dropped models with the original training data which contains 18910 records. The F-1 scores were marked in orange in Figure 2. It did help to raise the F1 scores, but it failed to counteract the impact of the augmented data. The performance was still lower than the baseline.

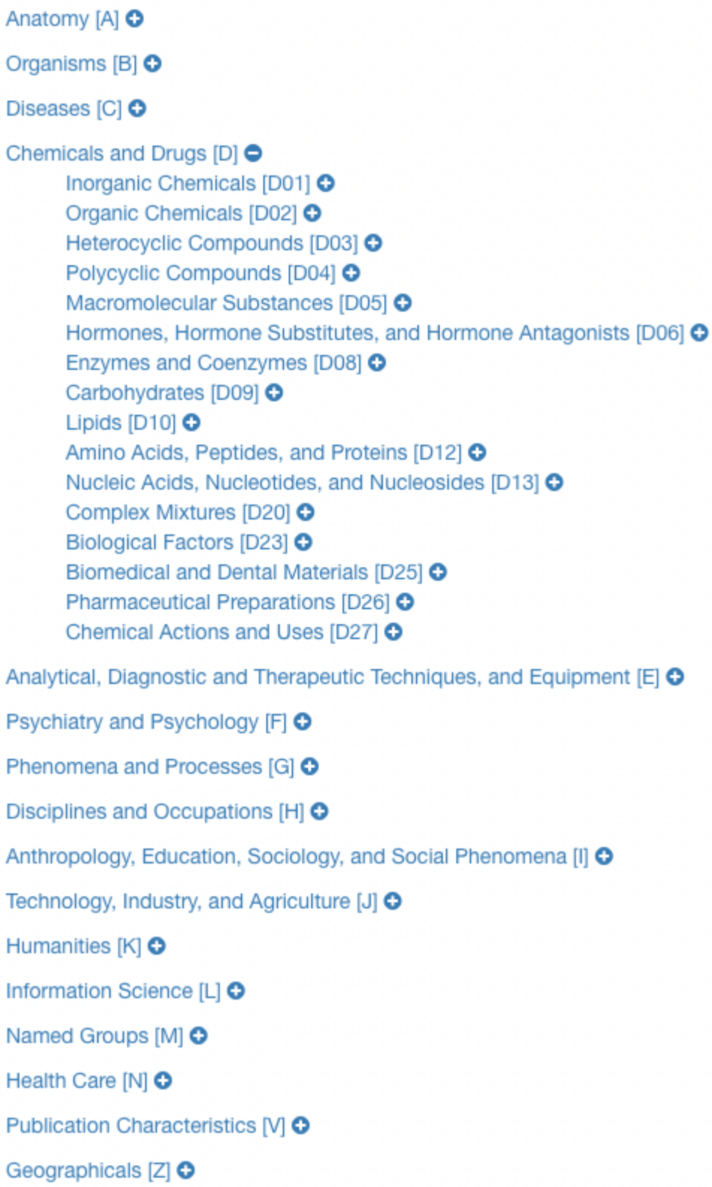
To the end of this semester-long project, a preprint research paper that conducted the same tasks became available online.[43] Their strategy of improving the performance is very similar to ours. However, instead of replacing chemical entities with meaningful MeSH terms, they used random text strings. Their findings confirmed that data augmentation of the original training data would not help improve the model much. Therefore, it is not surprising that my data augmentation did not work as we initially expected. To improve the model, I believe that different strategies should be explored.

Conclusion and Future work

This project has successfully identified the issues of the current BioBERT model in recognizing chemical entities in PubMed articles. While the model achieves a 0.85 F-1 score on the BC7T2-NLMCHEM dataset, it needs improvement in recognizing chemical abbreviations. To address this issue, additional training data was generated through data augmentation using MeSH terms. However, this approach did not help to improve the model yet.

Since a recent preprint research paper that conducted the same task also indicated that data augmentation could barely help the performance of the model and recognizing abbreviations was an issue, I concluded that different strategies should be implemented.

To make the data augmentation more meaningful for the model to learn is to substitute the chemical entities with chemical entities in the same category. This can be realized by mapping the entity names to the tree numbers provided by MeSH (Figure 3). So that using the tree numbers, other chemicals in the same category can be retrieved and used to replace the existing entities. The searching and locating of chemical names will require the implementation of the Trie data structure.



- Anatomy [A] ⊕
- Organisms [B] ⊕
- Diseases [C] ⊕
- Chemicals and Drugs [D] ⊖
    - Inorganic Chemicals [D01] ⊕
    - Organic Chemicals [D02] ⊕
    - Heterocyclic Compounds [D03] ⊕
    - Polycyclic Compounds [D04] ⊕
    - Macromolecular Substances [D05] ⊕
    - Hormones, Hormone Substitutes, and Hormone Antagonists [D06] ⊕
    - Enzymes and Coenzymes [D08] ⊕
    - Carbohydrates [D09] ⊕
    - Lipids [D10] ⊕
    - Amino Acids, Peptides, and Proteins [D12] ⊕
    - Nucleic Acids, Nucleotides, and Nucleosides [D13] ⊕
    - Complex Mixtures [D20] ⊕
    - Biological Factors [D23] ⊕
    - Biomedical and Dental Materials [D25] ⊕
    - Pharmaceutical Preparations [D26] ⊕
    - Chemical Actions and Uses [D27] ⊕
- Analytical, Diagnostic and Therapeutic Techniques, and Equipment [E] ⊕
- Psychiatry and Psychology [F] ⊕
- Phenomena and Processes [G] ⊕
- Disciplines and Occupations [H] ⊕
- Anthropology, Education, Sociology, and Social Phenomena [I] ⊕
- Technology, Industry, and Agriculture [J] ⊕
- Humanities [K] ⊕
- Information Science [L] ⊕
- Named Groups [M] ⊕
- Health Care [N] ⊕
- Publication Characteristics [V] ⊕
- Geographicals [Z] ⊕

Figure 3. The tree structure of MeSH terms.

Acknowledgment

This capstone project cannot happen without the direction of Dr. Steven Bethard. I am very grateful for his suggestions and guidance. The team members, Zeyu Zhang and Hyunju Song, also provided great support and help throughout the semester. It has been a great semester-long journey and I hope we will continue to work together in the future.

Reference

[1] U. Leser and J. Hakenberg, “What makes a gene name? Named entity recognition in the biomedical literature,” *Brief. Bioinform.*, vol. 6, no. 4, pp. 357–369, Dec. 2005, doi: 10.1093/bib/6.4.357.

[2] A. Mikheev, M. Moens, and C. Grover, “Named Entity Recognition without Gazetteers,” in *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, Bergen, Norway, Jun. 1999, pp. 1–8. Accessed: Nov. 24, 2021. [Online]. Available: https://aclanthology.org/E99-1001

[3] A. Goyal, V. Gupta, and M. Kumar, “Recent Named Entity Recognition and Classification techniques: A systematic review,” *Comput. Sci. Rev.*, vol. 29, pp. 21–43, Aug. 2018, doi: 10.1016/j.cosrev.2018.06.001.

[4] B. Alshaikhdeeb and K. Ahmad, “Biomedical Named Entity Recognition: A Review,” *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 6, p. 889, Dec. 2016, doi: 10.18517/ijaseit.6.6.1367.

[5] N. Perera, M. Dehmer, and F. Emmert-Streib, “Named Entity Recognition and Relation Detection for Biomedical Information Extraction,” *Front. Cell Dev. Biol.*, vol. 8, p. 673, 2020, doi: 10.3389/fcell.2020.00673.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” *ArXiv181004805 Cs*, May 2019, Accessed: Nov. 24, 2021. [Online]. Available: http://arxiv.org/abs/1810.04805

[7] K. M. Hettne *et al.*, “A dictionary to identify small molecules and drugs in free text,” *Bioinformatics*, vol. 25, no. 22, pp. 2983–2991, Nov. 2009, doi: 10.1093/bioinformatics/btp535.

[8] X. Wang, C. Yang, and R. Guan, “A comparative study for biomedical named entity recognition,” *Int. J. Mach. Learn. Cybern.*, vol. 9, no. 3, pp. 373–382, Mar. 2018, doi: 10.1007/s13042-015-0426-6.

[9] K. B. Cohen and L. Hunter, “Natural Language Processing and Systems Biology,” p. 27.

[10] J. Patrick and Y. Wang, “Biomedical Named Entity Recognition System,” p. 8.

[11] Y. Song, E. Kim, G. G. Lee, and B. Yi, “POSBIOTM—NER: a trainable biomedical named-entity recognition system,” *Bioinformatics*, vol. 21, no. 11, pp. 2794–2796, Jun. 2005, doi: 10.1093/bioinformatics/bti414.

[12] S. Zhang and N. Elhadad, “Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts,” *J. Biomed. Inform.*, vol. 46, no. 6, pp. 1088–1098, Dec. 2013, doi: 10.1016/j.jbi.2013.08.004.

[13] R. Klinger, C. Kolárik, J. Fluck, M. Hofmann-Apitius, and C. M. Friedrich, “Detection of IUPAC and IUPAC-like chemical names,” *Bioinforma. Oxf. Engl.*, vol. 24, no. 13, pp. i268-276, Jul. 2008, doi: 10.1093/bioinformatics/btn181.

[14] T. Tsai, W.-C. Chou, S.-H. Wu, T.-Y. Sung, J. Hsiang, and W.-L. Hsu, “Integrating linguistic knowledge into a conditional random fieldframework to identify biomedical named entities,” *Expert Syst. Appl.*, vol. 30, no. 1, pp. 117–128, Jan. 2006, doi: 10.1016/j.eswa.2005.09.072.

[15] B. Settles, “Biomedical Named Entity Recognition using Conditional Random Fields and Rich Feature Sets,” in *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, Geneva, Switzerland, Aug. 2004, pp. 107–110. Accessed: Nov. 24, 2021. [Online]. Available: https://aclanthology.org/W04-1221

[16] R. Leaman and G. Gonzalez, “Banner: an executable survey of advances in biomedical named entity recognition,” in *Biocomputing 2008*, WORLD SCIENTIFIC, 2007, pp. 652–663. doi: 10.1142/9789812776136_0062.

[17] N. Collier, C. Nobata, and J. Tsujii, “Extracting the Names of Genes and Gene Products with a Hidden Markov Model,” presented at the COLING 2000, 2000. Accessed: Nov. 25, 2021. [Online]. Available: https://aclanthology.org/C00-1030

[18] N. Ponomareva, F. Pla, A. Molina, and P. Rosso, “Biomedical Named Entity Recognition: A Poor Knowledge HMM-Based Approach,” in *Natural Language Processing and Information Systems*, Berlin, Heidelberg, 2007, pp. 382–387. doi: 10.1007/978-3-540-73351-5_34.

[19] D. Shen, J. Zhang, G. Zhou, J. Su, and C.-L. Tan, “Effective Adaptation of Hidden Markov Model-based Named Entity Recognizer for Biomedical Domain,” in *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, Sapporo, Japan, Jul. 2003, pp. 49–56. doi: 10.3115/1118958.1118965.

[20] A. McCallum, D. Freitag, and F. Pereira, “Maximum Entropy Markov Models for Information Extraction and Segmentation,” p. 26.

[21] B. Tang *et al.*, “A comparison of conditional random fields and structured support vector machines for chemical entity recognition in biomedical literature,” *J. Cheminformatics*, vol. 7, no. 1, p. S8, Jan. 2015, doi: 10.1186/1758-2946-7-S1-S8.

[22] H. Wei *et al.*, “Named Entity Recognition From Biomedical Texts Using a Fusion Attention-Based BiLSTM-CRF,” *IEEE Access*, vol. 7, pp. 73627–73636, 2019, doi: 10.1109/ACCESS.2019.2920734.

[23] L. Luo *et al.*, “An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition,” *Bioinformatics*, vol. 34, no. 8, pp. 1381–1388, Apr. 2018, doi: 10.1093/bioinformatics/btx761.

[24] L. Luo, P. Yang, Z. Yang, H. Lin, and J. Wang, “DUTIR at the BioCreative V.5.BeCalm Tasks: A BLSTM-CRF Approach for Biomedical Entity Recognition in Patents,” p. 12.

[25] R. Leaman, R. Islamaj Doğan, and Z. Lu, “DNorm: disease name normalization with pairwise learning to rank,” *Bioinformatics*, vol. 29, no. 22, pp. 2909–2917, Nov. 2013, doi: 10.1093/bioinformatics/btt474.

[26] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural Architectures for Named Entity Recognition,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, Jun. 2016, pp. 260–270. doi: 10.18653/v1/N16-1030.

[27] M. Habibi, L. Weber, M. Neves, D. L. Wiegandt, and U. Leser, “Deep learning with word embeddings improves biomedical named entity recognition,” *Bioinformatics*, vol. 33, no. 14, pp. i37–i48, Jul. 2017, doi: 10.1093/bioinformatics/btx228.

[28] W. Yoon, C. H. So, J. Lee, and J. Kang, “CollaboNet: collaboration of deep neural networks for biomedical named entity recognition,” *BMC Bioinformatics*, vol. 20, no. 10, p. 249, May 2019, doi: 10.1186/s12859-019-2813-6.

[29] T. H. Dang, H.-Q. Le, T. M. Nguyen, and S. T. Vu, “D3NER: biomedical named entity recognition using CRF-biLSTM improved with fine-tuned embeddings of various linguistic information,” *Bioinforma. Oxf. Engl.*, vol. 34, no. 20, pp. 3539–3546, Oct. 2018, doi: 10.1093/bioinformatics/bty356.

[30] S. K. Hong and J.-G. Lee, "DTranNER: biomedical named entity recognition with deep learning-based label-label transition model," *BMC Bioinformatics*, vol. 21, no. 1, p. 53, Feb. 2020, doi: 10.1186/s12859-020-3393-1.

[31] Z. Zhai, D. Q. Nguyen, and K. Verspoor, "Comparing CNN and LSTM character-level embeddings in BiLSTM-CRF models for chemical and disease named entity recognition," *ArXiv180808450 Cs*, Aug. 2018, Accessed: Nov. 23, 2021. [Online]. Available: http://arxiv.org/abs/1808.08450

[32] M. Cho, J. Ha, C. Park, and S. Park, "Combinatorial feature embedding based on CNN and LSTM for biomedical named entity recognition," *J. Biomed. Inform.*, vol. 103, p. 103381, Mar. 2020, doi: 10.1016/j.jbi.2020.103381.

[33] Q. Zhu, X. Li, A. Conesa, and C. Pereira, "GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text," *Bioinformatics*, vol. 34, no. 9, pp. 1547–1554, May 2018, doi: 10.1093/bioinformatics/btx815.

[34] I. Korvigo, M. Holmatov, A. Zaikovskii, and M. Skoblov, "Putting hands to rest: efficient deep CNN-RNN architecture for chemical named entity recognition with no hand-crafted rules," *J. Cheminformatics*, vol. 10, no. 1, p. 28, May 2018, doi: 10.1186/s13321-018-0280-0.

[35] Y. Peng, S. Yan, and Z. Lu, "Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets," in *Proceedings of the 18th BioNLP Workshop and Shared Task*, Florence, Italy, Aug. 2019, pp. 58–65. doi: 10.18653/v1/W19-5006.

[36] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A Pretrained Language Model for Scientific Text," *ArXiv190310676 Cs*, Sep. 2019, Accessed: Nov. 25, 2021. [Online]. Available: http://arxiv.org/abs/1903.10676

[37] J. Lee *et al.*, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, Feb. 2020, doi: 10.1093/bioinformatics/btz682.

[38] Y. Gu *et al.*, "Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing," *ACM Trans. Comput. Healthc.*, vol. 3, no. 1, pp. 1–23, Jan. 2022, doi: 10.1145/3458754.

[39] L. N. Phan *et al.*, "SciFive: a text-to-text transformer model for biomedical literature," *ArXiv210603598 Cs*, May 2021, Accessed: Nov. 25, 2021. [Online]. Available: http://arxiv.org/abs/2106.03598

[40] C. Raffel *et al.*, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.

[41] Z. Yuan, Y. Liu, C. Tan, S. Huang, and F. Huang, "Improving Biomedical Pretrained Language Models with Knowledge," *ArXiv210410344 Cs*, Apr. 2021, Accessed: Nov. 26, 2021. [Online]. Available: http://arxiv.org/abs/2104.10344

[42] H.-C. Shin *et al.*, "BioMegatron: Larger Biomedical Domain Language Model," *ArXiv201006060 Cs*, Oct. 2020, Accessed: Nov. 25, 2021. [Online]. Available: http://arxiv.org/abs/2010.06060

[43] A. Erdengasileng *et al.*, "A BERT-Based Hybrid System for Chemical Identification and Indexing in Full-Text Article." 2021. doi: 10.1101/2021.10.27.466183.