

American University in Cairo

AUC Knowledge Fountain

Theses and Dissertations

Student Research

Fall 2-15-2023

Named Entity Recognition from Biomedical Text

Maged Guirguis

magedrefaat@aucegypt.edu

Follow this and additional works at: <https://fount.aucegypt.edu/etds>



Part of the [Data Science Commons](#)

Recommended Citation

APA Citation

Guirguis, M. (2023). *Named Entity Recognition from Biomedical Text* [Master's Thesis, the American University in Cairo]. AUC Knowledge Fountain.

<https://fount.aucegypt.edu/etds/1983>

MLA Citation

Guirguis, Maged. *Named Entity Recognition from Biomedical Text*. 2023. American University in Cairo, Master's Thesis. *AUC Knowledge Fountain*.

<https://fount.aucegypt.edu/etds/1983>

This Master's Thesis is brought to you for free and open access by the Student Research at AUC Knowledge Fountain. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AUC Knowledge Fountain. For more information, please contact thesisadmin@aucegypt.edu.



The American
University in Cairo
الجامعة الأمريكية بالقاهرة

Graduate Studies

Named entity recognition from biomedical data

A THESIS SUBMITTED BY

Maged Guirguis

TO THE

Department of Computer science and engineering

8/15/2022

*in partial fulfillment of the requirements for the degree of
M.Sc. in Computer Science*

Declaration of Authorship

I, Maged Guirguis, declare that this thesis titled, "Named entity recognition from biomedical data" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Maged Refaat Rasmy Guirguis

Date:

8/15/2022

Abstract

As vast amounts of unstructured data are becoming available digitally, computer-based methods to extract relevant and meaningful information are needed. Information Extraction (IE) is the process of extracting relevant, useful, structured pieces of information from unstructured text. This field has been gaining a lot of attention from the scientific research community over the last period.

Named entity recognition (NER) is the task of identifying text spans that represent named entities, and to classify them into predefined categories. The NER task is one of the fundamental tasks in the information extraction domain and can be an initial step towards other tasks. A lot of research has been done in efforts to build better NER systems. Despite the existence of numerous and well-versed NER methods, they mainly focus on generic NER rather domain specific NER; NER tasks for biomedical domain remain under-studied. This research will be focusing on extracting relevant named entities from biomedical data.

The objective of this research is to identify an efficient technique for NER tasks from biomedical data. This is achieved by investigating using deep learning technologies namely pre-trained BERT [1] model and its variances SciBERT [2] and BioBERT [3]. Preprocessing the data before passing it for training influences model performance. There is also investigation with some preprocessing rules to monitor their effect on model performance.

To conduct this research, we built a baseline system and held different experiments to explore how changing certain factors would affect the results. Baseline system is initialized with BERT base model and it is finetuned on the ChemProt dataset [4] for 3 epochs with learning rate $3e-05$, **Precision:** 24.27%, **Recall:** 27.87%, **F1:** 25.94%. Based on the conducted experiments our findings are that initializing the system with SciBERT pre trained model and fine tuning it with ChemProt dataset has better results over other BERT variations. We also found out that applying preprocessing to the training data has a significant positive impact over model performance. Also, with the release of DrugProt [5] dataset, a newer version of ChemProt dataset we have the chance of increasing the training data which also have a positive impact over model performance. Our contribution is benchmarking for ChemProt dataset and building a baseline system for further research. Also, applying POS tagging in data preprocessing step helps filter out less relevant parts of text which improves model performance. The best

performance could be achieved by removing punctuation, stop words verbs and adjectives from the text and finetuning SciBERT pretrained model on DrugProt [5] with learning rate 3e-5 for 3 epochs, token level evaluation: **Precision:** 66.20%, **Recall:** 98.96%, **F1:** 79.33%, entity level evaluation: **Precision:** 47.62%, **Recall:** 77.34%, **F1:** 58.95%.

Acknowledgements

I would like to acknowledge Professor Ahmed Rafea for the guidance and knowledge that he provided throughout this research.

I would like to acknowledge Nada Gaballah for her continuous support and guidance throughout the MSc program.

Contents

Declaration of Authorship	1
Abstract	2
List of Figures	7
List of Tables	8
List of Abbreviations	9
Chapter 1	10
Introduction	10
1.1 Problem definition	10
1.2 Background.....	11
1.3 Research objective.....	12
1.4 Document Layout	12
Chapter 2	13
Literature review	13
2.1 Supervised learning approach	13
2.1.1 Rule based approach.....	13
2.1.2 Feature-based approach	14
2.1.3 Deep learning approach	14
2.2 Unsupervised approach.....	16
2.3 Hybrid approach.....	16
2.4 Ensemble classifiers	18
Chapter 3	19
Research objective and methodology.....	19
3.1 Research objective	19
3.2 Research methodology	19
Chapter 4	25
Baseline system.....	25
4.1 Biocreative VI ChemProt	25
4.2 Data preparation	27
4.3 Initializing the model	29
4.4 Running the model	29

4.5 Remarks.....	30
Chapter 5.....	31
Experimentations	31
5.1 Using pretrained SciBERT and BioBERT models	31
5.2 Preprocessing Impact	32
5.3 Using DrugProt dataset.....	34
5.4 Ensemble	36
5.5 Hierarchical entity extraction.....	38
5.6 Token level, and entity level evaluation analysis.....	40
Chapter 6.....	41
Conclusion and Future Work.....	41
References	43

List of Figures

Figure 1 Ensemble classifiers.....	18
Figure 2 Data preprocessing.....	22
Figure 3 Confusion matrix.....	24
Figure 4 Data preparation - entities.....	27
Figure 5 Data preparation - abstracts.....	28
Figure 6 Data preparation - abstracts.....	28
Figure 7 BERT model architecture.....	29
Figure 8 Evaluation Analysis.....	40

List of Tables

Table 1 Sample of abstracts	25
Table 2 Sample of entities file	26
Table 3 Baseline evaluation scores	30
Table 4 Experiment 1 results.....	32
Table 5 Experiment 2.1 results.....	33
Table 6 Experiment 2.2 results.....	34
Table 7 Experiment 3 results.....	35
Table 8 Experiment 4.1 results.....	36
Table 9 Experiment 4.2 results.....	37
Table 10 Experiment 5 results.....	39

List of Abbreviations

IE	Information Extraction
BERT	Bidirectional Encoder Representations from Transformers
NER	Named Entity Recognition
POS	Part Of Speech
UMLS	Unified medical language system
SVM	Support Vector Machines
CRF	Conditional Random Fields
HMM	Hidden Markov Models
DL	Deep learning
CNN	Convolution neural networks
RNN	Recurrent neural networks
LSTM	Long short-term memory
BI-LSTM	Bi-directional long short-term memory.
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

Chapter 1

Introduction

Information extraction is the process of data scanning looking for relevant pieces of information. Automated information extraction systems help in saving significant time and effort when getting specific information in a short time. With the recent medical challenges, we have been facing, it would be much easier to be up to speed and gain information that would help in overcoming these challenges by scanning the thousands of digitally available medical documents and extracting relevant information.

The problem at hand is extracting named entities from unstructured biomedical text and classifying them into pre-defined categories. Named entity recognition (NER) is a widely applicable natural language processing task and building block of question answering, topic modeling, information retrieval, etc. In the medical domain, NER plays a crucial role by extracting meaningful chunks from clinical notes and reports, which are then fed to downstream tasks like assertion status detection, entity resolution, relation extraction, and de-identification.

The following subsections elaborate more on the problem definition, and what is the scope of work we are doing. We will also go over background about named entity recognition (NER) and the document layout.

1.1 Problem definition

With the abundance of information on the internet, a great opportunity is presenting itself for being more educated and well informed. However, this comes with a drawback, that the data is not necessarily structured and extracting a relevant piece of information is becoming a challenging task. Different research has been concerned with extraction of useful structured information from unstructured information. Information extraction can be considered under the wide umbrella of text understanding. While in text understanding the main objective is to represent all information that can be found in the text, information extraction focuses on the

extraction of specific semantic information related to the extraction task [6]. Information extraction is used widely in extracting specific information from many documents, it is applied to scientific journals, hospital records, legal contracts, news records, etc. Applying information extraction helps reduce the human effort of going through many documents to extract answers to specific queries or relations between entities.

Though a lot of research has been done in the field of information extraction from text, fewer research was done for domain specific information extraction. Domain data is different in the sense of explicit terms that mean different things in different contexts and terms that are not common in everyday life.

The task of named entity recognition (NER) is probably the first step towards information extraction. Named entities extracted from text could be fed into a relation extraction task to extract relations between 2 entities. The presence of certain entities could also be an indication of a certain text classification. NER systems work by ingesting the input text and looking for named entities to extract and classify them into one of the predefined categories. There are two types of Named Entities (NE), Generic named entities like names of persons, organizations, locations, etc. and Domain Specific Named Entities like in the field of biomedical data, drug names, protein names, chemicals, etc.

This research will be focusing on extracting and classifying named entities from biomedical data.

1.2 Background

The origin of the term “Named Entity” (NE) was first introduced by R. Grishman and B. Sundheim in 1996, their main task was to identify names of all the people, organizations, and geographic locations in a text [7]. Later on, Petasis et al. [8] continued the work by limiting the definition of a named entity to proper nouns serving only as a name for something or someone. Following their work, there has been numerous efforts among researchers trying to settle on the type of nouns to be classified as NE. They came to agree on dividing NEs into two categories, generic NEs and domain specific NEs. Generic NEs like people names, locations, organizations,

etc. and Domain specific NEs to be defined according to the domain in question e.g., in biomedical domain proteins, drug names, diseases, etc. should be categorized as Named entities.

Named Entity Recognition and Classification (NERC), is the process of extracting and locating named entities within a text and classifying them into the different categories: person, organization, location, etc. It is considered to be an important step for text pre-processing and a vital phase in the process of Information Extraction (IE) [9].

1.3 Research objective

The objective of this research is to identify an efficient approach for extracting named entities from biomedical text and investigate the effects of applying different preprocessing methods on text before sending to the model.

1.4 Document Layout

This document contains the following chapters:

Chapter2 - Literature review: In this chapter we investigate what has already been done in the field. We also explore the related works to learn more about state-of-the-art technologies.

Chapter 3 - Research objective and methodology: In this chapter we highlight the main objective of this research. State the research questions and the steps to answer them.

Chapter 4 - Baseline system: In this chapter we give an overview of the dataset used, the data preparation steps. We will go over the model setup and the evaluation metrics used.

Chapter 5 - Experimentations: In this chapter we highlight the experiments conducted throughout this research, each experiment contains the hypothesis, setup observations, and results.

Chapter 6 - Conclusion and Future work: In this section we highlight our findings, conclusion and the future work proposed for this research.

Chapter 2

Literature review

NER techniques can be divided into two main approaches supervised and unsupervised learning approached. Supervised learning approaches are split into rule-based, feature-based, and deep learning approaches. In this section we through light on all the approaches that are presented.

2.1 Supervised learning approach

2.1.1 Rule based approach

This approach mainly relies on designed rules [10] thus no need for the presence of annotated data. Rules can be crafted according to the domain in question, based on common dictionaries in the domain [11] or syntactic-lexical patterns yielding better results in the cases of restricted domains. An example from the biomedical domain is ProMiner [12] which builds on a previously available dictionary of synonyms. It was initially proposed to help in the problem of identification of proteins and their gene names in text. The ProMiner system consists of three parts: dictionary generation, occurrence detection and filtering of matches. The first part is about generating a name dictionary by associating each biological entity with all known synonyms. The second part of the system is a highly sensitive search procedure that aims to detect all potential occurrences of a named entity and its synonyms in the text. The last part is for filtering and disambiguation to identify different types of named entities.

A drawback for this approach is the necessity of human expertise in the domain in question along with programming skills [13]. This approach consumes too much effort to design the system and fine tune it to a specific domain or application. In addition, it will not perform as good in a different domain [14]. Precision is generally high for these systems because of the lexicon; however, recall is often low because of domain and language-specific rules and often incomplete dictionaries [15].

2.1.2 Feature-based approach

These approaches are built on a carefully designed features set where each of them is engineered to represent a training example [20]. The feature engineering step is very crucial to the system. Features can be divided into 3 categories:

- Word-level features: if a word window of length five is used then five different features are considered which are w_0 (current word), w_{-1} (previous word), w_{+1} (next word), w_{-2} and w_{+2} . These five features form a feature group (forward word).
- Lexical features: Kazama et al. [21] selected the most frequent 10,000 words from the GENIA corpus as lexical features for the NE recognizer. The lexical feature set consists of three term lists: a single-term list, a functional-term list, and a general-term list. The single term list is a list of single words that can be used as an entity by itself. Functional term list is a list of terms that are devised to describe the function and characteristics of named entities. General terms are all terms that are classified neither as single terms nor function terms. There is no specific lexicon for general terms.
- Orthographical and morphological features: Orthographical features can be used for words that appear with very low frequency in the training corpus to alleviate the data sparseness problem.

These features shall be the seed of training in supervised machine learning algorithms like Support Vector Machines (SVM) [22], Conditional Random Fields [23], and Hidden Markov Models (HMM) [24].

2.1.3 Deep learning approach

Recently, DL-based approaches became the core of modern NER systems. Its main merits are the automatic identification of features as opposed to feature-based approaches and making the system more robust to domain change as opposed to rule-based approaches. A typical deep learning model consists of multiple layers of neural networks built on top of each other. These neural networks typically do a forward pass and a backward pass. In the forward

pass the weighted sum of the inputs is computed and passed through a nonlinear function. The backward pass computes the gradient of the function given the weights of the modules using the chain rule of derivatives [25]. The core Strengths of Deep Learning:

- Non-linear transformation, this produces non-linear mappings between the input and output which enables complex features learning yielding better results compared with linear models
- Deep Learning requires significantly less effort in feature design, it is effective in automating the learning of related representations.
- DL models can be trained end-to-end by gradient descent which enables more complex NER systems.

Convolution neural networks (CNN), recurrent neural networks (RNN), and their variant networks are the main application networks of this method. The application of convolutional neural networks to named entity recognition tasks was originally proposed by Collobert in 2019 [26]. Besides traditional convolutional neural networks, re-current neural networks have also been widely used in named entity recognition tasks. Several scholars opted to use a series of long-short-term memory network-based models, such as LSTM, BI-LSTM, and others [27].

2.1.3.1 KV-PLM, a unified pre-trained language model

KV-PLM [28], a unified pre-trained language model processing both molecule structures and biomedical text for knowledgeable and versatile machine reading. KV-PLM was developed by researchers at Tsinghua University, Beijing, China. KV-PLM takes the BERT[1] language model as its backbone. For the system to process the heterogeneous data in a unified model, it serializes molecule structures into SMILES [29] strings segment them using the BPE [30] algorithm. Then the system is pretrained using BERT masked language modeling task to learn the meta-knowledge between different semantic units. In the training phase, parts of the input tokens) are randomly masked, and the model is asked to reconstruct the masked tokens according to the context. In this way, the model can grasp the correlation between molecule

structure and biomedical text without any annotated data.

2.2 Unsupervised approach

The NLP research community has invested a lot of efforts in unsupervised approaches for these approaches do not rely on hand-labeled data. Unsupervised approaches aim to use automated algorithms to extract named entities from the text without relying on external resources or human intervention. A typical use case for this scenario is clustering. Early works rely on heuristics rules and lexical resources [16] [17] [18]. Zhang and Elhadad [19] introduced a novel unsupervised approach for NE extraction in the biomedical field. Their system opts to use terminologies and corpus statistics and minimal syntactic knowledge as a replacement for supervision. The first step in their system is seed term collection. In this step a dictionary is collected for each entity; this dictionary is supposed to contain a set of known terms for each class. Second step in boundary detection; by using a noun phrase chunker and inverse document frequency calculation. Then by filtering the noun phrases whose IDF value is lower than a certain threshold. Third step is entity classification; in this step a signature for each class is calculated and then a cosine similarity is calculated between each candidate word and a certain class signature. Based on the similarity calculation each word is assigned a class accordingly. If the similarity of the word to all classes is lower than the threshold, it is removed from the set of recognized named entities.

2.3 Hybrid approach

With the continuous advancements that are going in the named entity recognition field, a lot of researchers have a direction of utilizing a collection of the previous approaches, hybrid approach. By combining two or more approaches from the previous ones, researchers are able to overcome the limitations of each of them and capitalize on the strengths of each approach.

2.3.1 LSTM-CRF

Based on the methods suggested by Lample et al [31], research in Humboldt university in Berlin developed a domain-independent NER system that is independent of any kind of background knowledge. LSTM-CRF [32] combines the power of word embeddings, LSTMs and

CRFs into a single method for biomedical NER. The proposed method is completely agnostic to the type of the entity; all it requires is annotated data and word embeddings pre-computed on a large biomedical corpus. The system is comprised of three main layers. The first layer is the embedding layer which receives the raw sentence S made of the sequence of words and produces an embedding for each word in S . These are fed into a bi-directional LSTM-layer that produces a refined representation of the input, which is the input to a final CRF-layer.

2.3.2 BioNER

BioNER [27] combines a bi- directional LSTM network and a CRF network to form an BI-LSTM-CRF model. In addition to the past input features and sentence level tag information used in a typical LSTM-CRF model, a BI-LSTM-CRF model can use the future input features. The input data is passed to a POS module that assigns each word with a unique tag that indicates its syntactic role. In chunking, each word is tagged with its phrase type. For example, tag B-NP indicates a word starting a noun phrase. Then comes the phase of feature extraction, spelling features and context features are extracted. Then comes a layer of word embedding that plays a vital role in improving sequence tagging performance. BioNER is robust, and it has less dependence on word embedding as compared to the observation made by Collobert et al. [26].

2.3.3 XLNet-CRF

XLNet-CRF [33] uses XLNet [34] based on Self-Attention Permutation Language Model (PLM) to replace BERT as encoder in the pre-training phase. This avoids the problem of input noise from autoencoding language model (AutoEncoder LM). When fine-tuning the BioNER task, the output of the XLNet model is decoded with conditional random field (CRF) decoder. Because XLNet uses tagged input, the connection layer between XLNet and CRF is tuned with Label [X]. At first, text is serialized, and the input sequence is generated by the SentencePiece [35] based on the input text. Then, the input is word-embedded, and each input character is mapped to a vector which is the input to the following the multi-header attention model. Finally, the output vector of the final XLNet model after the attention model is linked by the n layer residue to the CRF layer which is used as the decoding layer to select the most appropriate label from the label collection. A is defined as a transition matrix to modify the current forecast

based on previous label information.

2.4 Ensemble classifiers

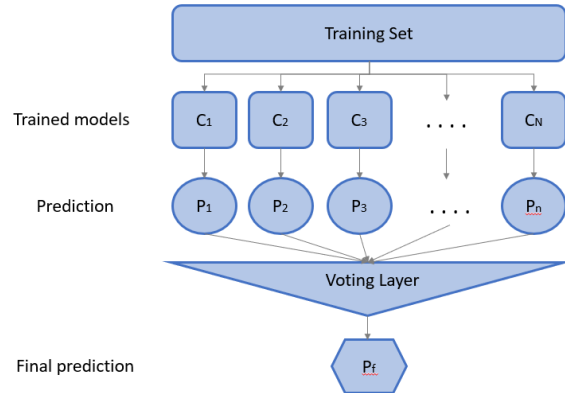


Figure 1 Ensemble classifiers

Ensemble classifiers combine the results of several models to improve the results by the collective system. Utilizing different models enables the collective system to have a better predictive performance compared to a single model. As shown in Figure 1, the basic idea is to get the predictions from multiple models and allow them to vote to reach a final consensus for the prediction. The most challenging part of ensemble classifiers is not finding good performing model, it is finding models that make different types of mistakes. This way the ensemble system can build on the strengths of all the underlying models.

Perhaps one of the earliest works on ensemble systems was the work discussed partitioning the feature space using two or more classifiers [36]. For biomedical NER Zhou et al., [37] proposed an ensemble of classifiers in which they used Support Vector Machine (SVM) and two discriminative Hidden Markov Models (HMM), and applied a weighted majority voting strategy to combine the output of the classifiers, which led to an improved F1 score.

Chapter 3

Research objective and methodology

In the following sections we state the research objective and the research questions we are trying to answer throughout this research. We also highlight the steps followed to answer the research questions.

3.1 Research objective

The objective of this research is to identify an efficient approach for extracting named entities from biomedical text and investigate the effects of applying different preprocessing methods on text before sending to the model.

To achieve this objective a set of research questions is proposed:

1. Will using a specific version of pretrained models of BERT like SciBERT or BioBERT lead to efficiently extracting named entities from biomedical text?
2. Will using different methods in preprocessing improve the model performance?
3. Does increasing the size of training data improve performance.
4. Does using ensemble classifiers improve scores?
5. Does building a multi-level hierarchical extraction model improve model performance?

3.2 Research methodology

In this section we highlight the proposed methodology to answer the research questions. To answer the first research question, we will follow the first two steps, and to answer the second research question we will apply the third step. To answer the third question, we will follow the fourth step. To answer the fourth question, we will follow the fifth step.

1. Build a baseline system, initialize it with BERT base model, and train it on a biomedical dataset.
2. Use the same setup, initialize the model with SciBERT and BioBERT pretrained models, observe the change in evaluation scores.
3. Apply different preprocessing methods and observe changes in overall model performance.
4. Experiment increasing training data by using DrugProt dataset[5].
5. Build an ensemble classification model to combine predictions from more than one model.
6. Build a 2-level hierarchical extraction model.

In the following subsections we will provide more details about each of the steps. An overview of the baseline systems and BERT variations that will be used throughout the research. We will also go over the rules and tools used in the preprocessing step. For post processing, we will provide the purpose of this step and the rules. The datasets and evaluation method are also described.

3.2.1 Baseline system

As a part of this research, we will be building an NER system as a test bed for our experiments. The Baseline system is expected to extract pre-defined, domain-specific named entities from biomedical unstructured text. We will be using BERT-based model for this task, and we will use the ChemProt [4] dataset for benchmarking.

3.2.2 BERT Variations

Since the release of the BERT model, a lot of researchers have increased interest in creating domain specific versions of BERT. In the research we will focus mainly on SciBERT and BioBERT since they are finetuned with data for biomedical domain.

BioBERT

BioBERT [3] is a variation of the BERT model from Korea University and Clova AI. It basically has the same structure as BERT. It is initialized with weights from BERT, which was pre-trained on general domain corpora. Then, BioBERT is trained with biomedical domain corpora (PubMed abstracts and PMC full-text articles). PubMed is a database of biomedical citations and abstractions, whereas PMC is an electronic archive of full-text journal articles. Their contributions were a biomedical language representation model that could manage tasks such as relation extraction and drug discovery to name a few. By having a pre-trained model that encompasses both general and biomedical domain corpora, developers and practitioners could now encapsulate biomedical terms that would have been incredibly difficult for a general language model to comprehend.

SciBERT

SciBERT [2] is a pre-trained BERT-based language model for performing scientific tasks in the field of Natural Language Processing (NLP). It was introduced by researchers at the Allen Institute for Artificial Intelligence (AllenAI) in September 2019. SciBERT follows the same architecture as BERT but trained on scientific text instead of general corpora. There are 4 different versions of SciBERT: `basevocab_cased`, `basevocab_uncased`, `scivocab_cased` and `scivocab_uncased`. The `basevocab` models are initialized with BERT model and finetuned on the scientific data. The `scivocab` models are trained from scratch on the scientific corpora. SciBERT is trained on a large multi-domain corpus of scientific publications to improve performance on downstream scientific NLP tasks. SciBERT is trained on a random sample of 1.14M papers from Semantic Scholar [38]. This corpus consists of 18% papers from the computer science domain and 82% from the broad biomedical domain.

3.2.3 Preprocessing

In natural language processing, text preprocessing is the practice of cleaning and preparing text data before using it to the needed task. The goal of cleaning and preparing text data is to reduce the text to only the words that you need for the task. Preprocessing the data before feeding it into the model could play an important role in model performance.

Tokenization is one of the most basic yet fundamental tasks in text preprocessing. Tokenization is the process of breaking down a piece of text into small units called tokens. A token may be a word, part of a word or just characters like punctuation. Preprocessing of data could follow on the following themes, data cleaning/filtering, data ordering or data augmentation, etc. Data cleaning is the process of adding missing data and correcting, repairing, or removing incorrect or irrelevant parts in the data. Data filtering is the process of removing parts of the data based on the filtering conditions applied. Data ordering is the process of arranging the data into some meaningful order to make it easier to understand, analyze or visualize. Data augmentation is the process of adding extra indicators or pointers to your data that should not alter your data itself or remove from it.

For the data preparation step for this research, we start off by tokenizing the named entities in the dataset and the abstracts. As a result of the tokenization step, labeled named entities in the dataset are split into one or more tokens. We have to expand the labels to cover the new tokenization entities for that we apply some basic data augmentation by using the BIO (Beginning, Inside, Outside) format [39] to augment labels of the entities that could be composed of more than one token. The first token would have “B-” prefix to their label and all subsequent labels will have “I-” prefix to their label, all other tokens that are not an entity are labelled “O”. BIO format is a common format for chunking tags and tokens in NER tasks. An example of this process is given in Figure 2.

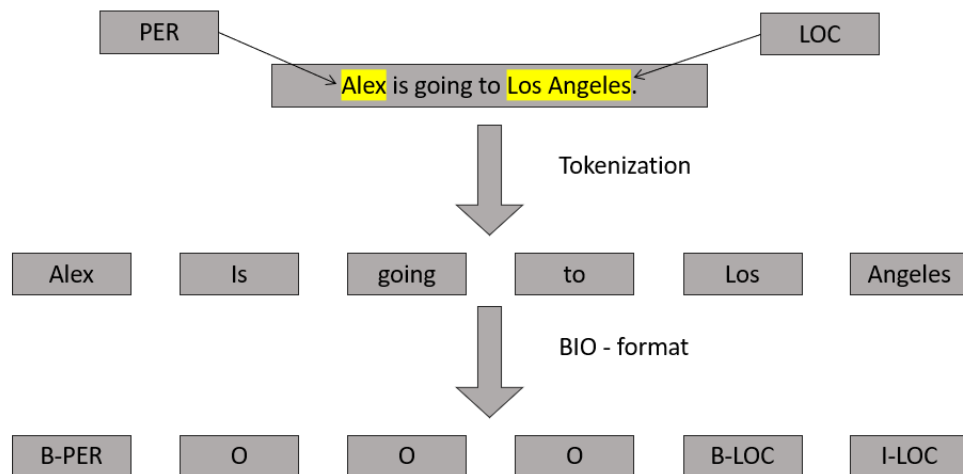


Figure 2 Data preprocessing

For data filtering, we investigate the effect of adding some basic preprocessing rules like dropping out punctuation, white spaces and stop words (articles, pronouns, prepositions, and conjunctions). We also investigate removing irrelevant parts of text based on part-of-speech tagging. Part-of-speech tagging (POS tagging) aims to identify which grammatical group a word belongs to, i.e., whether it is a noun, adjective, verb, adverb, etc., based on the context. For the POS tagger we use the common Natural Language Toolkit (NLTK) [40]. It takes a sentence as input and returns back list of tokens and their respective POS tags (e.g., NN (noun, singular or mass), VB (verb, base form), and VBD (verb, past tense)) [41], [42], [43]. Based on the tags returned by POS tagging we remove the adjectives and verbs (not likely to be named entities) from the training data before passing to the model.

3.2.4 Datasets

With the release of a newer version of ChemProt [4] dataset, the DrugProt [5] dataset, there is a chance to increase the size of the training set. The newly released dataset only contains a training set consisting of 3,500 abstracts with 195,000 labeled entities which is a superset of the chemprot training set; we will use the development and test sets of the chemprot set for validation and testing.

3.2.5 Evaluation

For the evaluation process we used the same data preparation steps for the model. Then we pass the list of tokenized abstracts to the system to predict labels for each token. The output of this process is a list of predicted labels for each input token respectively. We will be using 2 evaluation methods, the first one is token level, second one in entity level. The first method is simple, we compare the list of predicted labels to the list of expected tokens and calculate the metrics. For the second method, since each named entity can be broken down into multiple tokens and the model is trained on the token level and thus generates prediction on token level. So, to compute entity level evaluation, we will be grouping the token of the named entity and assign them a single label (based on the majority of the labeled tokens). This method guarantees that the whole named entity is extracted and given the correct class.

Given True Positive (TP): where prediction is the same as ground truth and both are not "O", True Negative (TN): where prediction is the same as ground truth and both are "O", False Positive (FP): where prediction is not the same as ground truth and prediction is not "O" and False Negative (FN): where prediction is not the same as ground truth, and the prediction is "O". Based on these definitions the count of TP, TN, FP and FN, a confusion matrix is constructed as shown in Figure 3 and evaluation metrics are calculated.

<p style="text-align: center;">TP Prediction == Ground truth Prediction != "O"</p>	<p style="text-align: center;">FN Prediction != Ground truth Prediction == "O"</p>
<p style="text-align: center;">FP Prediction != Ground truth Prediction != "O"</p>	<p style="text-align: center;">TN Prediction == Ground truth Prediction == "O"</p>

Figure 3 Confusion matrix

Precision is the measure of how precise/accurate the system is. It is the ratio between the True Positives and all the Positives. Precision reveals out of predicted positive entities, how many of them are actual positive (belong to the right entity type as predicted). $\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$.

Recall is the measure of the model's ability to extract actual positive entities. It is the ratio between the predicted True Positives and the labeled positives. Recall reveals out of actual tagged entities, how many of them are predicted correctly. $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$.

F1 Score is a combination of the Precision and Recall metrics, which measures the overall performance of the model. $\text{F1 Score} = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$.

Chapter 4

Baseline system

The following section will elaborate on the development of the baseline system. The benchmark dataset will be described, the data preparation will be explained, and steps for building the model are elaborated, and the evaluating metrics will be given.

4.1 Biocreative VI ChemProt

Biocreative VI ChemProt dataset [4], consists of abstracts of biomedical papers collected from PubMed. For each abstract there is a list of entities with their indices and respective labels. Each entity is tagged as a Chemical, Gene-Y or Gene-N. GENE-Y is a gene/protein mention type that can be normalized or associated to a biological database identifier while GENE-N is gene/protein mention type that cannot be normalized to a database identifier. You can see a sample of the abstract's files in Table 1 below, each abstract entry contains the abstract ID, title, and the abstracts text.

Table 1 Sample of abstracts

Abstract id	Title	Abstract text
23552263	Lipoxygenase and urease inhibition of the aerial parts of the Polygonatum verticillatum.	Over expression of lipoxygenase (LOX) and urease has already contributed to the pathology of different human disease. Targeting the inhibition of these enzymes has proved great clinical utility. The aim of the present study was to scrutinise the inhibitory profile of the aerial parts of the Polygonatum verticillatum enzyme against LOX, urease, acetylcholinesterase (AChE) and butyrylcholinesterase (BChE) using standard experimental protocols.....

Table 2 contains part of the entities file; each entry contains the abstract ID where the entity is observed entity ID with is a sequential identifier for entities per abstract. Entity label and start and end offsets of the entity and the entity text itself.

Table 2 Sample of entities file

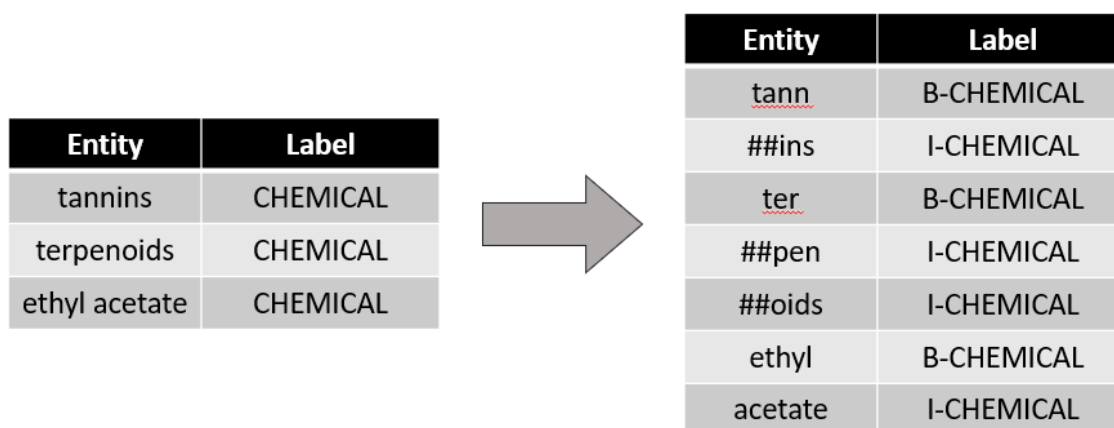
Abstract id	Entity id	Entity label	Start offset	End offset	Entity name
23552263	T1	CHEMICAL	1090	1097	tannins
23552263	T2	CHEMICAL	1102	1112	terpenoids
23552263	T3	CHEMICAL	646	659	ethyl acetate
23552263	T4	CHEMICAL	822	831	n-butanol
23552263	T5	CHEMICAL	1048	1056	saponins
23552263	T6	CHEMICAL	1069	1079	flavonoids
23552263	T7	CHEMICAL	1081	1088	phenols
23552263	T8	GENE-N	108	120	lipoxygenase
23552263	T9	GENE-N	122	125	LOX
23552263	T10	GENE-N	423	426	LOX
23552263	T11	GENE-N	428	434	urease
23552263	T12	GENE-Y	436	456	acetylcholinest erase
23552263	T13	GENE-Y	458	462	AChE
23552263	T14	GENE-Y	468	489	Butyrylcholine sterase
23552263	T15	GENE-Y	491	495	BChE
3552263	T16	GENE-N	131	137	urease
23552263	T17	GENE-N	557	569	lipoxygenase
23552263	T18	GENE-N	803	809	urease

Data is split into 3 sets: train, development, and test. The training set consists of 1020 abstract records with 25,752 labeled entities. The development set consists of 612 abstract records with 15,567 labeled entities. The test set consists of 900 abstract records with 20,828 labeled entities.

The data provided consists of two files in tab separated formats. One file includes the abstracts along with their IDs and paper titles. The second file includes the annotated chemical and gene entities mentioned in each abstract.

4.2 Data preparation

Based on the above section describing how data is available in the ChemProt dataset, we did the following. We started by importing the entities file and making a list of all the entities available in our data and each can be found in which abstract. Entities sometimes span more than one token and since we are doing token level labeling, we need to tokenize the entities and their labels as well to match our design. For this we used the BIO (Beginning, Inside, Outside) schema; the first token would have "B-" prefix to their label and all subsequent labels will have "I-" prefix to their label, all other tokens that are not an entity are labelled "O". Next, we tokenize the abstracts and for each token we assign a label based on the created entity labels list. Figure 4 shows the construction of the entities' labels list with BIO schema.



Entity	Label
tannins	CHEMICAL
terpenoids	CHEMICAL
ethyl acetate	CHEMICAL

Entity	Label
<u>tann</u>	B-CHEMICAL
##ins	I-CHEMICAL
<u>ter</u>	B-CHEMICAL
##pen	I-CHEMICAL
##oids	I-CHEMICAL
ethyl	B-CHEMICAL
acetate	I-CHEMICAL

Figure 4 Data preparation - entities

After constructing the list of entities, we get to tokenize each abstract and label it. This can be elaborated in Figure 5.

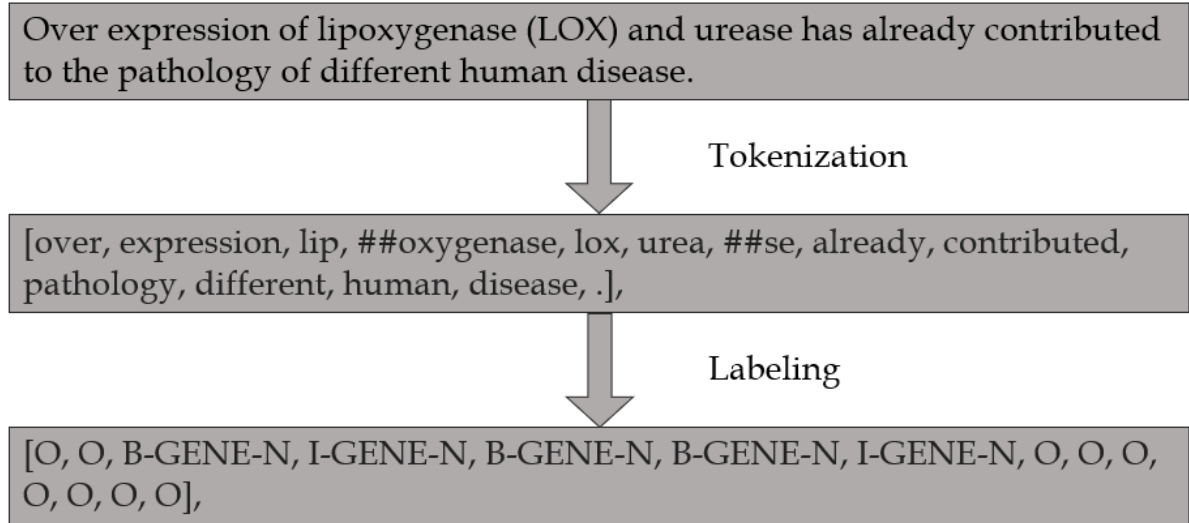


Figure 5 Data preparation - abstracts

The input to the model is two lists, one of tokenized abstracts and one is list of labelled abstracts. Each entry in the first list is a list of tokens in an abstract and each entry in the second abstract is a list of labels corresponding to the tokens in the first list. You can observe the format in Figure 6.

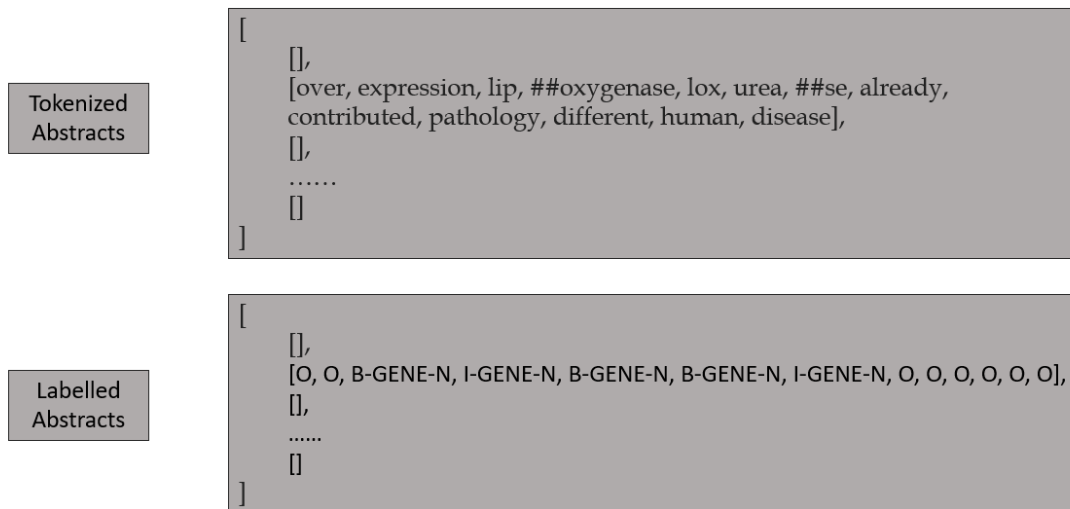


Figure 6 Model data input format

4.3 Initializing the model

We use the same architecture as BERT models which is a Transformer block of 12 layers, a hidden block of 768 layers with ReLU activation function then a self-attention block of 12 layers and an output layer with SoftMax activation function and the model's loss function is Cross-Entropy function. The BERT model architecture can be seen in Figure 7 [1].

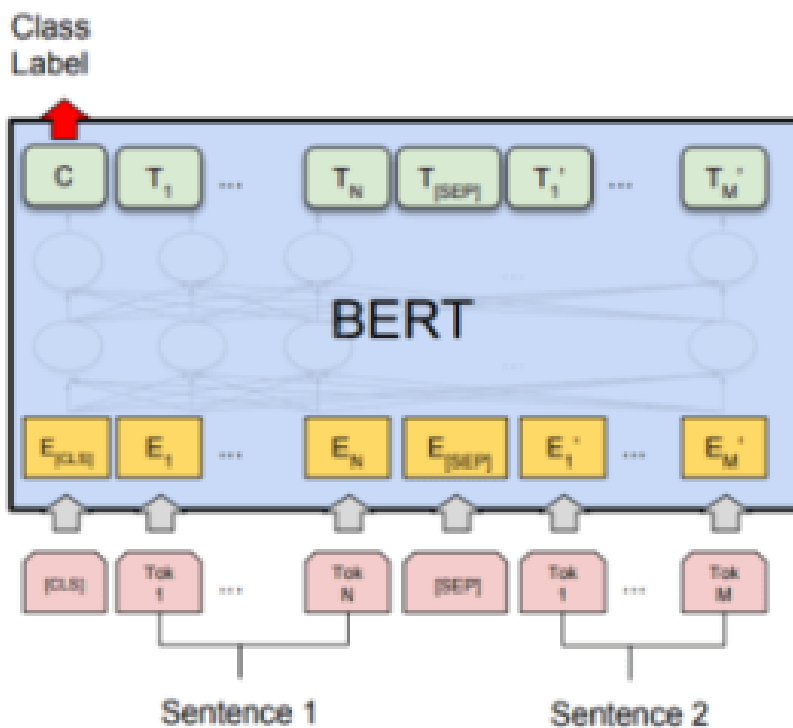


Figure 7 BERT model architecture

4.4 Running the model

After prepping the training data and initializing the model we do parameter tuning to find optimal parameters to train the model with. After initializing the model with BERT base model, we train it for one of the following epochs counts (3, 8 ,20) with one of the following learning rates (3e-4, 8e-5, 3e-5, 3e-6). We use the ChemProt training set for training and the development set for evaluation. Based on the results of the tuning step the optimal model parameter are 3 epochs and learning rate of 3e-5. Then we use the training and development set for training and the test for testing. Table 3 shows the baseline evaluation scores.

Table 3 Baseline evaluation scores

Token level eval	Precision	Recall	F1 Score
BERT (Baseline)	24.27%	27.87%	25.94%

4.5 Remarks

The observed results for the baseline system are way lower than expected. This is because we initialized the baseline system using the vanilla BERT model which is trained on general domain corpora and there was no preprocessing of any sorts for the data.

Chapter 5

Experimentations

In this chapter we highlight the experiments conducted throughout this research, each experiment contains the hypothesis, setup observations, and results. These experiments use bert-sklearn [44] which is scikit-learn wrapper for BERT based on the Hugging Face [45] and PyTorch [46] port. We run these experiments on Amazon Elastic Compute Cloud P3 machines. The machine is equipped with high frequency Intel Xeon E5-2686 v4 (Broadwell) processors 2ith 16 vCPUs and a Tesla V100 GPU with 5,120 CUDA Cores and 640 Tensor Cores.

5.1 Using pretrained SciBERT and BioBERT models

This experiment was conducted to answer the first research questions by applying step 1 and step 2 in the methodology

Hypothesis

Initializing the system with SciBERT or BioBERT models would improve model performance since these models are trained biomedical data.

Setup

For this experiment we will use the same setup in the baseline system, but we will be changing the tokenizer and the base model to SciBERT and BioBERT. After tokenization and data preparation, we will initialize the model with the pretrained weights from SciBERT and BioBERT base models and train each for 3 epochs with learning rate $3e-5$.

Results and findings

Table 4 shows the results of this experiment:

Table 4 Experiment 1 results

Token level eval	Precision	Recall	F1 Score
SciBERT	23.07%	33.34%	27.27%
BioBERT	24.87%	25.50%	25.18%

In table 4 we can observe the results of finetuning both SciBERT and BioBERT models on ChemProt dataset. The results observed for the SciBERT and BioBERT model are comparable to those of the BERT base model, no significant improvement could be noticed.

5.2 Preprocessing Impact

This experiment was conducted to answer the second research question by applying step 3 in the methodology section. This experiment is divided into two parts, the first is by adding basic preprocessing rules and the second part is adding POS tagging to filter out less meaningful parts of text.

Hypothesis

Dropping less meaningful tokens from training data passed to the model would improve model performance.

5.2.1 Apply basic preprocessing rules

Setup

For this experiment we will add some basic preprocessing rules like dropping out punctuation and stop words (articles, pronouns, prepositions, and conjunctions).

Results and findings

The following table shows the results of this experiment:

Table 5 Experiment 2.1 results

Token level eval	Precision	Recall	F1 Score
BERT (preprocessing)	54.92%	84.19%	66.47%
SciBERT (preprocessing)	61.06%	86.62%	71.63%
BioBERT (preprocessing)	55.13%	85.39%	67.00%
Entity level eval	Precision	Recall	F1 Score
BERT (preprocessing)	16.15%	76.44%	26.66%
SciBERT (preprocessing)	18.39%	78.62%	29.81%
BioBERT (preprocessing)	16.60%	77.51%	27.35%

In table 5 we can observe the results of finetuning both SciBERT and BioBERT models on ChemProt dataset while applying basic preprocessing rule. We can observe a significant improvement of scores because of the added rules. This indicates that the filtered entities were having a negative impact on the model performance. We can also observe higher scores for SciBERT model while no significant improvement for BioBERT model. Precision: 61.06%, Recall: 86.62%, F1: 71.3%.

5.2.2 Part of speech tagging in preprocessing step

Setup

For this experiment we will add an extra step in preprocessing phase of data preparation where we pass the whole abstract to a POS tagging module and based on the output of this module; we remove verbs and adjectives from the text before passing to the model.

Results and findings

The following table shows the results of this experiment:

Table 6 Experiment 2.2 results

Token level eval	Precision	Recall	F1 Score
SciBERT (preprocessing)	61.06%	86.62%	71.63%
SciBERT (preprocessing + POS)	65.93%	93.46%	77.32%
BioBERT (preprocessing)	55.13%	85.39%	67.00%
BioBERT (preprocessing + POS)	55.18%	95.92%	70.06%
Entity level eval	Precision	Recall	F1 Score
SciBERT (preprocessing)	18.39%	78.62%	18.39%
SciBERT (preprocessing + POS)	46.02%	81.92%	58.94%
BioBERT (preprocessing)	16.60%	77.51%	27.35%
BioBERT (preprocessing + POS)	40.52%	57.90%	47.68%

In the table above we can observe the results of adding POS tagging into the preprocessing and finetuning on both SciBERT and BioBERT model. A significant increase in F1 score is observed when removing verbs and adjectives based on POS tagging in the preprocessing step. Precision: 65.93%, Recall: 93.46%, F1:77.32%. A significant increase in F1 score is observed when removing verbs and adjectives based on POS tagging in the preprocessing step.

5.3 Using DrugProt dataset

This experiment was conducted to answer the third research question by applying step 4 in the methodology.

Hypothesis

The increase in training data would improve the results.

Setup

DrugProt [5] dataset which is a superset of the ChemProt training set will be used for training instead of the training set of ChemProt. Since we are changing the dataset, we will repeat the parameter tuning step to find optimal parameters to train the model with. After initializing the model with both SciBERT and BioBERT base models. We train it for one of the following epochs counts (3, 8 ,20) with one of the following learning rates (3e-4, 8e-5, 3e-5, 3e-6). After we settle on the optimal parameters, we use both the training and development sets to train the model and evaluate it on the test to measure the improvement with the change of the dataset.

Results and findings

The following table shows the results of this experiment:

Table 7 Experiment 3 results

Token level eval	Precision	Recall	F1 Score
SciBERT (ChemProt)	65.93%	93.46%	77.32%
SciBERT (DrugProt)	66.20%	98.96%	79.33%
BioBERT (ChemProt)	55.18%	95.92%	70.06%
BioBERT (DrugProt)	55.68%	99.48%	71.40%
Entity level eval	Precision	Recall	F1 Score
SciBERT (ChemProt)	46.02%	81.92%	58.94%
SciBERT (DrugProt)	47.62%	77.34%	58.95%
BioBERT (ChemProt)	40.52%	57.90%	47.68%
BioBERT (DrugProt)	43.16%	55.19%	48.44%

Based on the finetuning step of this experiment, the best results are observed for SciBERT when model is finetuned on DrugProt for 3 epochs with learning rate 3e-5 and for

BioBERT model when finetuned on DrugProt for 3 epochs with learning rate 3e-5. Based on the results in Table 7, SciBERT model is outperforming BioBERT model. A significant increase in F1 score for SciBERT model is observed when DrugProt dataset is used in the training. Precision: 62.20%, Recall: 98.96%, F1: 79.33%.

5.4 Ensemble

This experiment was conducted to answer the fifth research question by applying step-6 in the methodology. This step will be split into 2 experiments, one of them is by using SciBERT, BioBERT, BERT models in the ensemble classifiers. The other is by splitting the training data among multiple SciBERT models and using them in the ensemble classifier.

Hypothesis

Combining the predictions of more than one model would improve the scores.

5.4.1 Different models

Setup

After preparing the training data from the DrugProt dataset we use three models, one initialized with SciBERT, one initialized with BioBERT, and one initialized with BERT model and train each of them according to their optimal parameters. After training the model we get the predictions from each of the models and pass it to a voting layer that takes a vote from each model and returns the label with the most votes.

Results and findings

The following table shows the results of this experiment:

Table 8 Experiment 4.1 results

Token level eval	Precision	Recall	F1 Score
SciBERT (5.3)	66.20%	98.96%	79.33%
SciBERT, BioBERT, BERT (Ensemble)	62.56%	98.86%	76.63%

Entity level eval	Precision	Recall	F1 Score
SciBERT (5.3)	47.62%	77.34%	58.95%
SciBERT, BioBERT, BERT (Ensemble)	47.41%	75.28%	58.18%

Based on the observed results in Table 8, there is a drop in the scores which means that the results from the other two models are lowering the performance of the SciBERT model.

5.4.2 Split dataset

Setup

After preparing the training data from the DrugProt dataset we split into multiple separate sets, each set contains some randomly selected datasets from the training data. We initialize multiple models with SciBERT base model and finetune it on each of the sets. After all the models are trained, we pass their predictions into a voting layer that takes a vote from each model and returns the label with the most votes.

Results and findings

The following table shows the results of this experiment:

Table 9 Experiment 4.2 results

Token level eval	Precision	Recall	F1 Score
SciBERT (5.3)	66.20%	98.96%	79.33%
SciBERT 5 models dataset size 2000 (Ensemble)	66.13%	99.00%	79.30%
SciBERT 5 models dataset size 1500 (Ensemble)	66.06%	98.95%	79.23%

SciBERT 7 models dataset size 1500 (Ensemble)	66.08%	98.94%	79.24%
Entity level eval	Precision	Recall	F1 Score
SciBERT (5.3)	47.62%	77.34%	58.95%
SciBERT 5 models dataset size 2000 (Ensemble)	48.21%	77.24%	59.36%
SciBERT 5 models dataset size 1500 (Ensemble)	47.71%	73.92%	58.00%
SciBERT 7 models dataset size 1500 (Ensemble)	48.27%	76.36%	59.15%

Based on the observed results in Table 9, there is a slight drop in the scores when splitting the dataset, regardless of the count of splits or the size of the dataset, there seems to be no improvement in the scores.

5.5 Hierarchical entity extraction

This experiment was conducted to answer the sixth research question by applying step-7 in the methodology.

Hypothesis

By building a 2-level extraction system, a first level to extract CHEM and GENE entities and a second level to classify between GENE-N and GENE-Y named entities, this would lead to better overall performance.

Setup

After preparing the training data from the DrugProt dataset, we create a version of it where we join the GENE-N and GENE-Y entities under a common class "GENE". We train a 2-

level extraction system, the first level classifies the data into CHEMICAL or GENE and the second level classifies the data into GENE-N or GENE-Y. We pass the test data into the first model, if the label is GENE, we pass it into the second model to classify it into GENE-N or GENE-Y.

Findings and results

The following table shows the results of this experiment:

Table 10 Experiment 5 results

Token level eval	Precision	Recall	F1 Score
SciBERT (5.3)	66.20%	98.96%	79.33%
SciBERT CHEM-GENE classifier	78.40%	99.14%	87.56%
SciBERT GENE-N-GENE-Y classifier	56.38%	83.58%	67.34%
SciBERT hierarchical	66.28%	98.71%	79.31%
Entity level eval	Precision	Recall	F1 Score
SciBERT (5.3)	47.62%	77.34%	58.95%
SciBERT CHEM-GENE classifier	57.11%	80.36%	66.77%
SciBERT GENE-N-GENE-Y classifier	23.57%	81.28%	36.54%
SciBERT hierarchical	47.67%	77.14%	58.93%

Based on the results in Table 10, the hierarchical recognition system didn't improve the results however when observed the results of the level 1 classifier between CHEM and GENE, there is a significant improvement in scores verifying that the confusion is coming from GENE-N and GENE-Y classification, as they have a close semantic meaning and may come up in similar contexts which makes the distinction between them harder.

5.6 Token level, and entity level evaluation analysis

As observed from the above experiments there is drop in the scores between token level evaluation and named entity evaluation. This is expected as entity level evaluation doesn't give partial score to entities that were extracted correctly but treats all tokens in a named entity as one. You can view this in the example highlighted on Figure 8 below

Entity	Label	Token level Prediction	Entity level Prediction
terpenoids	CHEMICAL	B-CHEMICAL	GENE
estramustine phosphate	CHEMICAL	I-CHEMICAL	
methyltrienolone	CHEMICAL	I-CHEMICAL	
ter	B-CHEMICAL	B-CHEMICAL	O
##pen	I-CHEMICAL	I-GENE	
##oids	I-CHEMICAL	I-GENE	
estr	B-CHEMICAL	B-CHEMICAL	CHEMICAL
##amus	I-CHEMICAL	I-GENE	
##tin	I-CHEMICAL	O	
##e	I-CHEMICAL	O	
phosphate	I-CHEMICAL	O	
methyl	B-CHEMICAL	B-CHEMICAL	CHEMICAL
##tri	I-CHEMICAL	I-CHEMICAL	
##enol	I-CHEMICAL	I-CHEMICAL	
##one	I-CHEMICAL	O	

Figure 8 Evaluation Analysis

Taking the above tokens as an example, the token level analysis for those would be TP: 5, TN: 0, FP: 3 and FN: 4. Precision: 62.50%, Recall: 55.56%, F1: 58.82%. However, for entity level evaluation the first entity "terpenoids" out of its 3 tokens, only one token was predicted correctly so it was given a wrong label, same thing with second entity, the third entity is given the correct token although one of its tokens was not predicted correctly. For entity level evaluation we have TP: 1, TN: 0, FP: 1 and FN:1. Precision: 50.00%, Recall: 50.00%, F1: 50.00%.

Chapter 6

Conclusion and Future Work

Throughout this research, we have built a baseline system by initializing the model with BERT base model and finetuned it on the ChemProt dataset. Next, we experimented by initializing the model with SciBERT or BioBERT instead of BERT and we did a parameter tuning experiment to find the optimal model parameters. We also experimented with applying basic preprocessing rules and filtering out verbs and adjectives based on POS tagging. We also experimented with the DrugProt dataset to increase the training data size. Further experiments were conducted for Ensemble classifiers and hierarchical entity extraction models.

Based on the conducted experiments our findings are that initializing your system with SciBERT pre trained model and fine tuning it with ChemProt dataset has better results over other BERT variations. We also found out that applying preprocessing to the training data has a significant positive impact over model performance. Also filtering out verbs and adjectives by adding POS tagging to the preprocessing phase. With the release of DrugProt [5] dataset, a newer version of chemprot dataset we have the chance of increasing the training data which also have a positive impact over model performance. It is also worth noting that we have implemented two methods for evaluation, token level evaluation, entity level evaluation, the first one evaluates the model based on the prediction per token. The second method takes into account the exact entity and only counts a successful recognition if the whole entity was extracted and given the correct label. As expected scores from the second method are lower than that of the first method as they require higher certainty.

So, coming back to the research questions; to answer the first question, initializing the system with SciBERT model yields better results when it comes to extracting named entities from biomedical text. This can be observed throughout the experiments that SciBERT model has the best scores. To answer the second question preprocessing the data before passing it the model have a significant impact in improving model performance, this can be observed with the significant improvement with basic preprocessing and further improvement when POS tagging

is added to these rules. To answer the third question, increasing the size of training data using the DrugProt dataset improves model performance. To answer the fourth question, using ensemble classifiers didn't improve the overall system performance. To answer fifth question, building a hierarchical extraction model didn't improve model performance but identified that a significant reason for the drop in scores comes from the model's inability to distinguish between the different types of genes in the text.

To conclude, the best performance could be achieved by removing punctuation, stop words verbs and adjectives from the text and finetuning with SciBERT pretrained model with learning rate $3e-5$ for 3 epochs, token level evaluation: **Precision:** 66.20%, **Recall:** 98.96%, **F1:** 79.33%, entity level evaluation: **Precision:** 47.62%, **Recall:** 77.34%, **F1:** 58.95%. NCBI-Disease corpus [47] is one of the common benchmark dataset in the field of biomedical NER. F1 score for NCBI dataset with Spark NLP [48] model is 90.48% and with BioBERT model[3], F1 is 89.71%.

References

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," 2018, doi: 10.48550/ARXIV.1810.04805.
- [2] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A Pretrained Language Model for Scientific Text," *ArXiv190310676 Cs*, Sep. 2019, Accessed: Nov. 16, 2021. [Online]. Available: <http://arxiv.org/abs/1903.10676>
- [3] J. Lee *et al.*, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, p. btz682, Sep. 2019, doi: 10.1093/bioinformatics/btz682.
- [4] O. Taboureau *et al.*, "ChemProt: a disease chemical biology database," *Nucleic Acids Res.*, vol. 39, no. Database, pp. D367–D372, Jan. 2011, doi: 10.1093/nar/gkq906.
- [5] Krallinger, Martin, Rabal, Obdulia, Miranda-Escalada, Antonio, and Valencia, Alfonso, "DrugProt corpus: Biocreative VII Track 1 - Text mining drug and chemical-protein interactions." Zenodo, Jun. 29, 2021. doi: 10.5281/ZENODO.4955410.
- [6] R. Grishman, "Information extraction: Techniques and challenges," in *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*, vol. 1299, M. T. Pazienza, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 10–27. doi: 10.1007/3-540-63438-X_2.
- [7] R. Grishman and B. Sundheim, "Message Understanding Conference-6: a brief history," in *Proceedings of the 16th conference on Computational linguistics -*, Copenhagen, Denmark, 1996, vol. 1, p. 466. doi: 10.3115/992628.992709.
- [8] C. Nobata, S. Sekine, H. Satoshi, and R. Grishman, "Summarization system integrated with named entity tagging and IE pattern discovery.," *Eur. Lang. Resour. Assoc. ELRA*, vol. Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02), 30 2002, [Online]. Available: <https://nlp.cs..edu/pubs/papers/nova-lrec02.pdf>
- [9] G. Petasis, A. Cucchiarelli, P. Velardi, G. Paliouras, V. Karkaletsis, and C. D. Spyropoulos, "Automatic adaptation of proper noun dictionaries through cooperation of machine

- learning and probabilistic methods,” in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '00*, Athens, Greece, 2000, pp. 128–135. doi: 10.1145/345508.345563.
- [10] D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” *Linguisticae Investig.*, vol. 30, no. 1, pp. 3–26, Aug. 2007, doi: 10.1075/li.30.1.03nad.
- [11] O. Etzioni *et al.*, “Unsupervised named-entity extraction from the Web: An experimental study,” *Artif. Intell.*, vol. 165, no. 1, pp. 91–134, Jun. 2005, doi: 10.1016/j.artint.2005.03.001.
- [12] D. Hanisch, K. Fundel, H.-T. Mevissen, R. Zimmer, and J. Fluck, “ProMiner: rule-based protein and gene entity recognition,” *BMC Bioinformatics*, vol. 6, no. S1, p. S14, May 2005, doi: 10.1186/1471-2105-6-S1-S14.
- [13] 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.
- [14] J. Li, A. Sun, J. Han, and C. Li, “A Survey on Deep Learning for Named Entity Recognition.” arXiv, Mar. 18, 2020. Accessed: Aug. 14, 2022. [Online]. Available: <http://arxiv.org/abs/1812.09449>
- [15] V. Yadav and S. Bethard, “A Survey on Recent Advances in Named Entity Recognition from Deep Learning models.” arXiv, Oct. 24, 2019. Accessed: Aug. 15, 2022. [Online]. Available: <http://arxiv.org/abs/1910.11470>
- [16] L. F. Rau, “Extracting company names from text,” in [1991] *Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*, Miami Beach, FL, USA, 1991, vol. i, pp. 29–32. doi: 10.1109/CAIA.1991.120841.
- [17] S. Coates-Stephens, “The Analysis and Acquisition of Proper Names for the Understanding of Free Text,” *Comput. Humanit.*, vol. 26, no. 5–6, pp. 441–456, Dec. 1992, doi: 10.1007/BF00136985.
- [18] Fellbaum, Christiane, “Towards a representation of idioms in WordNet.,” *Usage WordNet Nat. Lang. Process. Syst.*, 1998.
- [19] 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.

- [20]S. K. Saha, S. Narayan, S. Sarkar, and P. Mitra, "A composite kernel for named entity recognition," *Pattern Recognit. Lett.*, vol. 31, no. 12, pp. 1591–1597, Sep. 2010, doi: 10.1016/j.patrec.2010.05.004.
- [21]Kazama J, Makino T, Ota Y, and Tsujii J., "Tuning support vector machines for biomedical named entity recognition.," *Proc. ACL Workshop Nat. Lang. Process. Biomed. Domain*, pp. 1–8, 2002.
- [22]M. Majumder, U. Barman, R. Prasad, K. Saurabh, and S. K. Saha, "A Novel Technique for Name Identification from Homeopathy Diagnosis Discussion Forum," *Procedia Technol.*, vol. 6, pp. 379–386, 2012, doi: 10.1016/j.protcy.2012.10.045.
- [23]Y. Wang *et al.*, "Supervised methods for symptom name recognition in free-text clinical records of traditional Chinese medicine: An empirical study," *J. Biomed. Inform.*, vol. 47, pp. 91–104, Feb. 2014, doi: 10.1016/j.jbi.2013.09.008.
- [24]Y. Zhang, H. Lin, Z. Yang, J. Wang, and Y. Sun, "Chemical–protein interaction extraction via contextualized word representations and multihead attention," *Database*, vol. 2019, p. baz054, Jan. 2019, doi: 10.1093/database/baz054.
- [25]Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [26]R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (almost) from Scratch." *arXiv*, Mar. 02, 2011. Accessed: Aug. 14, 2022. [Online]. Available: <http://arxiv.org/abs/1103.0398>
- [27]Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging." *arXiv*, Aug. 09, 2015. Accessed: Aug. 14, 2022. [Online]. Available: <http://arxiv.org/abs/1508.01991>
- [28]Z. Zeng, Y. Yao, Z. Liu, and M. Sun, "A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals," *Nat. Commun.*, vol. 13, no. 1, p. 862, Dec. 2022, doi: 10.1038/s41467-022-28494-3.
- [29]D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *J. Chem. Inf. Model.*, vol. 28, no. 1, pp. 31–36, Feb. 1988, doi: 10.1021/ci00057a005.
- [30]R. Sennrich, B. Haddow, and A. Birch, "Neural Machine Translation of Rare Words with

- Subword Units,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany, 2016, pp. 1715–1725. doi: 10.18653/v1/P16-1162.
- [31] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural Architectures for Named Entity Recognition,” 2016, doi: 10.48550/ARXIV.1603.01360.
- [32] 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.
- [33] Z. Chai, H. Jin, S. Shi, S. Zhan, L. Zhuo, and Y. Yang, “Hierarchical shared transfer learning for biomedical named entity recognition,” *BMC Bioinformatics*, vol. 23, no. 1, p. 8, doi: 10.1186/s12859-021-04551-4.
- [34] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “XLNet: Generalized Autoregressive Pretraining for Language Understanding.” arXiv, Jan. 02, 2020. Available: <http://arxiv.org/abs/1906.08237>
- [35] X. Wang *et al.*, “Cross-type biomedical named entity recognition with deep multi-task learning,” *Bioinformatics*, vol. 35, no. 10, pp. 1745–1752, May 2019, doi: 10.1093/bioinformatics/bty869.
- [36] B. V. Dasarathy and B. V. Sheela, “A composite classifier system design: Concepts and methodology,” *Proc. IEEE*, vol. 67, no. 5, pp. 708–713, 1979, doi: 10.1109/PROC.1979.11321.
- [37] G. Zhou, D. Shen, J. Zhang, J. Su, and S. Tan, “Recognition of protein/gene names from text using an ensemble of classifiers,” *BMC Bioinformatics*, vol. 6, no. S1, p. S7, May 2005, doi: 10.1186/1471-2105-6-S1-S7.
- [38] W. Ammar *et al.*, “Construction of the Literature Graph in Semantic Scholar,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, New Orleans - Louisiana, 2018, pp. 84–91. doi: 10.18653/v1/N18-3011.
- [39] L. A. Ramshaw and M. P. Marcus, “Text Chunking using Transformation-Based Learning,” 1995, doi: 10.48550/ARXIV.CMP-LG/9505040.
- [40] Bird, Steven, Edward Loper, and Ewan Klein, *Natural Language Processing with Python*. O’Reilly Media Inc., 2009.

- [41] A. Voutilainen, in *Part-of-Speech Tagging*, vol. 1, R. Mitkov, Ed. Oxford University Press, 2012. doi: 10.1093/oxfordhb/9780199276349.013.0011.
- [42] H Schmid, "Probabilistic Part-of-Speech Tagging Using Decision Trees," *Proc. Int. Conf. New Methods Lang. Process.*, vol. 12, pp. 44–49.
- [43] Y. Tian and D. Lo, "A comparative study on the effectiveness of part-of-speech tagging techniques on bug reports," in *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, Montreal, QC, Canada, Mar. 2015, pp. 570–574. doi: 10.1109/SANER.2015.7081879.
- [44] <https://github.com/charles9n/bert-sklearn>, "bert-sklearn." [Online]. Available: <https://github.com/charles9n/bert-sklearn>
- [45] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, 2020, pp. 38–45. doi: 10.18653/v1/2020.emnlp-demos.6.
- [46] "PyTorch." [Online]. Available: <https://pytorch.org/>
- [47] R. I. Doğan, R. Leaman, and Z. Lu, "NCBI disease corpus: A resource for disease name recognition and concept normalization," *J. Biomed. Inform.*, vol. 47, pp. 1–10, Feb. 2014, doi: 10.1016/j.jbi.2013.12.006.
- [48] V. Kocaman and D. Talby, "Biomedical Named Entity Recognition at Scale," 2020, doi: 10.48550/ARXIV.2011.06315.