Chemical Named Entity Recognition of Role Labelled Synthetic Chemical Procedures from Patents

Thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science in **Computational Natural Sciences** by Research*

by

Shubhangi Dutta
2018113004
shubhangi.dutta@research.iiit.ac.in

International Institute of Information Technology, Hyderabad
(Deemed to be University)
Hyderabad - 500 032, INDIA
March, 2024

International Institute of Information Technology

Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "**Chemical Named Entity Recognition of Role Labelled Synthetic Chemical Procedures from Patents**" by **Shubhangi Dutta**, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Dr. Prabhakar Bhimalapuram

To the family that I chose.

Acknowledgments

I am very grateful to my advisor, Dr. Prabhakar Bhimalapuram, for helping me throughout my research journey, and helped me to evaluate my work and solutions. I would also like to thank Dr. Manish Shrivastava, who guided me through my problem statement and my work at every step, and provided me with motivation to continue and explore through the ups and downs I faced during my work.

I would like to thank my aunts and uncles, Saibali Pal, Sanjukta Sarkar, Sarthak Pal, and Soumen Mitra, who supported me through this process, and provided me with love and positivity when I needed it most.

I am extremely grateful to my friends, Alok Kar, Siddharth Bhat, and Kavya Kolli, without whose support and encouragement this work would not have been possible. I also thank my friends, Tejaswini Anuhya Suma, Anshita Khandelwal, Aishwarya Shrivastava, Sanjana Sunil and Shine, who helped and encouraged me through various parts of my journey.

Abstract

Discovering new reaction pathways lies at the heart of drug discovery and chemical experimentation.

Chemical patent texts contain new reactions and reaction pathways. Thus, a huge amount of drug reaction data lies in unannotated patent texts which are not machine-readable. Reaction roles play an important part in analysing chemical pathways, and tracing chemicals through them, and while there is a vast body of chemical data available, the unavailability of reaction role annotated data is a blocker to effectively deploy deep learning methods for reaction discovery.

To overcome this hurdle, this work introduces a new dataset, WEAVE 2.0, an expansion of the existing WEAVE dataset, a chemical NER dataset obtained from chemical patents. WEAVE 2.0 augments WEAVE named entities along with full, manual, annotations of novel chemical reactions with reaction role information.

We provide baseline models for chemical entity recognition from our raw dataset, using simple architectures commonly used for chemical NER and related tasks, such as biomedical NER.

As the size of the dataset is small, we introduce dataset augmentation techniques to improve learning. These techniques can be used to generate further data from other patent-based datasets, such as WEAVE [17].

We also introduce and test improved models, which structure the problem as two smaller parts instead of one, both against the raw dataset, as well as data augmented using the above methods. Further, we compare our best models against other similar datasets for chemical NER, showing its performance across multiple similar tasks.

Our dataset and associated models form the foundation of neural understanding of chemical reaction pathways via reaction roles and will allow models trained for downstream tasks to utilise this information to generally lead to better predictions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We explain an overview of the problem of Chemical Named Entity Recognition, and the motivation behind it. We also briefly summarise our contributions.

Chemical discovery relies heavily on discovering new synthesis pathways. The search space of all possible syntheses is extremely high dimensional, and cannot be naively enumerated. Thus, to create novel synthesis pathways, we need methods to rapidly search through existing pathways, derive insights, and compare these.

Machine learning methods have proved highly effective in exploring and organising unstructured data in various fields, including vision, language, and physics. There has also been work done on producing similar datasets and models for chemistry. Prior work has focused on extracting data from research paper abstracts, patents, and medical records. Thus, we aim to utilise existing data, along with machine learning methods, to extract relevant chemical reaction data, and further aid the process of chemical synthesis and drug discovery.

Recognising chemical entities within texts is an important first step to analysing any chemical texts. However, this step can prove to be a challenging problem due to several reasons. These include:

- The complexity of chemical names, as they can be very long and complex, consisting of symbols, uncommon spellings, and abbreviations. Further, many chemical entities have similar but distinct names that may be difficult to distinguish.

- Chemical entities can have ambiguity, as many terms are not used to refer to the same entity: e.g. chlorine can be used for the element or the chemical disinfectant. Recognising these as separate entities requires an understanding of the context.

- Various chemical names may refer to the same chemical, such as by having common names, IUPAC names, formulas, etc.

- Chemical entities may require domain-specific knowledge to be identified. It may require a domain expert to understand and disambiguate the different formulas and chemical names.

While there is prior work that recognises chemical entities and their types (i.e. by naming conventions such as trivial, IUPAC etc.), in this thesis, we extend this by adding the critical information of *reaction role labels*, to chemical named entities.

This is a related task to semantic role labelling, where the focus is on understanding the meaning of a sentence by identifying the roles of the different elements within them. These roles can include agent, instrument, location, etc. Semantic role labelling involves two main steps: identifying the words or sentences that serve as the participants of the predicate, and then classifying each participant into its role.

Similarly, in chemical documents, these reaction role labels give important information. For this task, we must recognise the token as part of a chemical entity, and then classify it. We can use this to better understand the role of the chemicals in the whole process, by allowing the same chemical entity to be recognised in the different roles across reactions, as in the step-by-step processes described in patents, the product of one reaction is often the reactant in the next. Further, reaction roles give us important information about the environment and nature of each reaction, which may be valuable for learning the chemical pathway steps.

1.1 Contributions

The WEAVE dataset that was introduced in Nittala et al[17] is comprised of chemical reactions from patent data, annotated for chemical named entities. Our work builds upon this dataset and existing work done in chemical NER and chemical patent data extractions. Concretely, our contributions are listed as follows:

- We introduce a new corpus called the WEAVE 2.0 corpus (doi: 10.5281/zenodo.8386295), which utilises a randomly selected subsection of the WEAVE corpus, which contains these role labels *in addition to* the information present in the WEAVE data. The additional information present in the WEAVE 2.0 corpus may also be used to train models and further classify larger existing datasets that contain only type labels or only role labels.

- Furthermore, we experimentally verify that this information is useful by training *baseline* models, which are simple architectures using previously successful ideas in NER, to recognise the reaction role labels along with the chemical entities themselves.
- We also introduce *improved* models which formulate the problem of the two-part labels (type and role labels) in different ways, with various architectures. As each label has two parts, the type labels and the reaction role labels, they can be combined together, as in the baseline models, or trained together with two different classifier heads, as in the joint model, or as a two-step process where the classification is done by two separate models. This lets us explore which of the formulations represents the dataset best.
- Since we use Machine Learning models for this task, and the large number of labels in the dataset may lead to the data for each label being smaller, we also introduce data augmentation methods to increase the data size, and improve learning by the models. These methods may also be used to improve data quality for other similar tasks.
- We also extract and present, as part of the dataset, other relevant information that is contained in the tables of the EXAMPLE section, which is not annotated as part of the standoff annotation.

We aim to improve the existing resources in the domain of chemical NER, and provide a dataset, data augmentation methods, and model architectures to this end. These may allow better machine readability of chemical patent data, and improve and aid the process of machine-aided chemical data processing.

1.2 Thesis Organisation

In Chapter 2 we review the background for the chemical NER task, along with the survey of the existing literature. We discuss the necessity of this task, as well as the challenges in the field. We also survey existing literature for datasets and models that are applicable to this task.

Our contributing chapters are chapters 3-6. Chapter 3 introduces our new dataset, WEAVE 2.0 and how it helps to further the chemical NER task. Chapter 4 introduces our baseline models for the chemical NER task for the WEAVE 2.0 dataset, and we improve upon those architectures in Chapter 5. We discuss and analyse our results from our experiments in Chapter 6.

Chapter 2

Background and Related Work

We explore the relevant chemical concepts that are used in chemical named entity recognition, and the basic structure of a chemical patent, which forms our dataset to be built on, and to better understand the data we need to extract from them. Further, we also look at the natural language processing steps involved in extracting such chemical data from a patent.

2.1 Chemical Patents

A chemical patent is a document that is given to inventors or companies for novel chemical inventions. This means that the invention must be new, and not previously available or disclosed to the public. Further, patents are also non-obvious, which means that the inventions must have a step that is non-obvious to a person skilled in the relevant field. Patent inventions also have a practical or industrial utility.

Chemical patents usually contain the chemical compound or composition, its method of synthesis, and its applications, along with relevant experimental data and methods related to the compound's properties.

They contain the detailed steps of the chemical entity being produced, as a chemical synthesis pathway, which is the sequence of steps required to convert the reactants to the final products. It describes the step-by-step process for each chemical reaction in the sequence, and includes the intermediates formed as well.

In each step of a chemical synthesis pathway, specific reagents, catalysts and other reaction conditions such as temperature or pressure are applied to a specific reactant to achieve the desired reaction. Some steps of the pathways may also include purification or isolation steps.

These pathways are often used to create complex organic molecules, for example, drugs, polymers, and industrial chemicals, and are essential in these fields to develop new compounds, which are impossible or difficult to produce through natural processes, as well as to replicate existing compounds.

Chemical synthesis pathways are also often optimised for yield, cost, reaction sensitivity etc.

Synthesis pathways are often generated using a process called retrosynthetic analysis. This breaks down the target molecule into smaller precursor molecules, and uses this to plan the pathway by using a suitable starting route.

2.1.1 The EXAMPLE Section

Chemical patents contain a detailed process such that anyone with the relevant skills is able to replicate the results of the patent. Thus, they tend to contain an "*EXAMPLE*" section that details the steps of these processes. This section contains the chemicals (reactants, reagents, catalysts, and products), as well as the relevant identifying chemical tests, such as mass spectroscopy, NMR spectroscopy etc. This information is relevant to replicate and extrapolate chemical information from these reactions.

"*EXAMPLE*" sections comprise of steps to achieve the product or the reaction pathway described. They also contain, for each step, the chemical identifying tests for various products. Due to the step-by-step nature of the text, the product of one reaction may be the reactant of the next. However, all this information is in textual form, and the individual chemical entities must be extracted from the text to be machine-readable. A majority of what makes a patent novel is included in the EXAMPLE section, and while this information is available in the patents, it is not indexed in a format that would allow it to be used for training models.

2.2 Word Vector Representations

Word vectors or word embeddings, are numerical representations of tokens in a high-dimensional vector space, which allows them to be used in NLP models.

The underlying assumption in generating word vectors is that the words that appear in similar contexts have similar or related meanings. Hence, it is possible to extract these relationships using a numerical form, given large enough datasets.

Word vectors aim to capture both syntactic and semantic information from a corpus, by mapping them to a vector space. In this space, semantically similar words are positioned closer together.

Word vectors may be learned by analysing large corpora by using methods like Word2Vec, GloVe, and BERT, all of which use different algorithms to generate these vectors. Encoding the semantic data into a numerical form allows the information to be used by machine learning models, and convey the "meaning" of the words, allowing for better natural language understanding by the model.

2.2.1 GloVe

"Global Vectors for Word Representation" (GloVe) is an unsupervised, global word embedding algorithm that given a corpus, creates a mapping of the corpus vocabulary to n-dimensional vectors.

GloVe constructs word embeddings by using global co-occurrence data from a large corpus. It aims to capture the semantic and syntactic relationships between words by analysing their distributional patterns across the entire corpus. Unlike count-based approaches like word2vec, GloVe utilises a matrix factorisation technique to learn word representations.

The training process involves constructing a co-occurrence matrix, which represents the frequency of word pairs occurring together in the corpus. This matrix is then factorised to obtain word vectors that best represent the observed word co-occurrence patterns. The resulting word embeddings possess desirable properties such as vector arithmetic analogies, where relationships like "king - man + woman = queen" can be mathematically expressed. These vector embeddings encode semantic relationships in the vector space structure.

2.2.2 Transformers

Transformers are a type of deep learning model that were introduced by Vaswani et al[23]. Transformers have the following key components:

- Attention mechanism: Transformers use a mechanism called attention that allows the model to focus on relevant parts of the input sequence when processing each element. It assigns weights to different positions based on their relevance to each other, allowing the model to attend to important information and disregard irrelevant parts. Specifically, transformers use multi-head attention, which allows the model to simultaneously look at different parts of the input sequence.

- Encoder and decoder: Transformers also consist of an encoder and a decoder. The encoder processes the input sequence into an encoding, while the decoder generates the output from the encoding. Each encoder and decoder layer in the Transformer architecture consists of multiple

attention heads and feed-forward neural networks. The encoder and decoder layers are stacked on top of each other to form a deep network.

- Positional encoding: Since Transformers do not rely on recurrent connections like RNNs, they lack inherent information about the position or order of elements in a sequence. To address this, positional encoding is introduced to provide positional information to the model. Positional encodings are added to the input embeddings and carry information about the position of each element in the sequence.

2.2.2.1 BERT Embeddings

"Bidirectional Encoder Representations from Transformers" (BERT) is a natural language processing (NLP) model introduced by Google in 2018. It is a transformer-based model, which utilises self-attention mechanisms to capture relationships between words in a sentence or sequence.

2.3 Machine Learning Models

We explore the machine learning models and architectures that are commonly used in Named Entity Recognition. This includes Recurrent Neural Networks, and a type of RNNs called LSTMs. It also includes Conditional Random Fields, which are not a neural network, but a type of statistical model.

2.3.1 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks are a type of neural network which is used to process sequential data. They are commonly used for tasks where the sequence has a temporal nature, such as text-based analysis.

A key feature in RNNs is the ability to utilise information from previous *time steps* in the process. The RNN uses connections such that the output from a previous step is fed into the input of the current step. The hidden state of the RNN holds the information from the previous steps, and can be used to predict the next output. This hidden state is also updated with each step.

Traditional RNNs suffer from an issue called the vanishing gradient problem, where the effect of the earlier steps diminishes with each time step. to remedy this, variants such as LSTMs and Gated Recurrent Units (GRUs) have been introduced.

2.3.1.1 Long Short Term Memory Networks (LSTMs)

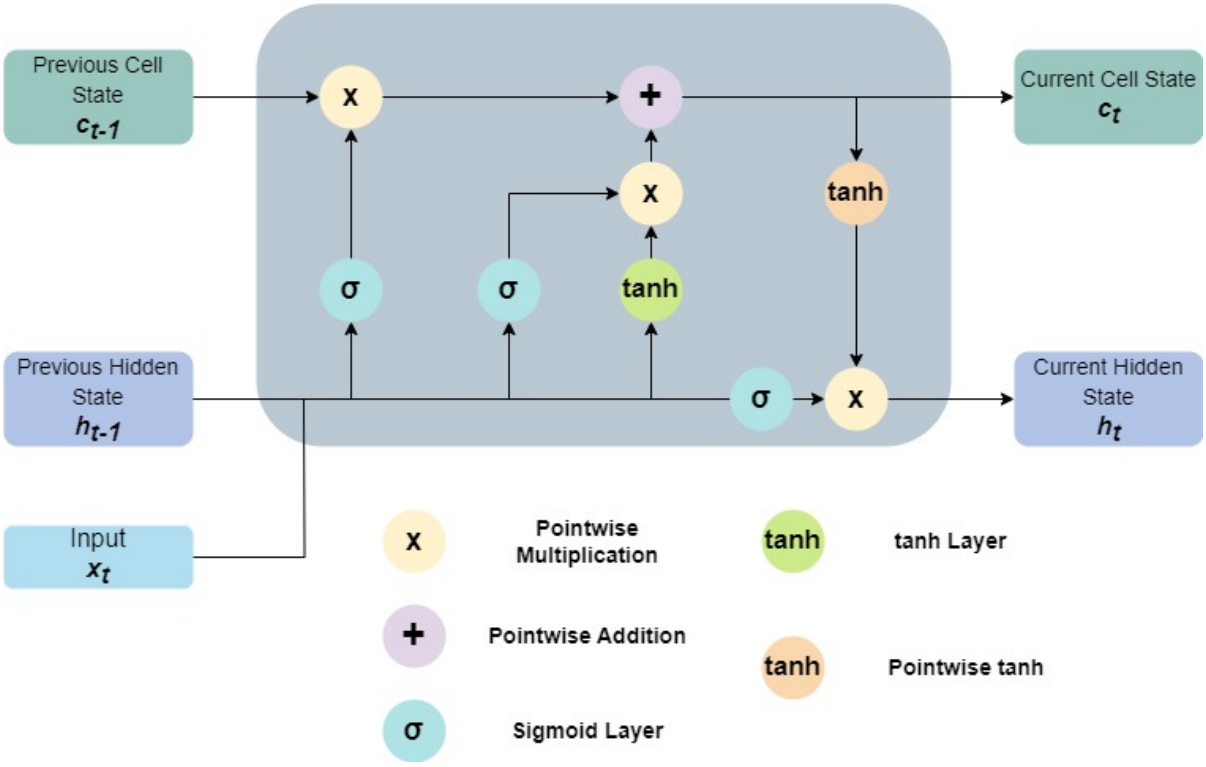

Figure 2.1: Architecture of baseline model

Long Short Memory Networks (LSTMs) are a type of RNNs that were designed to reduce the vanishing gradient problem. As the name suggests, LSTMs are able to retain information over both long and short memory intervals. This is done through the use of a memory cell, input gate, forget gate, and output gate.

The memory cell stores the internal memory state. the information flowing in and out of the memory cell is controlled by the input gate, forget gate, and output gate.

The input gate controls the new information coming into the memory cell. It uses a combination of the current state and produces a gating signal that controls the update of the memory cell's state.

The forget gate determines how much of the current information stored in the memory cell is to be forgotten in the next time step. It uses the state of the memory cell and the current input, and produces a gating signal that determines how much of the previous memory cell state should be retained.

The output gate determines the amount of information that is passed from the memory cell to the next hidden state. It uses the current input and the updated memory cell state and produces an output.

The combination of these gates allows the LSTM to retain information about long-term dependencies *and* short-term relevance. This makes the LSTM a good choice for tasks involving a temporal component like RNNs, while also mitigating the vanishing gradient.

2.3.2 Conditional Random Fields (CRFs)

Conditional Random Fields (CRFs) use an undirected graphical model to classify or predict labels for samples. CRFs are able to take into account context from neighbouring samples, where the neighbours are defined based on the way the graph is structured, and are therefore suited for tasks where the state of the neighbours is relevant to the state of the current prediction.

In a linear chain CRF, for example, the prediction only depends on the immediate neighbours, while in higher order CRFs, the graph may be connected to many nearby locations to ensure better predictions based on the neighbours.

2.4 Named Entity Recognition (NER)

Named Entity Recognition (NER) is a text classification task that is used as the first step for many other NLP tasks. We explore the background and prior work done in NER, and more specifically, on NER from chemical documents and related fields.

2.4.1 Sequence Labelling Tasks

Sequence labelling tasks are a class of machine learning tasks where an input sequence is given to the model and each token or element is assigned a label. This is usually used for text-based tasks.

Sequence labelling aims to capture the sequential information present in the data by assigning a label to every element. For example, this is commonly used in Part-of-Speech (POS) tagging, where a grammatical category (e.g. noun, adjective, verb etc) is assigned to each word in a sentence.

Named Entity Recognition (NER) is a sequence labelling task that involves the identification and classification of Named Entities (NEs) which are entities that are unique identifiers of interest, into predefined categories, such as names, places, etc. It is often the first step of other information extraction tasks, as it is able to extract these entities from unstructured text. This allows the models to understand and process the information more effectively.

2.4.2 Chemical Named Entity Recognition

Domain-specific NER, as compared to general NER, and in particular, chemical NER, is the task of recognising named entities that are specialised to a particular domain, in our case, chemical reactions. This is a more difficult task for machine learning, as there are fewer datasets available, and requires specialised architecture to ensure syntactically different but semantically similar entities (e.g. different chemical formulae) are recognised as different.

2.4.3 Datasets

For the task of Chemical NER, there exist datasets that extract chemical entities from various sources such as medical documents, abstracts, and patents. Some of the commonly used datasets for the chemical NER tasks are detailed in the next few sections.

2.4.3.1 CHEMDNER

The CHEMDNER [11] dataset was created as part of the BioCreative challenge, and is a widely used chemical NER dataset, often used to benchmark models. The dataset comprises of 10,000 PubMed abstracts, annotated with labels corresponding to the chemical entities in the text. The labels distinguish between the various types of chemical entities such as systematic IUPAC names, common chemical names, abbreviations, etc. It consists of a total of 84,355 manually annotated chemical entities, and 19,805 unique chemical names. This corpus is further elaborated on in 5.7

2.4.3.2 ChEMU

ChEMU [16], standing for Cheminformatics Elsevier Melbourne University, is an evaluation lab that presents two information extraction tasks. The corpus has 1500 chemical reaction snippets, including named entities and roles, from 170 patents.

The first task is a Chemical NER task that uses 10 different entity types, and requires the models to recognise them as chemical entities, as well as to differentiate between the different types. The entities are classified by their roles in the reaction (e.g. REACTION_PRODUCT, YIELD_PERCENTAGE, etc.), but not by their name type, using the BRAT annotation format. In some of the test data, there may be missing tags, as well as incorrectly identified tags, which the models are also required to fix.

The second task is an Event Extraction task that requires the model to recognise and extract the reaction steps that cause the reaction to progress into the next step. (e.g. "obtain" is a REACTION_STEP tag). These allow the "events" that link two steps or two related reactions to be recognised and hence allow better tracing of the synthesis pathway defined in the corpus.

2.4.3.3 NextMove

The NextMove Patent Reaction Dataset [14] is a large dataset, consisting of over 3 million reactions extracted from over 9 million patents and patent applications. It was improved upon to make it searchable and allow analytics of the reactions to be shown, creating an interface named Pistachio.

2.4.3.4 WEAVE

The WEAVE dataset, introduced in [17] is a large chemical NER dataset, extracted from chemical patents and manually annotated. In this work, we improve upon it and introduce the WEAVE 2.0 dataset, as detailed in 3.

2.4.4 Related Models

For NER tasks, specifically, chemical and biomedical NER a combination of BiLSTM and CRF is often used, as shown in [4],[15]. In chemical NER, ChemSpot [20] is an early model that uses this architecture, where a CRF along with a dictionary of brand and trivial names to identify NERs. WBI-NER [19] improves upon ChemSpot and makes it purely ML-based, removing the need for a dictionary. Later models include TmChem [13], which has 2 CRFs, both using different features and tokenisations and the model used by Zhang et al[26] which uses ChemSpot's output as a feature to generate word embeddings. Works like Zhang et al[26] also use unsupervised learning, such as clustering models using the ChemSpot dataset, along with rule-based post-processing, to achieve good results, showing that unsupervised methods can be used.

As BiomedicalNER is a related field, we survey some works, such as [6]. Commonly, we note that BiLSTM-CRFs are used in combination with other methods, on some common datasets such as ChEBI, BC-II, BC-V and NCBI[7],[11],[21], [9]. Cho et al [5] uses a combination of partial matching and strict matching to find the correct entity, which is a technique that could be adapted to chemical NER are

similar compounds may be found using partial matching, since chemical patents may contain compounds that are not available anywhere else.

Some works such as ChemTok [1] use novel tokenisation methods that allow better performance in named entity recognition tasks. They show that while regex may do well in extracting the compounds, their rule-based system can do better.

The initial ideas of using a CRF and another ML-based model are improved upon in many of these models, and thus this is used as a base for most of our models in this project, as they perform well for NER models. Most of our work uses a BiLSTM-CRF in combination with other methods.

2.5 Conclusion

We survey other work that has been done in chemical NER and related fields, showing that while there exist datasets that contain large amounts of information, some key features that may aid chemical NER models are missing in most datasets. Further, we analyse the existing models and build our work based on the methods shown by the existing state-of-the-art models, which include deep learning based LSTM-CRF architectures.

Chapter 3

The WEAVE 2.0 Dataset

The WEAVE 2.0 dataset, which can be found at 10.5281/zenodo.8386295 is a chemical NER dataset that is introduced and used for the experiments in this work. The dataset is based on a subset of the WEAVE dataset, which is also a chemical NER dataset, but does not include role labels unlike WEAVE 2.0.

The WEAVE dataset extracts information from the core of the patent, i.e. the Example section, as opposed to focusing on the introduction or abstract. The WEAVE dataset annotates the full reaction discourse of the Example section from the patents, unlike other similar datasets such as ChEMU[16]. This is to ensure that if there are entities referred to beyond the edges of the selected text, the model is able to obtain that context. Further, patents by definition are required to be complete, which makes them a great resource if analysed in their entirety. Hence, we augment a subset of the WEAVE dataset with role labels, in order to better understand the information extracted from it.

3.1 The WEAVE Dataset

The WEAVE dataset was introduced in Nittala et al[17], and it contains a total of 180 chemical patent documents, containing either A61K (Preparations for Medical, Dental, or Toilet purposes) or C07D (Heterocyclic compounds), from the United States Patent and Trademark Office (USPTO) in English. The texts are converted from an XML format, to have UTF-8 character encodings. The documents are annotated in the BRAT standoff annotation format [22] and classifies the named entities into seven labels: `ABBREVIATION`, `FAMILY`, `FORMULA`, `IDENTIFIER`, `MULTIPLE`, `SYSTEMATIC`, and `TRIVIAL`, based on the naming scheme used to identify the entity. Hydrochloric Acid.

ABBREVIATION refers to the shortened version of a longer chemical name, for example, THF for *Tetrahydrofuran*.

IDENTIFIER refers to chemical database identifiers, for example, CAS-RN IDs.

FORMULA refers to a chemical formula, such as HCl for *Hydrochloric Acid*.

MULTIPLE refers to occurrences that are not a continuous string of characters.

SYSTEMATIC refers to the IUPAC name of the chemical entity. This is a method of naming chemical compounds, usually organic compounds that is recommended by the International Union of Pure and Applied Chemistry that allows each compound to have a unique name, that can be used to generate an unambiguous chemical structure.

TRIVIAL refers to a commonly used name or a brand name that represents a chemical compound, for example, silica gel is used to refer to the form of silicon dioxide.

Methanesulphonyl chloride[SYSTEMATIC] is added dropwise (1 equiv.) to a solution of the corresponding ethylene glycol [SYSTEMATIC] (1 equiv.) and NEt3[FORMULA] (0.8 mol equiv.) in THF[ABBREVIATION] (100 mL) under an argon[SYSTEMATIC] atmosphere and at 0° C.

3.1.1 Replication of Results from WEAVE

The results reported on the WEAVE paper [17] were replicated. For the word vectors, 50-dimensional GloVe vectors were trained on a corpus of chemical patents from 2015-2020, downloaded from the United States Patent and Trademark Office (USPTO). The corpus was then evaluated using the model used by Yadav et al[25], against the BioCreative V corpus, which is comprised of the CHEMDNER patents, containing chemical and biological data. The accuracy obtained was 91%, which matches with that reported by the original paper.

3.2 Role Labels

While a reaction description in a scientific document might have entities belonging to the above-mentioned categories, it is important to note that these entities play a specific role within a verbose reaction description.

As can be imagined, there are some very common roles, such as REACTANT, REAGENT, PRODUCT, and SOLVENT. But in a reaction description (specifically in a patent document), a number of other

interesting categories might be considered important for understanding verbose reaction descriptions. These are reaction participant categories.

Aside from the reaction participants, chemical reactions require specific environments in order to take place. These can comprise of a CATALYST, inert gas environments (ENVINERT), pressure (ENVPRES), temperature (ENVTEMP), etc. These are reaction environment categories.

The details and chemical properties of the product of a reaction in a patent may be detailed with relevant entities, including YIELD, and the results of chemical tests such as NMR spectroscopy, MASS spectroscopy, and measurement of the EE or enantiomeric excess. These constitute the yield property categories.

A description of a process in a scientific text may also require references to different parts of the text itself, or to other texts, using discourse connectives. Patent data may therefore contain named STEP, METHOD, and EXAMPLE entities. These, as well as other chemical entities, may be referred to in other parts of the text using either a COREFERENCE for unnamed references (e.g. "above crude product") or a REFERENCETO a named product (e.g. REFERENCETO_STEP "1").

3.3 WEAVE 2.0

This work introduces the WEAVE 2.0 dataset, which contains 33 manually role-labelled documents that comprise a randomly selected subset of the documents from the WEAVE dataset, that contain an expanded tagset. WEAVE 2.0 adds role label tags that introduce reaction roles for each chemical, as well as introduces tags from the environment, yield properties, and discourse connective categories. Therefore, many labels have two parts, separated by an underscore, e.g. SYSTEMATIC_REACTANT. The first part of the labels refers to the "type of nomenclature" of the chemical name in the NER, and is similar to the labels introduced for the chemical entities in WEAVE. These labels are referred to as "type labels" through the text to distinguish them from role labels, which are the second parts of the labels, depicting the role of the chemical entity in the reactions.

Methanesulphonyl chloride$^{\text{SYSTEMATIC_}\textit{REACTANT}}$ is added dropwise (1 equiv.) to a solution of the corresponding ethylene glycol $^{\text{SYSTEMATIC\_}\textit{SOLVENT}}$ (1 equiv.) and NEt3$^{\text{FORMULA_}\textit{REACTANT}}$ (0.8 mol equiv.) in THF$^{\text{ABBREVIATION_}\textit{SOLVENT}}$ (100 mL) under an argon$^{\text{SYSTEMATIC_}\textit{ENVINERT}}$ atmosphere and at 0° C.

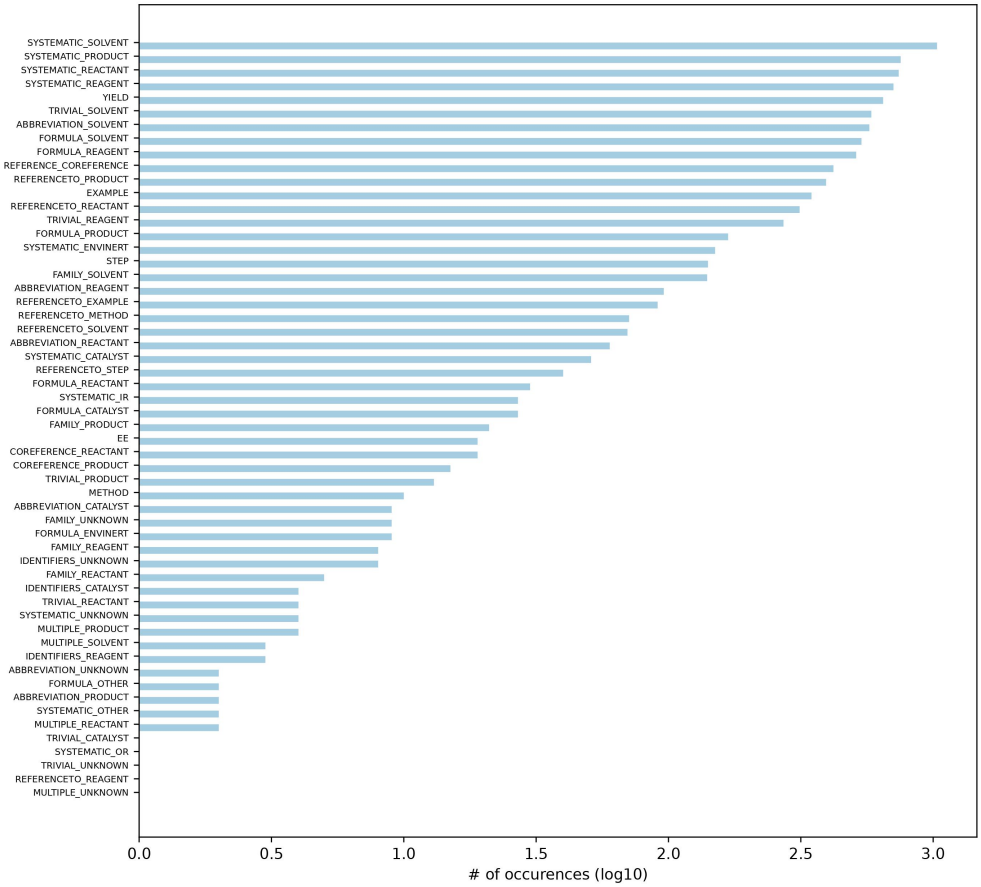

Figure 3.1: Distribution of the labels in WEAVE 2.0

The corpus, therefore, contains a much larger number of labels (71 labels), and a total of 17177 Named Entities, as compared to the WEAVE dataset which has 8 labels and 498807 Entities.

All of our experiments are conducted in the CONLL format, converted from the BRAT dataset using the BRAT toolkit provided[1].

[1] https://github.com/nlplab/brat

3.3.1 Data Statistics

The following tables show the distribution of the labels in the WEAVE 2.0 corpus. These are also depicted graphically.

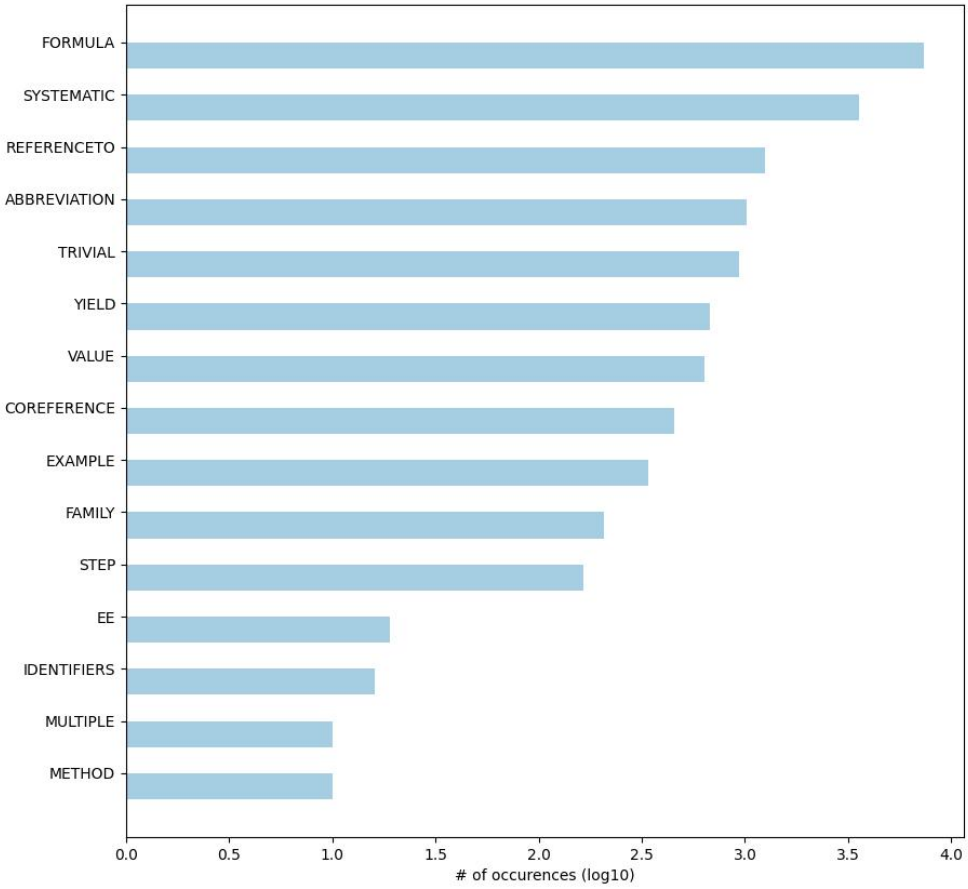

Figure 3.2: Distribution of the type labels in WEAVE 2.0

For our experiments, the dataset is tokenised using the CONLL tokenisation method for all experiments. The dataset contains IUPAC names of chemicals which are tokenised as multiple words by many standard tokenisers, including the CONLL tokeniser. The dataset is split into training and test corpora with a 70-30 split, and only the EXAMPLE section of the patent is used in training the models, as they contain

Label	Occurences	Label	Occurences
TRIVIAL_CHN	1	TRIVIAL_MASS	1
MULTIPLE_UNKNOWN	1	REFERENCETO_REAGENT	1
REFERENCETO_NMR	1	SYSTEMATIC_MASS	1
FAMILY_NMR	1	TRIVIAL_UNKNOWN	1
SYSTEMATIC_OR	1	MULTIPLE_REACTANT	2
TRIVIAL_CATALYST	2	ABBREVIATION_PRODUCT	2
ABBREVIATION_UNKNOWN	2	IDENTIFIERS_REAGENT	3
TRIVIAL_NMR	3	REFERENCETO_OTHER	3
MULTIPLE_SOLVENT	3	MULTIPLE_PRODUCT	4
SYSTEMATIC_UNKNOWN	4	TRIVIAL_REACTANT	4
IDENTIFIERS_CATALYST	4	FAMILY_REACTANT	5
FAMILY_OTHER	5	IDENTIFIERS_UNKNOWN	8
FAMILY_REAGENT	8	FORMULA_CHN	9
SYSTEMATIC_NMR	9	FAMILY_UNKNOWN	9
ABBREVIATION_CATALYST	9	ABBREVIATION_OTHER	9
METHOD	10	IDENTIFIERS_OTHER	12
TRIVIAL_PRODUCT	13	COREFERENCE_PRODUCT	15
TRIVIAL_OTHER	15	FORMULA_ENVINERT	17
EE	19	COREFERENCE_REACTANT	20
FAMILY_PRODUCT	21	FORMULA_CATALYST	27
SYSTEMATIC_IR	27	FORMULA_REACTANT	35
REFERENCETO_STEP	40	SYSTEMATIC_CATALYST	51
FORMULA_OTHER	52	ABBREVIATION_REACTANT	66
REFERENCETO_METHOD	71	SYSTEMATIC_OTHER	88
REFERENCETO_EXAMPLE	94	ABBREVIATION_REAGENT	100
SYSTEMATIC_ENVINERT	150	STEP	152
FAMILY_SOLVENT	167	FORMULA_PRODUCT	168
REFERENCETO_SOLVENT	250	ABBREVIATION_NMR	262
REFERENCETO_REACTANT	321	TRIVIAL_REAGENT	330
EXAMPLE	359	REFERENCETO_PRODUCT	407
COREFERENCE_COREFERENCE	420	FORMULA_REAGENT	569
ABBREVIATION_SOLVENT	604	FORMULA_MASS	608
TRIVIAL_SOLVENT	629	VALUE_MASS	636
FORMULA_SOLVENT	668	YIELD	686
SYSTEMATIC_REAGENT	787	SYSTEMATIC_REACTANT	816
SYSTEMATIC_PRODUCT	819	SYSTEMATIC_SOLVENT	1130
FORMULA_NMR	5330		

Table 3.1: Tabular distribution of the labels in the WEAVE2.0 corpus

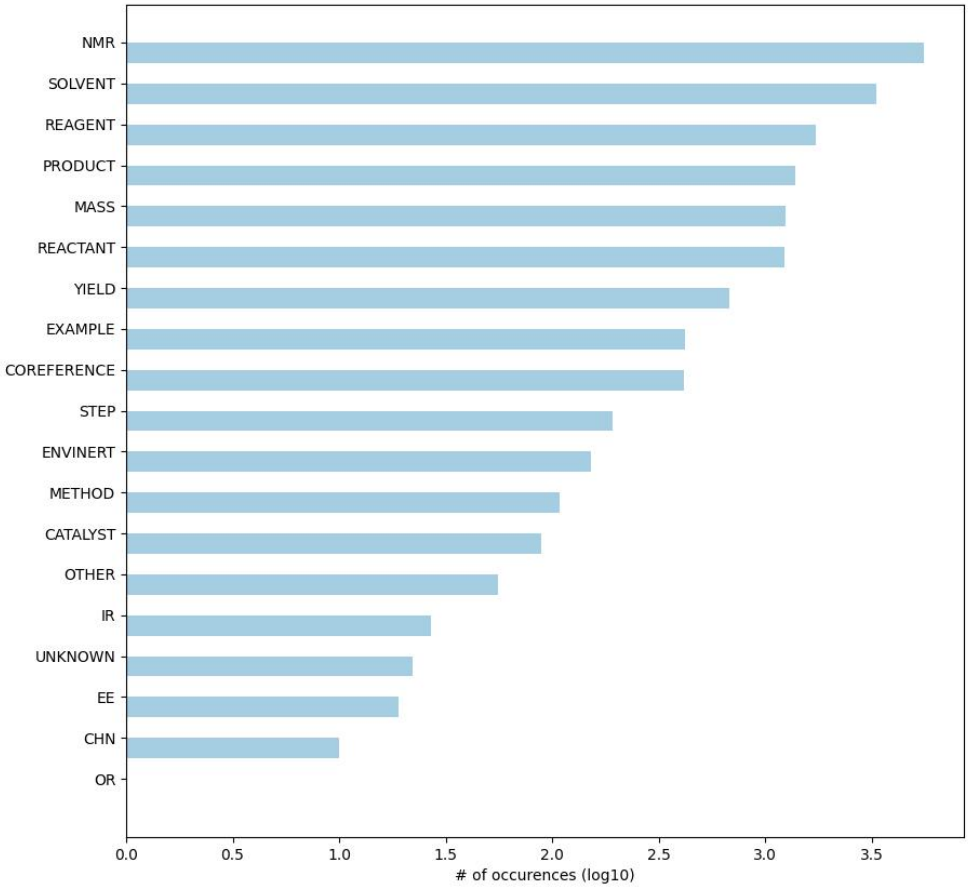

Figure 3.3: Distribution of the role labels in WEAVE 2.0

the highest density of the NERs. This reduces the number of labels to 68, and the total number of named entities to 15740.

3.3.2 Preprocessing Data

The data used in the original WEAVE paper was annotated by the type of naming convention used for each chemical compound. However, it did not contain the role-labelled named entities (i.e., labelling the named entities by their role in the reaction), which is required for later identifying the steps and storing the reaction data.

The raw role-labelled data as annotated by the expert annotators, required cleaning to adhere to the BRAT format. Further, as there were documents that were annotated by multiple annotators, it was also necessary to collate these, and remove duplicate annotations. Some of the documents did not contain any annotations, and were thus removed from the dataset, while other documents were annotated in parts, due to their length, and had to be collated and merged.

All of this was done to ensure that the completed dataset had a uniform format, and to remove any potential issues due to human error that may interfere with any machine learning predictions.

3.3.3 Table Extraction

Chemical recognition tests, as well as tables containing important data about the compounds, related to chemical tests or other relevant information, were not annotated as part of the role labelled data and hence were extracted using regex methods from the patent data and stored in a readable format for use in the downstream tasks. For example, the example table given in 3.2 shows compounds that are used for a transformation as part of a chemical process. However, these compounds were not annotated as part of the BRAT annotations in WEAVE 2.0. In order not to lose this information, these tables are extracted and stored.

R2a	R2b	R2c	R5
Cl	H	Cl	CH2CH3
Cl	H	Cl	CH2-i-Pr
Cl	H	Cl	CH2CH2Cl
Cl	H	Cl	CH2CH2OH
Cl	H	Cl	CH(Me)CH2OH
Cl	H	Cl	CH2CH(Me)OH
Cl	H	Cl	CH2C(Me)2OH
Cl	H	Cl	CH2CH2CH2OH
Cl	H	Cl	CH2C(Me)2CH2OH
Cl	H	Cl	CH2CH2CH(Me)OH
Cl	H	Cl	CH2C(=O)N(H)Et

Table 3.2: Example table taken from a patent in the WEAVE 2.0 corpus

Chapter 4

Baseline Models

We introduce our **Baseline Models**, which are trained on the WEAVE 2.0 dataset, and provide a benchmark for other models to compare against. These models are based on commonly used architectures in chemical NER and biomedical NER.

4.1 Model Architecture

The baseline models primarily consist of a word embedding that is input into another encoder, along with a decoder which acts as the classifier. Both models in this section treat each class as completely independent, and hence have a large number of labels due to the two-part nature of the labels.

4.1.1 Embeddings

Word embeddings are the inputs given to the model, and hence play a key role in the ability of the model to predict the correct labels. In a domain like chemical NER where there are many domain-specific entities, including chemical compounds that the model may not have seen before, domain-specific word embeddings are essential.

4.1.1.1 GloVe

For our task, GloVe word embeddings were trained on a chemical patent corpus taken from the United States Patent and Trademark Office (USPTO), with patents from 2016 to 2019. About 230,000,000 lines of patent data were used with a symmetric window size of 15, which means that for each word, the co-occurrence matrix 'looks' at words up to a distance of 15 away from the centre word. Each GloVe vector was of 100 dimensions.

4.1.1.2 Fine-Tuned BERT Embeddings

For our baseline, a pre-trained BERT-uncased model [8] was fine-tuned on patent data from USPTO, with patents from 2016 to 2019, using a total of 300,000 lines of patent data. This was done using Masked Language Modelling (MLM), which masks about 15% of the words in the input, with the model having to predict the masked words. This fine-tuning step ensures domain-specific learning for the embeddings themselves, leading to better representations as well as better recognition of chemistry-related words, as opposed to using the general BERT model.

In a separate experiment, the GloVe embeddings and the BERT embeddings were concatenated and used as a sentence embedding as input to the classifier, in order to collate the information both embeddings provide.

4.1.2 BiLSTM-CRF

As discussed in 2.3.1.1, LSTMs are well-suited to a sequential task like NER. However, in a standard LSTM, information is always processed in a forward direction, i.e. each element in a sequence uses information learned from previous elements. This does not allow for information from the succeeding elements to be used.

This problem is solved with the introduction of a Bidirectional LSTM (BiLSTM), which comprises of two LSTMs. One of the LSTMs processes information in the forward direction, while the other processes the information in the reverse direction. This gives BiLSTMs the key feature of being able to use information from both past and future elements, and retain long-term dependencies as well.

Based on the good performance of BiLSTM-CRF architectures (e.g. [4], [6], [15]) for NER tasks in general, as well as the usage of a BiLSTM-CRF model for the WEAVE [17] corpus baseline, a BiLSTM-CRF model was used for the baseline models for the WEAVE 2.0 corpus.

We use the combination of the two models, the BiLSTM and CRF, as following a BiLSTM with a simple classifier makes each prediction conditionally independent. However, a CRF is able to take context into account while decoding.

Hence, the BiLSTM was used as the encoder, with 50 hidden states. The output from the BiLSTM was then sent to a CRF layer which classifies the labels. Hyperparameter tuning was done on the learning rate, number of epochs and the number of hidden layers, with the best performance being at 25 epochs.

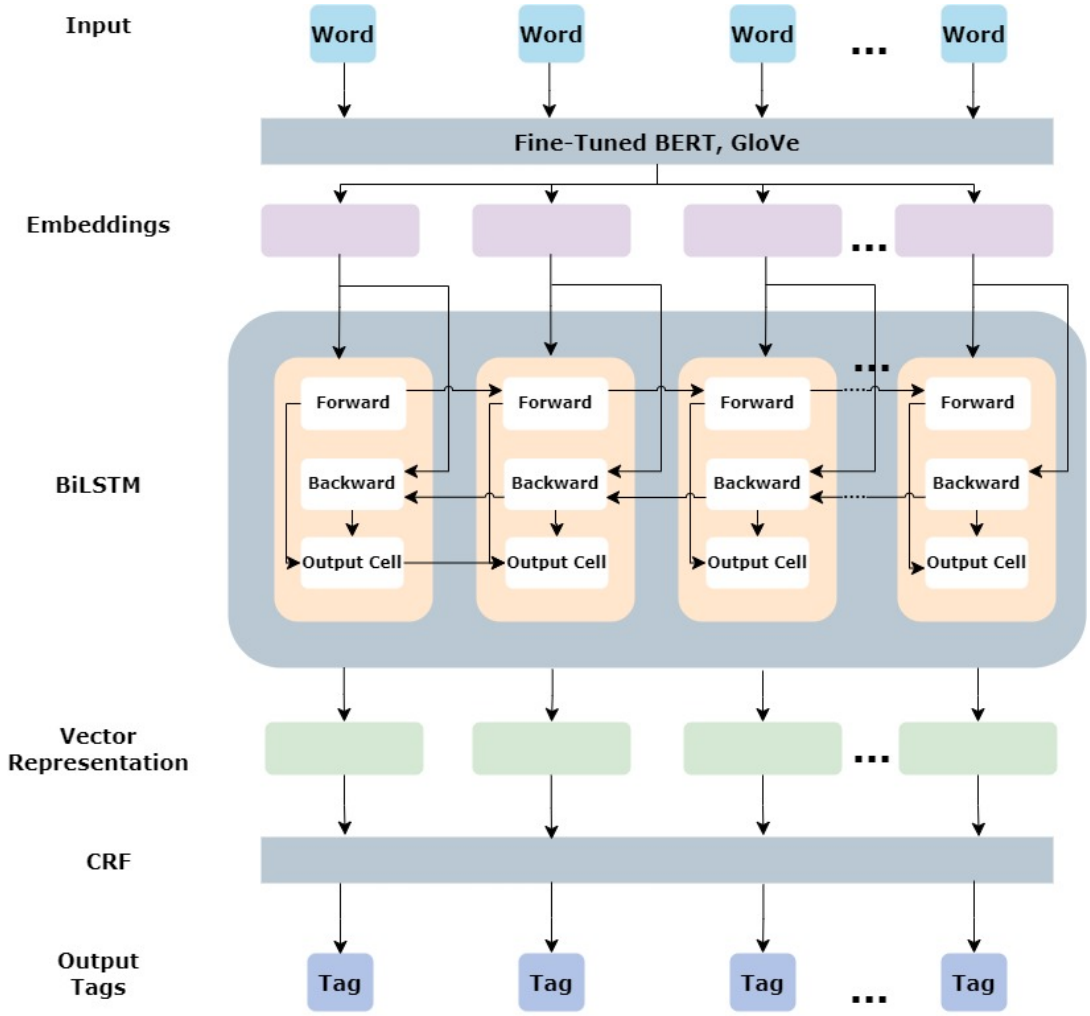

Figure 4.1: Architecture of baseline BiLSTM model

4.1.3 BERT + Fully Connected Layer

The base BERT model is trained on a large corpus of unlabelled text from the internet, enabling it to learn general language representations. It then undergoes fine-tuning on specific downstream tasks, such as question answering, text classification, named entity recognition, and more. This two-step process helps BERT transfer its acquired knowledge to various NLP applications.

By utilising transformer architectures and a large-scale pre-training approach, BERT has achieved state-of-the-art results on several benchmark NLP tasks and has become a foundational model for many subsequent advancements in the field. Its ability to capture contextualised word representations has made it widely used in both research and industry settings for a range of natural language understanding tasks.

Our model consists of BERT embeddings with a fully connected layer to act as the classifier. The fine-tuned BERT embeddings are used as an input to a single fully connected layer to create a classifier, in place of the BiLSTM-CRF model, for all the architectures detailed above. In a fully connected layer, each neuron applies a linear transformation to the input, in this case, the output from BERT. This allows every element from the input vector to influence every output element generated by the fully connected layer. However, these models generally performed poorly, showing that only the information obtained through the BERT embeddings is not enough to generate an accurate prediction.

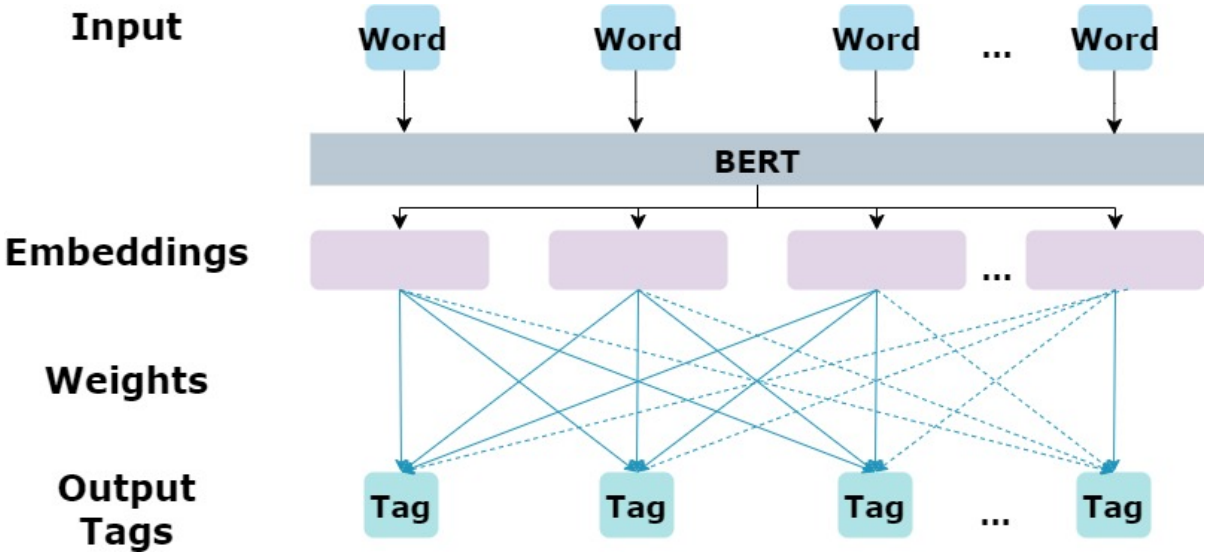

Figure 4.2: Architecture of baseline BERT+Fully Connected Layer model

Chapter 5

Further Experiments

All the models that are introduced in this section are based on the baseline model. Each one consists of an embedding that is the input to the model, one or multiple encoder layers, and one or multiple decoder layers.

5.1 Improvements to Embeddings

We use pre-trained BERT embeddings[1]. These ChemicalBERT embeddings are trained using chemical texts, including chemical Wikipedia articles, starting from the SciBERT [2] checkpoint, which is trained on more general scientific text on a large scale. While there do exist other similar models such as ChemBerta[3], they are trained for a specific task, and thus we choose the *chemical-bert-uncased* model. They perform significantly better than the BERT embeddings which were fine-tuned locally by us on chemical patent data which was introduced in 4.1.1.2. These embeddings are referred to as ChemicalBERT in this text.

5.2 Joint Models

For this set of models, we break the task down to two parts: i.e. the name labels and the type labels. However, these two parts are trained simultaneously, using a single hidden layer. This means that each task is also able to use the information learned by the model from the other task. Both the name and type label assignments depend on the information learned by the model *jointly*.

[1] https://huggingface.co/recobo/chemical-bert-uncased

5.2.1 Model Architecture

This model consists of a BiLSTM-CRF based model, with a joint hidden layer, with different classifier heads for the type labels and role labels. The model is given one input, and it predicts two outputs, classifying each word into the type of chemical NER name it has (e.g. `SYSTEMATIC`), and the role label (e.g. `REACTANT`). During training, the loss for both the outputs is back-propagated into the same hidden layer, as well as to the ChemicalBERT model. A joint hidden layer allows for better feature representations in the latent space that contains information from both the type and role labels.

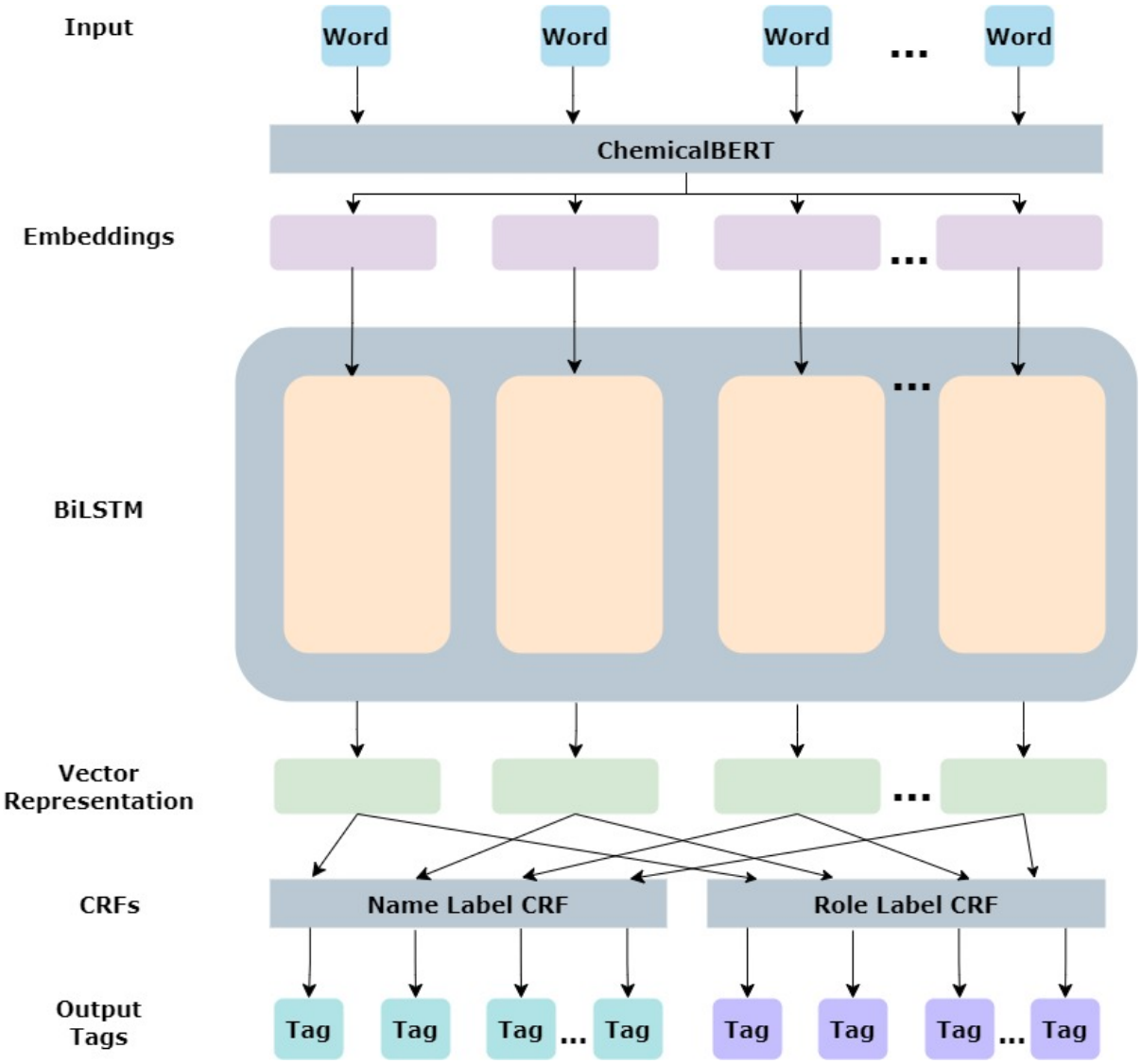

Figure 5.1: Architecture of ChemicalBERT + Joint Model

5.3 Two Step Models

The task is again broken down into two. However, in this case, the two tasks are treated sequentially and have different hidden layers. As the type labels are predicted first, they are not dependent on the training of the role labels. The role labels are trained later, and given the extra information, that is, the type labels, to improve its accuracy.

This two-step formulation of the task allows the large number of labels to be reduced, without losing the amount of information. Further, since there are more and larger datasets available for the type labels, (e.g. WEAVE, CHEMDNER), this allows the first step to potentially be trained on a larger dataset and leverage this information to better predict the role labels.

5.3.1 Model Architecture

This model consists of a two-step process. Each of the individual models used is a BiLSTM-CRF, similar to the baseline model. Each of the training data labels are split into two labels for this model, creating separate lists for the type and role labels. In the case of the labels that did not have 2 parts, e.g. `YIELD`, the same tag was repeated in both the sections.

For the first step, the model was trained on the WEAVE 2.0 corpus for the type label, and the model is not sent the role labels. ChemicalBERT generates embeddings for the BiLSTM, and the model predicts the type label associated with each word. This model is trained for a given number of epochs.

The predictions from the first step are then fed into the second model, along with the sentence embeddings from ChemicalBERT. This model then predicts the role labels.

During training, while the second model is trained, the loss is back-propagated into both models, as well as the ChemicalBERT model.

5.4 Attention-Based Models

An attention model aims to preserve the context of each element in a sequence input to it, by assigning it a *weight* that is relative to the other words. This allows the model to not lose the context even if the input sequence is very large. Thus we use an attention-based model as one of the improved experiments for our chemical NER task.

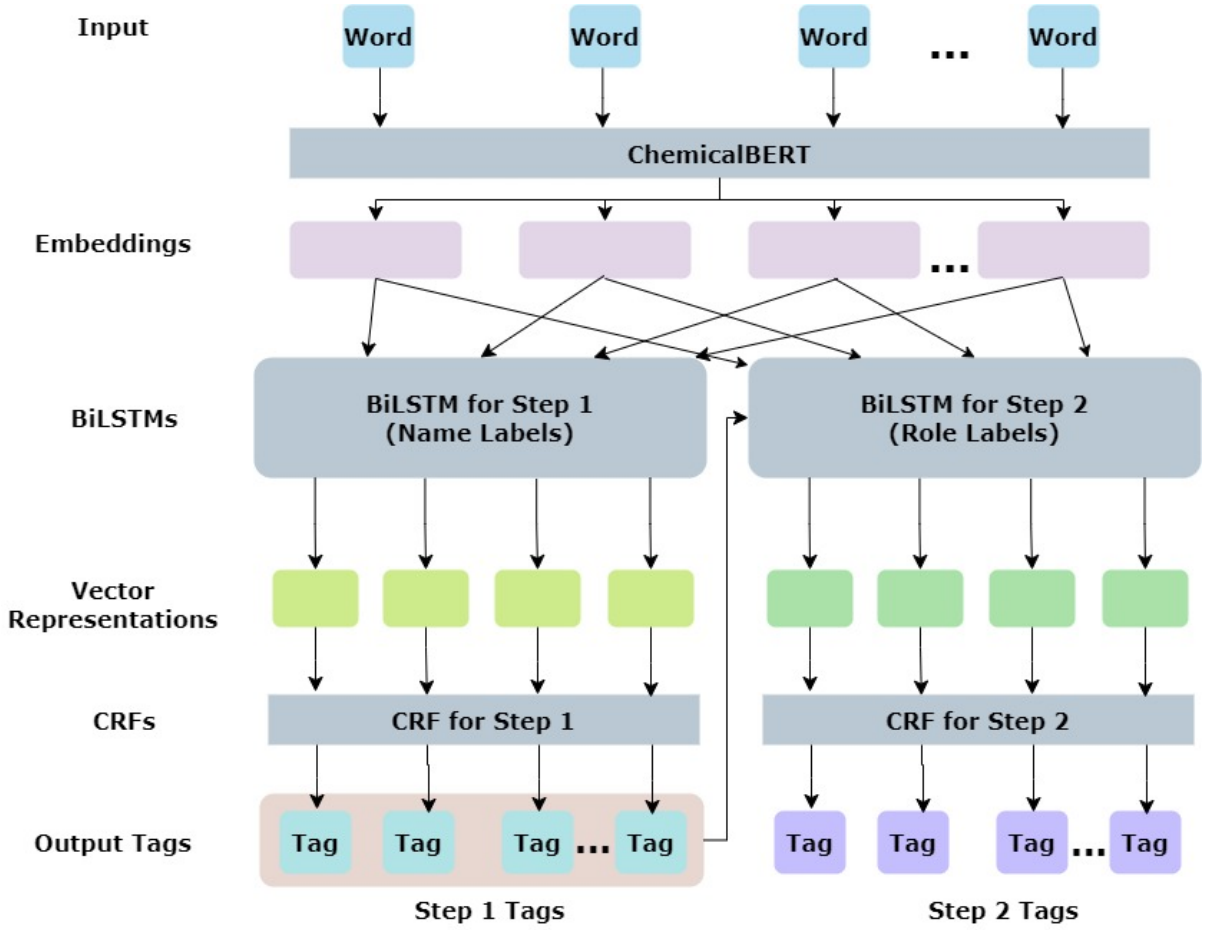

Figure 5.2: Architecture of ChemicalBERT + 2Step Model

5.4.1 Model Architecture

In the attention-based model, in place of the CRF layer, the BiLSTM layer is followed by an attention layer, a fully connected layer, an attention layer, and a final fully connected layer, which acts as the classifier. The same ChemicalBERT embeddings are used for the input. This model performs well generally, which shows that the attention and fully connected layers are a good decoder for the BiLSTM encodings.

5.5 Class-Balanced Loss Models

As the class imbalance in our data is very high, a notable feature from initial experiments was that the labels which had a high number of occurrences (for example, the O label), were learned better by the

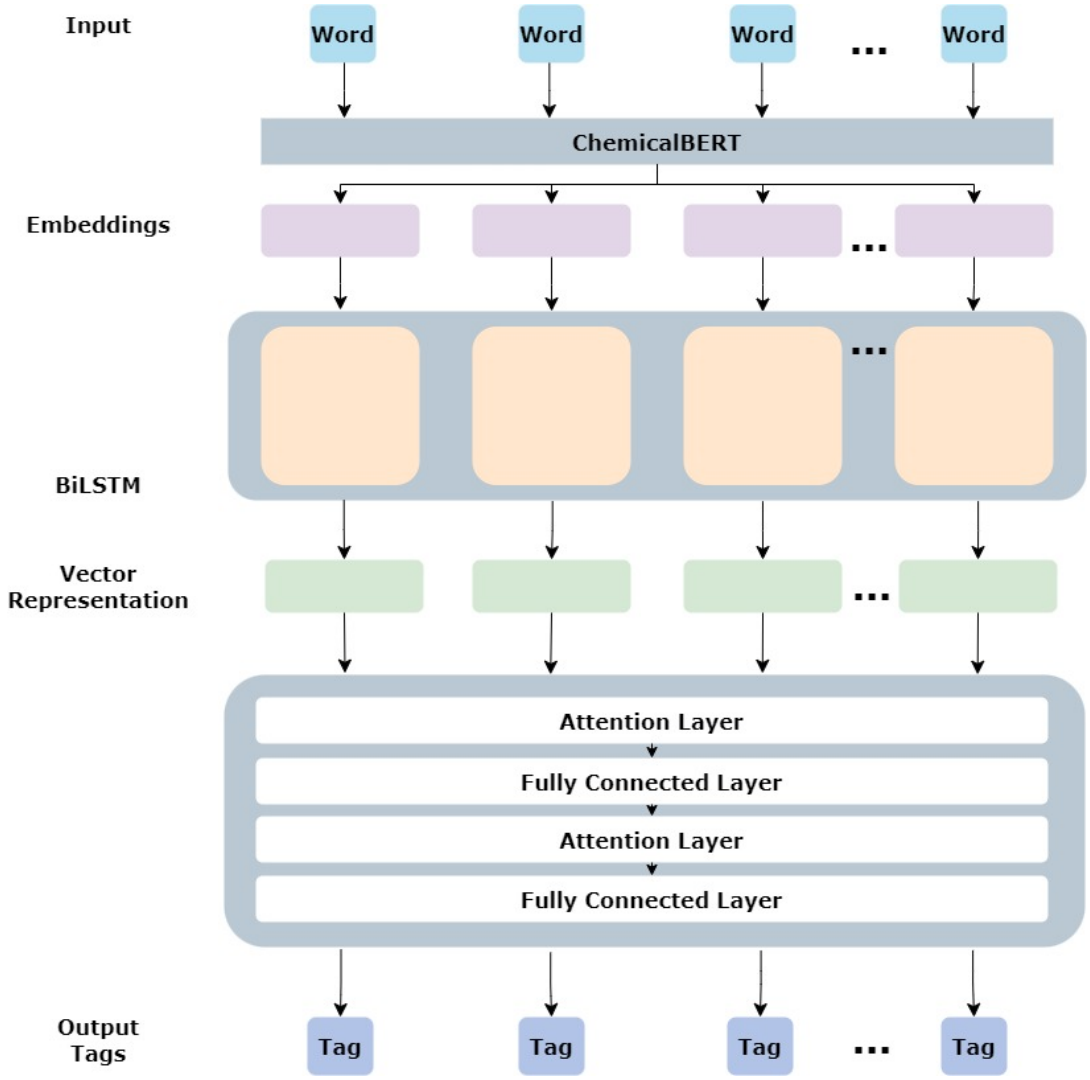

Figure 5.3: Architecture of ChemicalBERT + Attention Model

models than the less frequent labels. However, training for more epochs leads to the model overfitting on these frequent labels.

In order to resolve this, we experiment with Loss-balanced models. These models include a custom loss function, where the loss is divided by the number of occurrences of the label, i.e. scaled down according to the frequency of the label. This allows us to potentially reduce the chances of overfitting and improve the model learning.

5.5.1 Model Architecture

The BERT+Fully Connected Layer model introduced in 4.1.3, and the Attention-based Model explained in 5.4, had the custom loss function introduced, while the rest of the architecture remained the same. The loss was back-propagated through the model as before, with the only change being the scaling factor.

5.6 Dataset Augmentation

The dataset is imbalanced, as it contains a large number of classes but a smaller number of labels. Further, due to the nature of the chemical patents, the number of labels in each class has a large disparity. This can be seen in 3.1, which shows the high class imbalance. While treating the problem as two steps does reduce the number of classes, the number is still high, and it may benefit from data augmentation.

Since all the models used here are based on neural networks, increasing the quantity and quality of the data leads to better learning by the models. To improve the results in the previous experiments, we augment the dataset to correct the large class imbalance. This is done in the following ways:

5.6.1 Shuffling Sentences

The inputs to all the improved models are sentence embeddings from the BERT models. Therefore, the sentences of the whole training text can be shuffled, while keeping the order of words in each sentence the same, which keeps each sentence embedding the same. The resulting corpus is appended to the existing corpus. This does cause the issue of the result text not making sense as a whole, however, the corpus size and therefore the number of each of the labels is increased.

5.6.2 Replacing with Random Strings

Following the process in Task 2 of BioCreative VII, in Erdengasileng et al[10], a list of randomly generated strings of length 3-10 characters is created. Some of the Named Entities in the corpus are then probabilistically replaced with one of these strings. The goal is to ensure that the model is able to classify entities it has not seen before in any training data. This method is used in conjunction with the shuffling method described previously. The text generated in this way is appended to the existing corpus to generate the augmented corpus.

5.6.3 Replacing Named Entities

In this method, some of the Named Entities are randomly replaced by other similar Named Entities (e.g. `SYSTEMATIC_REACTANT` may be replaced with `SYSTEMATIC_REAGENT` entities, but not with `TRIVIAL_REAGENT` entities). This leads to sentences that make sense in English, but do not make chemical sense. However, since the goal is to make sure that the model is able to correctly recognise the type and role of the NER based on its position in the sentences, as well as the general structure of the token(s), we are able to augment the dataset using this method. This method is used in conjunction with the shuffling method described previously. This method is the most useful in reducing the class imbalance issue, as the classes with a lower number of labels can have a higher number of instances when they are added into the new text. The text generated in this way is appended to the existing corpus to generate the augmented corpus.

All the corpora generated by augmentation methods are tried against the improved models detailed in 5.

5.7 The BioCreative V Task: The CHEMDNER Dataset

The CHEMDNER dataset is a collection of scientific articles specifically curated for the CHEMDNER challenge. It consists of a large corpus of biomedical literature that has been annotated with various types of chemical and biomedical entities.

The dataset includes articles from various sources, such as PubMed, which cover a wide range of topics in the field of chemistry and biomedicine.

Each article in the CHEMDNER dataset is manually annotated by domain experts. The annotations typically include chemical compound names, enzymes, genes, proteins, and other relevant terms. The dataset contains a total of 7 tags: `ABBREVIATION`, `IDENTIFIER`, `MULTIPLE`, `FORMULA`, `SYSTEMATIC`, `TRIVIAL`, `FAMILY`.

5.7.1 Experiments using the CHEMDNER dataset

We use this dataset to test our Attention-based Model, as introduced in 5.4. As this model performs well using the WEAVE 2.0 dataset, we aim to establish that it is possible to use this model for similar tasks. The CHEMDNER dataset is not augmented in any way for this task. As the CHEMDNER dataset

only includes 7 tags similar to the "type" tags from WEAVE 2.0, this is not a role labelling task, and thus only involves identifying and classifying the entities into type of naming convention.

The hyperparameters are tuned to achieve the best results on this dataset, however, all the remaining model architecture is retained from 5.4.

Chapter 6

Results and Analysis

We report our results for the WEAVE dataset during the replication of the WEAVE results, and also for the first step of our Two Step Process in 5.3. We then study the performance of our WEAVE 2.0 dataset by first using the unchanged dataset as a baseline, and then augmenting the dataset by (a) shuffling sentences 6.3.2, (b) replacing NERs with random strings 6.3.3, and (c) replacing NERs with semantically similar NERs 6.3.4. We demonstrate that this augmentation is beneficial, with the final dataset providing the best performance. Further, we also test the accuracy of our best model using the CHEMDNER dataset, and achieve a high F1 score.

6.1 WEAVE Results

Our Two Step Model (5.3) contains the first step of the model as just a type-labelling task, as a result of which we were able to train it on WEAVE, which is larger than WEAVE 2.0. In Table 6.1, we report the results of the (replicated) baseline of the original WEAVE paper, as well as the first step of our model, which outperforms it.

Model Name	**Precision**	**Recall**	**F1**
Replication using original model from [25]	**0.935**	0.89	0.91
ChemicalBERT+2Step BiLSTM-CRF Step 1	0.93	**0.94**	**0.93**

Table 6.1: Models using WEAVE corpus

Model Name	Precision	Recall	F1
GloVe+BiLSTM-CRF	0.80	0.80	0.79
BERT+BiLSTM-CRF	**0.81**	0.83	**0.80**
GloVe+BERT+BiLSTM-CRF	0.72	**0.85**	0.78
BERT+Fully-connected Layer	0.74	0.69	0.71

Table 6.2: Baseline Models using WEAVE 2.0 corpus

6.2 Baseline Models

We first analyse the results from the baseline models with various values for the hyperparameters. In order to examine the effects of the data augmentation techniques, we also include the results from the experiments with the augmented datasets.

6.3 Improved Models

In this section, we look at the improved models, and compare them to the baseline models. We also provide possible reasons for the changes in performance between the baseline and improved models.

6.3.1 Without Data Augmentation

The first experiments were conducted without using any data augmentation, using only the EXAMPLE section of the dataset.

These include the baseline models tabulated in Table 6.2. The baseline BiLSTM-CRF had an 0.80 F1 score, and 0.81 precision score, when used with the fine-tuned BERT embeddings. The combination of GloVe and BERT embeddings with the same model architecture has the highest recall score of 0.85. The ChemicalBERT+Fully-connected Layer models generally performed poorly. While transformer architecture is state-of-the-art in many domains, the BERT model in this case is unable to learn a representation that is able to differentiate the different named entities, without a decoder that is able to capture more information.

Model Name	Precision	Recall	F1
ChemicalBERT+Joint BiLSTM-CRF	0.79	0.83	0.79
ChemicalBERT+2Step BiLSTM-CRF	**0.83**	0.79	0.80
ChemicalBERT+Attention Model	0.82	**0.85**	**0.82**

Table 6.3: Improved Models using WEAVE 2.0 corpus

Model Name	Precision	Recall	F1
ChemicalBERT+2Step BiLSTM-CRF Step 1	0.93	0.94	0.93
ChemicalBERT+2Step BiLSTM-CRF Step 2	0.92	0.92	0.92

Table 6.4: Two Step model stepwise

Most of the improved models, as tabulated in Table 6.3, performed better than the baseline model, and the best recall and F1 scores are by the Attention-Based Model, with the 2-Step Model having the best precision score. We expect this, as the improved models contain better sentence embeddings, as well as architecture that may treat the role and type labels separately, reducing the complexity of each task.

We also present the results of each step of the two step model, independent of the other step, i.e. the results of the first step were tabulated separately, and the model in step 2 was given true values instead of the step 1 output. These results are shown in 6.4. We see that individually, each step of the model performs better than the whole model. We expect this, since the error across the steps will multiply for the final accuracy since error in either step counts as an error in the final result.

6.3.2 With Data Augmentation: Adding shuffling of sentences

The augmented data generated by sentence shuffling (as explained in 5.6.1) is tested on the improved models, as well as the BERT+Fully-connected Layer model. The performance of the BERT+FC model is lower than the other models. The best results are achieved by the Attention Model, however, we note that in general the results of all models improve with this data augmentation technique. This shows us that the data augmentation technique leads to the models learning better or forming better latent

Model Name	Precision	Recall	F1
BERT+Fully-connected Layer	0.71	0.67	0.69
ChemicalBERT+Joint BiLSTM-CRF	0.84	0.87	0.84
ChemicalBERT+2Step BiLSTM-CRF	0.82	0.85	0.80
ChemicalBERT+Attention Model	**0.85**	**0.88**	**0.86**

Table 6.5: Models using shuffled sentences

Model Name	Precision	Recall	F1
BERT+Fully connected Layer	0.71	0.67	0.69
ChemicalBERT+Joint BiLSTM-CRF	0.84	0.87	0.84
ChemicalBERT+2Step BiLSTM-CRF	0.80	0.84	0.80
ChemicalBERT+Attention Model	**0.84**	**0.87**	**0.86**

Table 6.6: Models using shuffled sentences and replacing words with random strings

representations for the data. We also expect this, as while shuffling the sentences will not change the sentence embeddings, changing the order of sentences will have an effect in the training of the BiLSTM.

6.3.3 With Data Augmentation: Adding shuffling of sentences and replacing words with random strings

The improved models and the BERT+FC model are all tested against the augmented data that is generated by the process detailed in 5.6.2 The best results are again achieved using the Attention-based Model, however, this form of augmentation appears to decrease the performance, compared to simply shuffling the sentences. This may be due to the random words being incorrectly classified, as these strings are not similar to either the chemical NEs or the tokens that are not NEs. These words may appear to hence cause the model to learn incorrect labels and lower performance.

Model Name	Precision	Recall	F1
BERT+Fully connected Layer	0.71	0.67	0.68
ChemicalBERT+Joint BiLSTM-CRF	0.84	0.87	0.86
ChemicalBERT+2Step BiLSTM-CRF	0.82	0.85	0.80
ChemicalBERT+Attention Model	**0.85**	**0.88**	**0.87**

Table 6.7: Models using shuffled sentences and replacing NERs with similar NERs

6.3.4 With Data Augmentation: Adding shuffling of sentences and replacing NERs with other NERs of similar types

An augmented dataset is generated by the process described in 5.6.3, and then tested against all the improved models and the BERT+FC Model. The results are tabulated in Table 6.7. The Attention-based Model performs the best, however, all the model results show that this augmentation method produces the best results overall, as the F1 score increases for all the models.

This may be due to the fact that the replacements are made with similar entities, which then allows the model to see the same type labels in different role label contexts, making it learn a better representation of the entities.

6.3.5 Performance on CHEMDNER

The CHEMDNER corpus is a widely used corpus in chemical NER. It was also used by the WEAVE [17] dataset as a comparison. however, due to a lack of role labels, and therefore also having a smaller number of labels, it does not have the same task description as WEAVE 2.0.

We show that our Attention-Based Architecture performs well across all the datasets. When trained and tested on the CHEMDNER dataset it achieves 95% precision, 96%recall and a 95% F1 score.

This shows us that the models that we introduced can also be used in comparable and similar tasks, and performs well in them.

This model also therefore provides a comparison for the same architecture being trained on the two datasets for the chemical NER task. As it performs worse in the WEAVE 2.0 dataset, we believe that the role labels make it a more challenging task in that case.

6.3.6 Comparative Bar Graphs and Label-wise Performance

We also analyse the labelwise performance of our models. This is to see which of the labels contribute to the performance to the models. For every model, we notice that much of the contribution comes from the 'O' tags which are the tags with the highest occurrences. However, these are also the tokens that are not named entities. We do notice that the results are not purely from the 'O' tags. Also we notice that tags which had a lower number of occurrences often had a lower accuracy. We expect this, as the model requires a lot of data to learn.

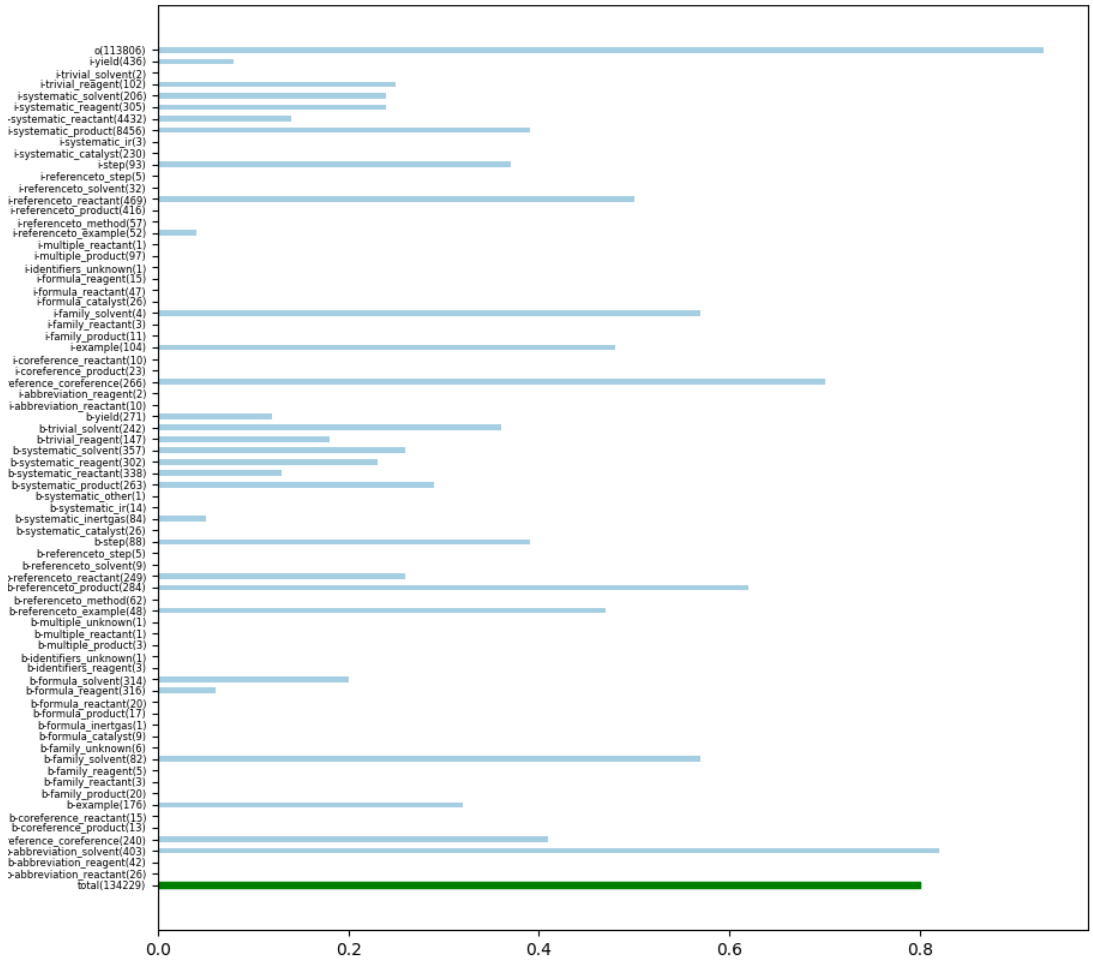

Figure 6.1: Label-wise Performance of Glove with LSTM model

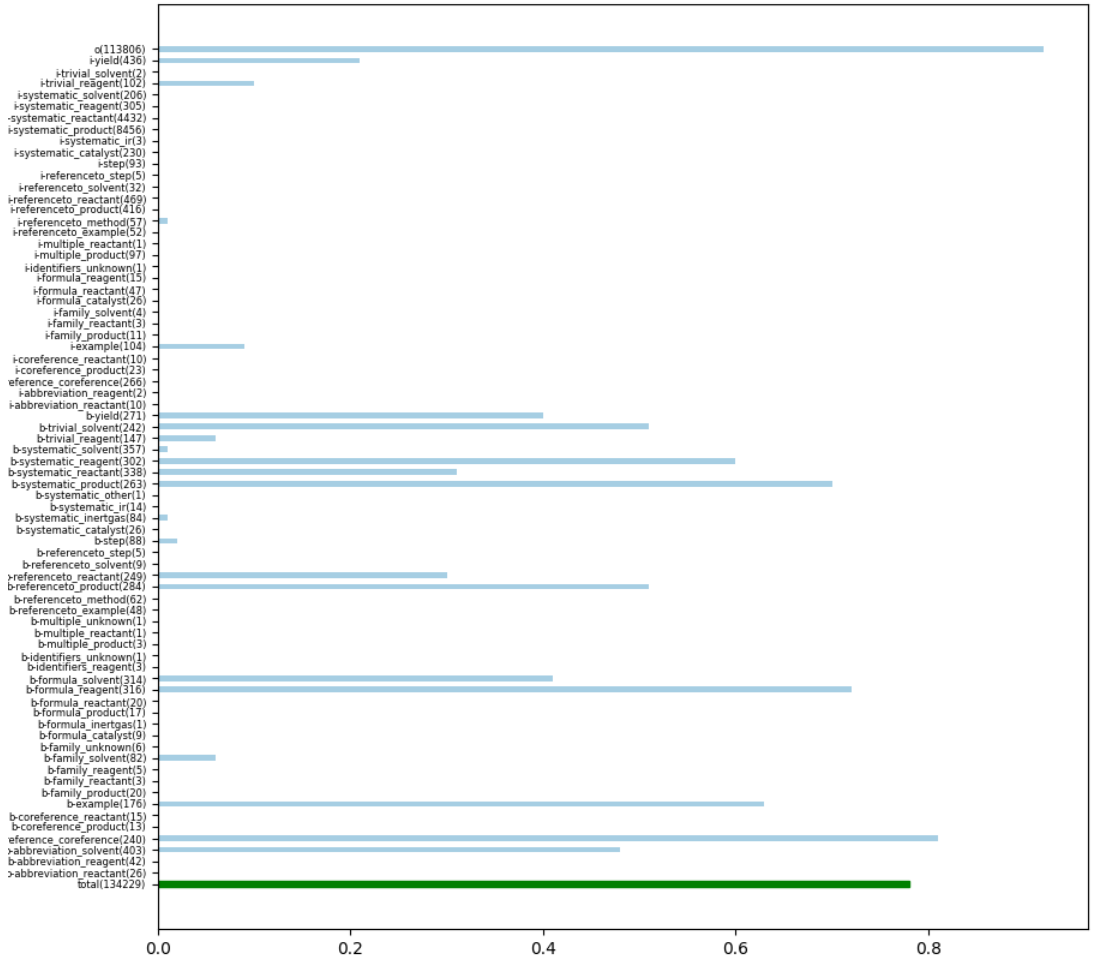

Figure 6.2: Label-wise Performance of Bert and Glove with LSTM model

Figure 6.3: Label-wise Performance of Bert + Fully Connected model

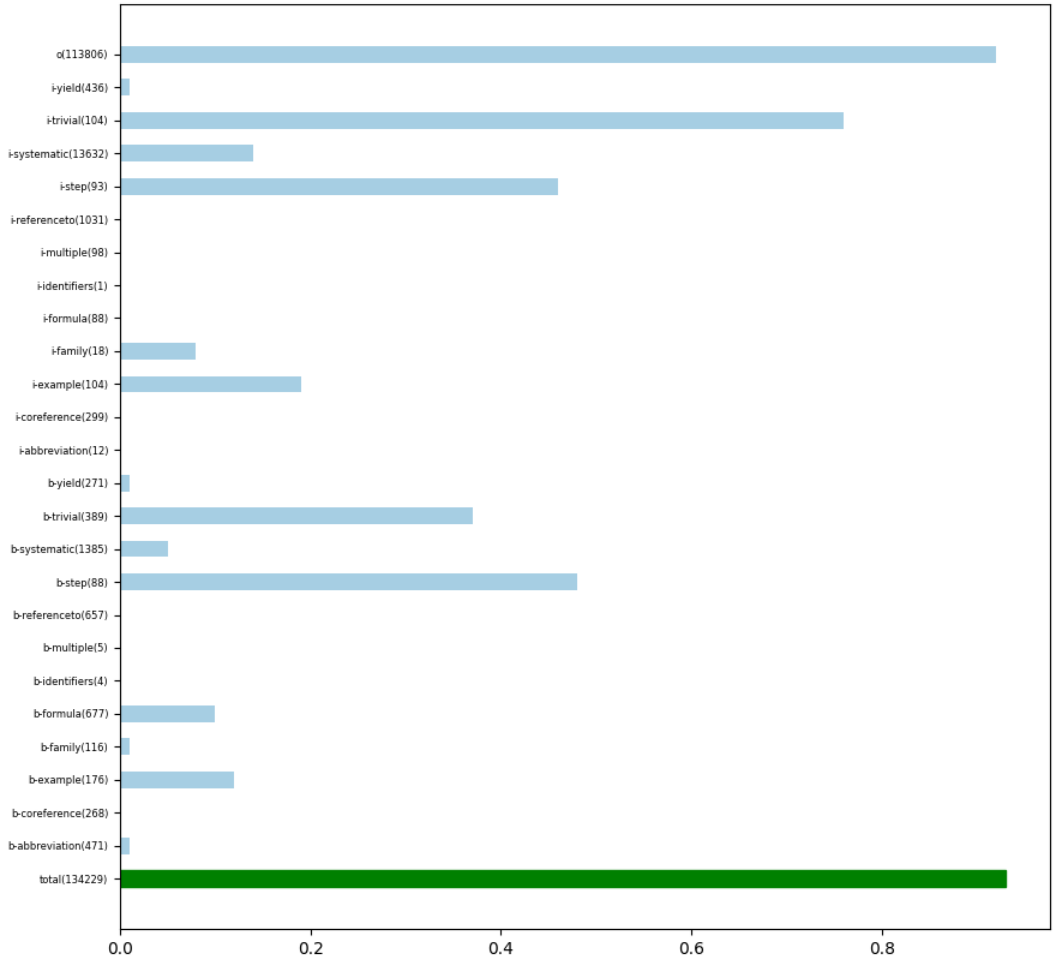

Figure 6.4: Label-wise Performance of Step 1 of 2 step model

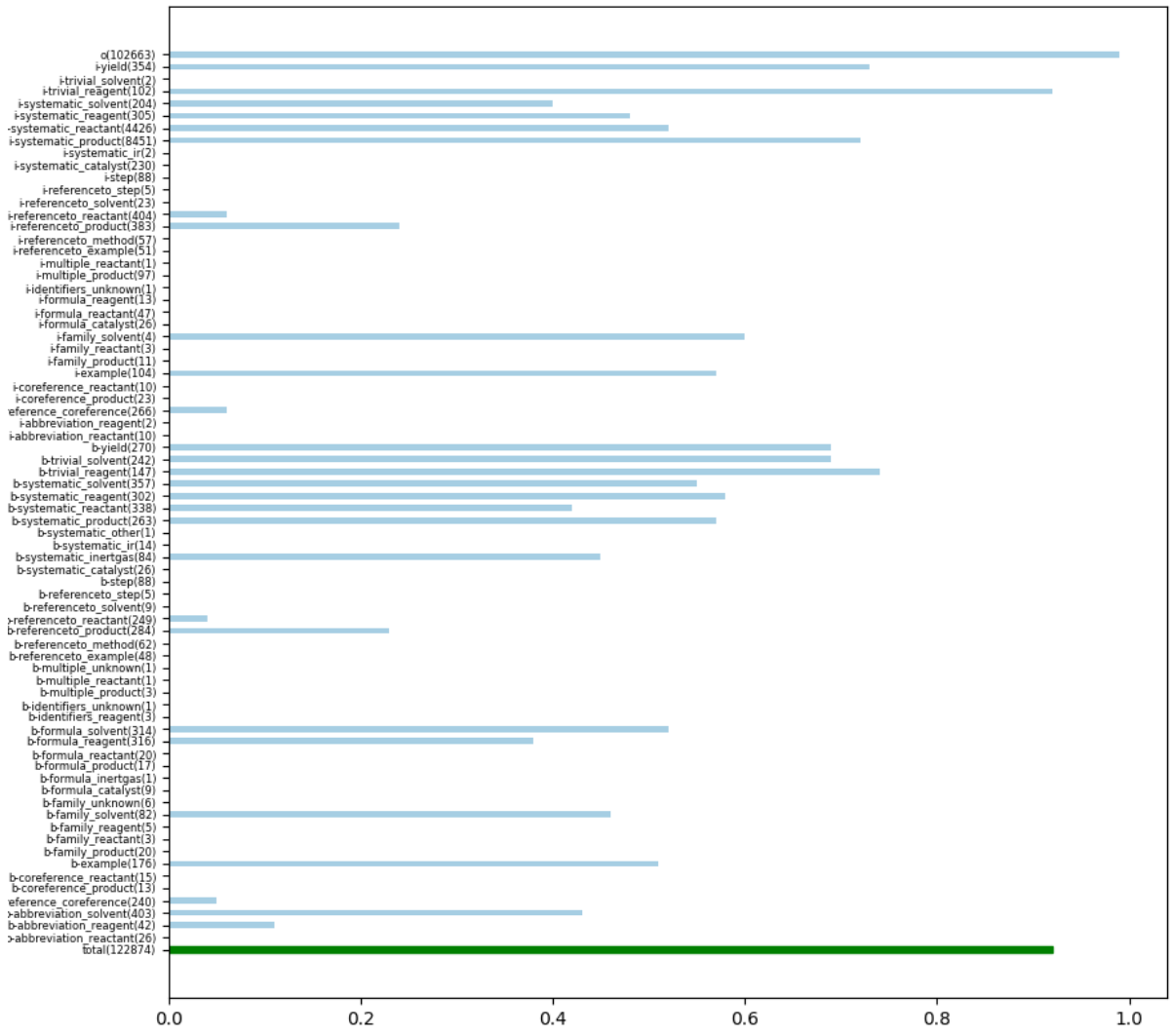

Figure 6.5: Label-wise Performance of Step 2 of 2 step model

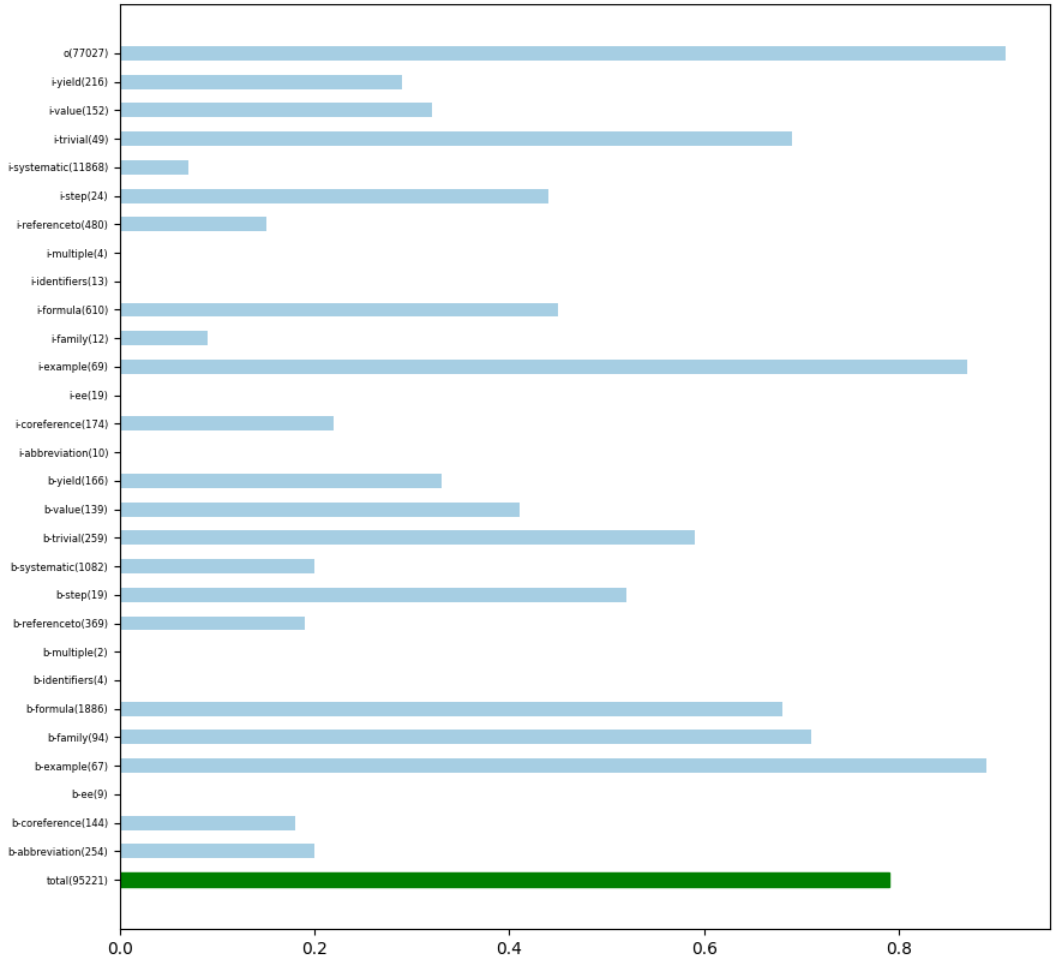

Figure 6.6: Label-wise Performance of Side 1 of Joint model

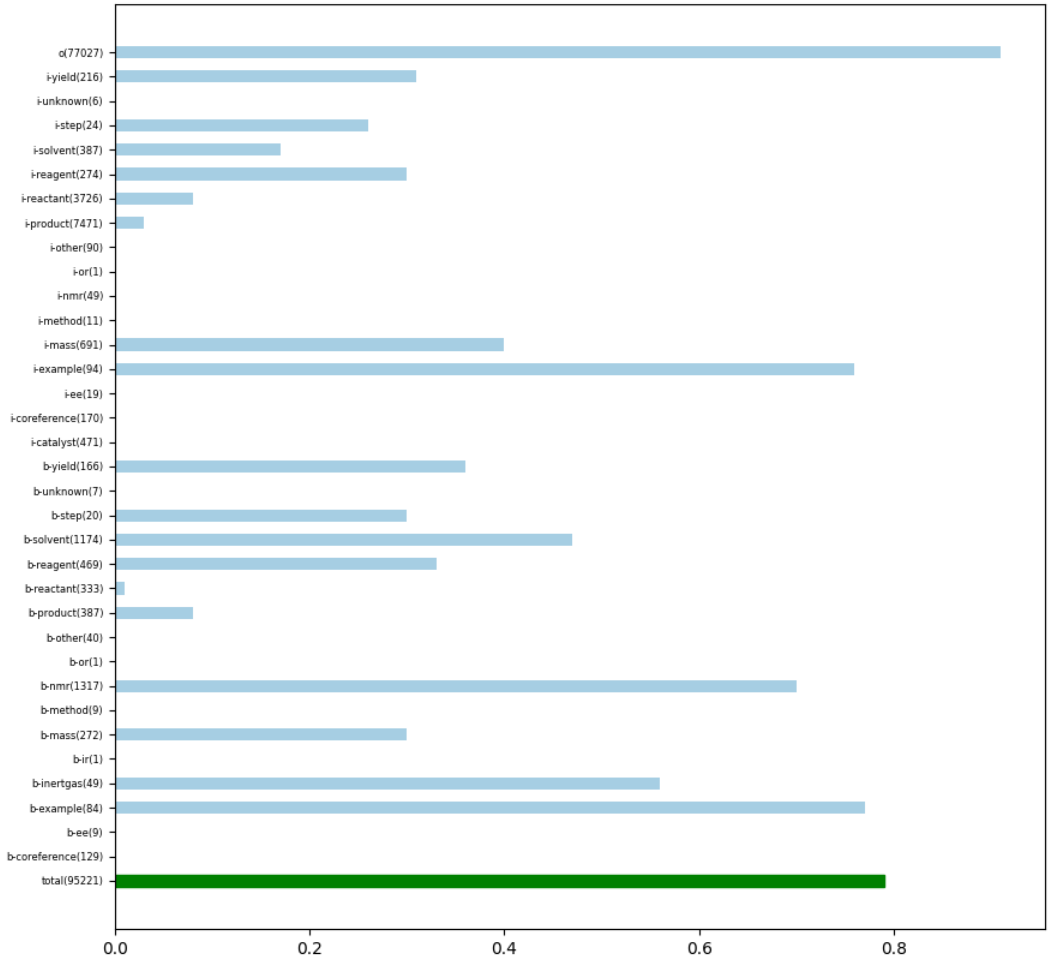

Figure 6.7: Label-wise Performance of Side 2 of Joint model

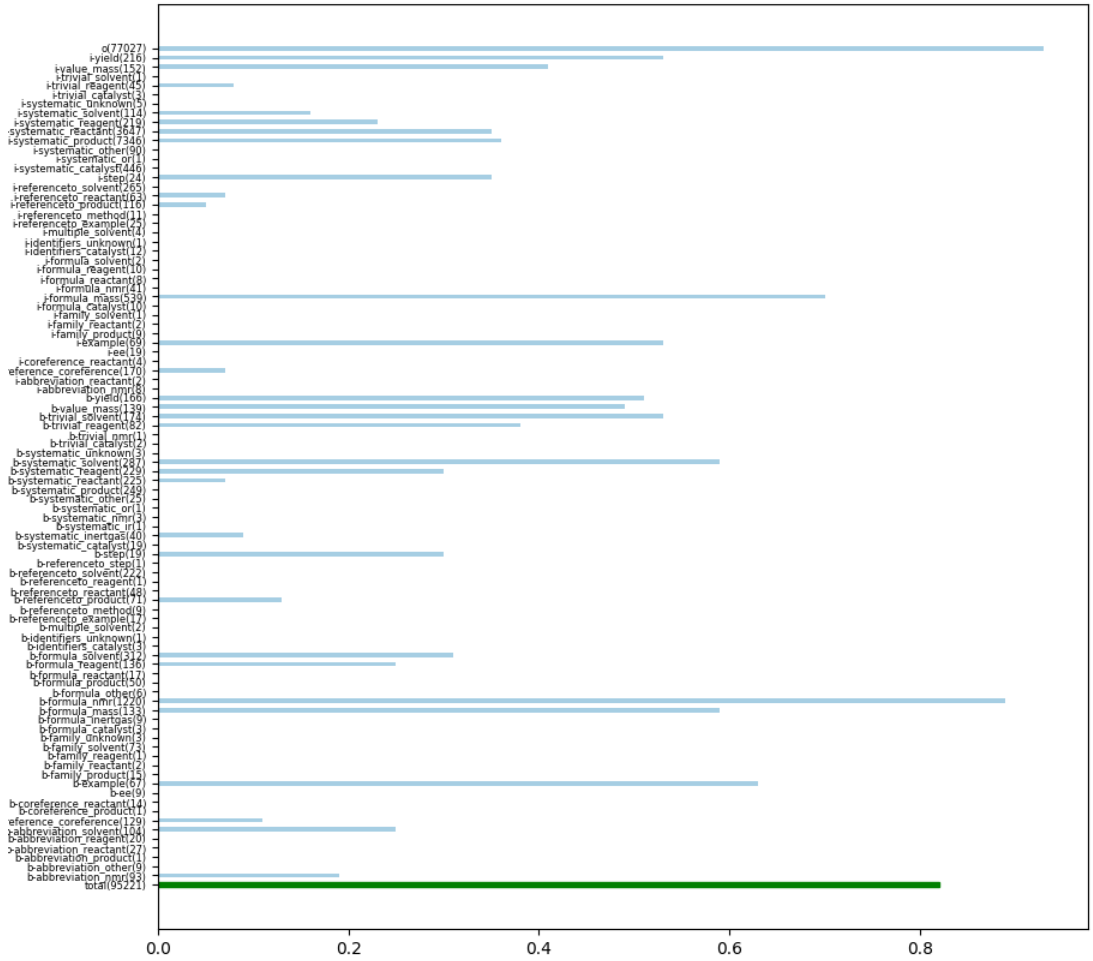

Figure 6.8: Label-wise Performance of Attention Model on Weave 2.0 dataset

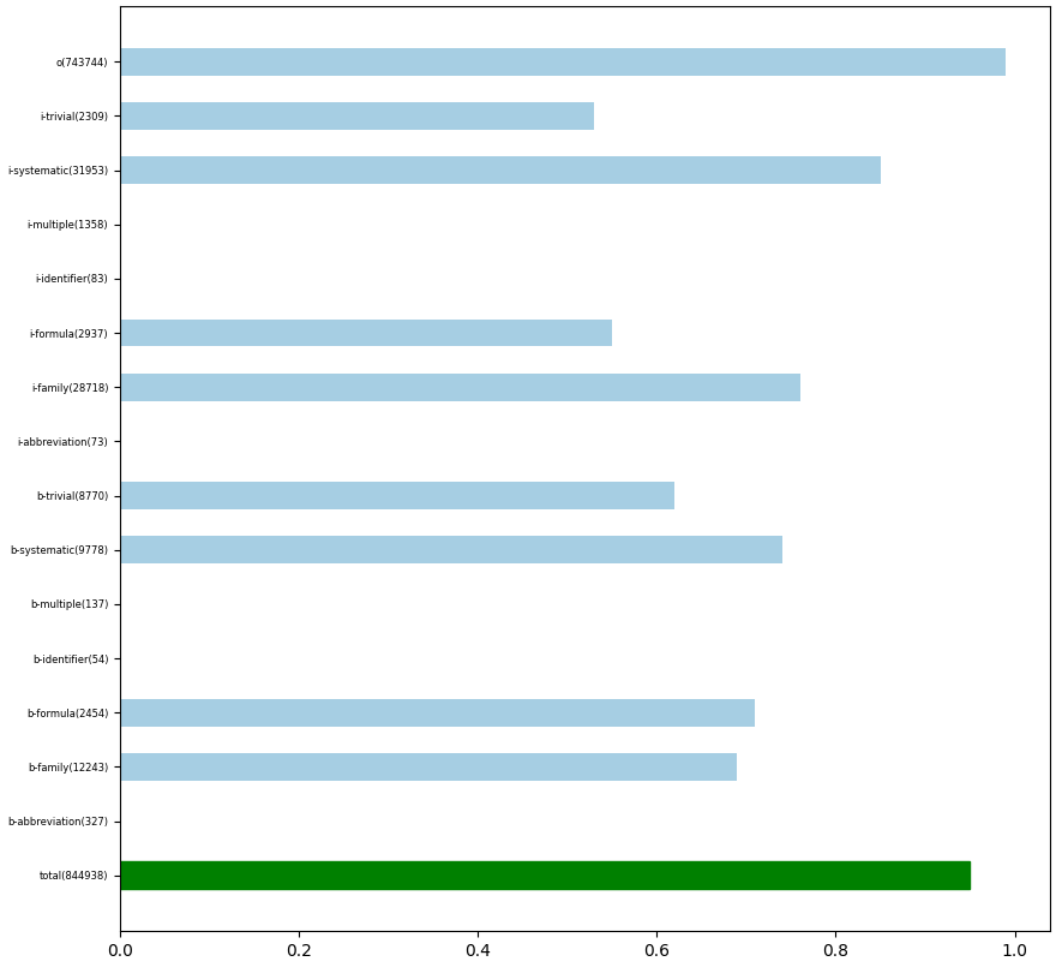

Figure 6.9: Label-wise Performance of Attention Model on CHEMDNER dataset

Chapter 7

Conclusion and Future Work

7.1 Conclusions

We discuss our contributions in this thesis, and the conclusions derived from the analysis of the WEAVE 2.0 dataset, as well as the findings from the models is trained on this dataset.

7.1.1 The WEAVE 2.0 dataset

We introduce a new dataset, WEAVE 2.0, using actual annotated patent data, that adds **role labels** that denote what reaction role the entity performs alongside the existing kinds of labels, usually denoting the type of nomenclature of the chemical entity, to chemical NER tasks. This enhancement of labels would enable downstream tasks to have more information, and allow easier tracking and searching of chemical entities through patents.

Training models on this dataset also enables other, unannotated patent data, as well as data annotated without role labels to be classified using role labels. The dataset also presents a challenging task due to the high number of labels, each of which has two parts, and can thus be formulated in different ways in different architectures, allowing for novel interpretations and representations of this problem.

7.1.2 Chemical NER Models

We also introduce baseline models for the dataset, that utilise a simple architecture, as well as improved models, that are structured for a two-part label, domain-specific task. The improved models involve better architecture including domain-specific embeddings. We show that these improved models not only perform better than the baseline on the WEAVE 2.0 dataset, but on comparing the best model on

a different but similar task (CHEMDNER) that is commonly used in the chemical NER domain, it is able to achieve good results.

7.2 Future Work

As an NER dataset, the WEAVE 2.0 dataset can be used to potentially enhance the performance of downstream models. However, this work does not explore the usage of the WEAVE 2.0 dataset for downstream tasks such as summarisation, question answering etc.

For all of our tasks, we use the default CoNLL tokenisation. However, there exist tokenisers that are specifically designed for chemical tasks, such as Chemtok[1] which may improve the performance of the models.

The WEAVE 2.0 dataset can also be used as a gold standard to train a model and create other secondary datasets on similar chemical document data to train models further or create a better resource for downstream tasks.

The data of the compounds, once extracted, must be stored in a machine-readable format, that also allows similarity between compounds to be detected. This is necessary for searching effectively through the data. We may achieve this by using a format that represents the molecular properties, such as Smiles[24], Selfies[12], or Deepsmiles[18].

Our models use transformer architecture like BERT, however, we do not use more complex architecture like Large Language Models, which may perform better as they have been shown to do extremely well on natural language tasks. This is a potential area for future exploration for this dataset.

Related Publications

Shubhangi Dutta, Manish Shrivastava, Prabhakar Bhimalapuram, *The WEAVE 2.0 Corpus: Role Labelled Synthetic Chemical Procedures from Patents with Chemical Named Entities*. (Pacific Asia Conference on Language, Information and Computation (PACLIC 37), accepted as a full paper for oral presentation.)

Bibliography

[1] A. Akkasi, E. Varoğlu, and N. Dimililer. Chemtok: A new rule based tokenizer for chemical named entity recognition. *BioMed Research International*, 2016:1–9, 01 2016.

[2] I. Beltagy, K. Lo, and A. Cohan. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China, Nov. 2019. Association for Computational Linguistics.

[3] S. Chithrananda, G. Grand, and B. Ramsundar. Chemberta: Large-scale self-supervised pretraining for molecular property prediction, 2020.

[4] H. Cho and H. Lee. Biomedical named entity recognition using deep neural networks with contextual information. *BMC Bioinformatics*, 20, 12 2019.

[5] H. Cho and H. Lee. Biomedical named entity recognition using deep neural networks with contextual information. *BMC bioinformatics*, 20:1–11, 2019.

[6] 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. *Bioinformatics*, 34(20):3539–3546, 04 2018.

[7] K. Degtyarenko, P. Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcántara, M. Darsow, M. Guedj, and M. Ashburner. Chebi: A database and ontology for chemical entities of biological interest. *Nucleic acids research*, 36:D344–50, 02 2008.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018. cite arxiv:1810.04805Comment: 13 pages.

[9] R. I. Doğan, R. Leaman, and Z. Lu. Ncbi disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10, 2014.

[10] A. Erdengasileng, Q. Han, T. Zhao, S. Tian, X. Sui, K. Li, W. Wang, J. Wang, T. Hu, F. Pan, Y. Zhang, and J. Zhang. Pre-trained models, data augmentation, and ensemble learning for biomedical information extraction and document classification. *Database*, 2022, 08 2022. baac066.

[11] M. Krallinger, O. Rabal, F. Leitner, M. Vazquez, D. Salgado, Z. lu, R. Leaman, Y. Lu, D. Ji, D. Lowe, R. Sayle, R. Batista-Navarro, R. Rak, T. Huber, T. Rocktäschel, S. Matos, D. Campos, B. Tang, W. Qi,

and A. Valencia. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7:S2, 03 2015.

[12] M. Krenn, F. Häse, A. Nigam, P. Friederich, and A. Aspuru-Guzik. Selfies: a robust representation of semantically constrained graphs with an example application in chemistry, 05 2019.

[13] R. Leaman, C.-H. Wei, and Z. lu. Tmchem: A high performance approach for chemical named entity recognition and normalization. *Journal of Cheminformatics*, 7:S3, 03 2015.

[14] D. Lowe. *Extraction of chemical structures and reactions from the literature*. PhD thesis, University of Cambridge, 10 2012.

[15] L. Luo, Z. Yang, P. Yang, Y. Zhang, L. Wang, H. Lin, and J. Wang. An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. *Bioinformatics*, 34(8):1381–1388, 11 2017.

[16] D. Q. Nguyen, Z. Zhai, H. Yoshikawa, B. Fang, C. Druckenbrodt, C. Thorne, R. Hoessel, S. A. Akhondi, T. Cohn, T. Baldwin, and K. Verspoor. Chemu: Named entity recognition and event extraction of chemical reactions from patents. In J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, and F. Martins, editors, *Advances in Information Retrieval*, pages 572–579, Cham, 2020. Springer International Publishing.

[17] R. Nittala and M. Shrivastava. The WEAVE corpus: Annotating synthetic chemical procedures in patents with chemical named entities. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 1–9, Indian Institute of Technology Patna, Patna, India, Dec. 2020. NLP Association of India (NLPAI).

[18] N. O'Boyle and A. Dalke. Deepsmiles: An adaptation of smiles for use in machine-learning of chemical structures, 09 2018.

[19] T. Rocktäschel, T. Huber, M. Weidlich, and U. Leser. WBI-NER: The impact of domain-specific features on the performance of identifying and classifying mentions of drugs. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 356–363, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[20] T. Rocktäschel, M. Weidlich, and U. Leser. ChemSpot: a hybrid system for chemical named entity recognition. *Bioinformatics*, 28(12):1633–1640, 04 2012.

[21] L. Smith, L. K. Tanabe, R. J. n. Ando, C.-J. Kuo, I.-F. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C. M. Friedrich, K. Ganchev, et al. Overview of biocreative ii gene mention recognition. *Genome biology*, 9:1–19, 2008.

[22] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, and J. Tsujii. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France, Apr. 2012. Association for Computational Linguistics.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.

[24] D. Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[25] V. Yadav, R. Sharp, and S. Bethard. Deep affix features improve neural named entity recognizers. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 167–172, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[26] Y. Zhang, J. Xu, H. Chen, J. Wang, Y. Wu, M. Prakasam, and W. Qi. Chemical named entity recognition in patents by domain knowledge and unsupervised feature learning. *Database*, 2016:baw049, 04 2016.