B.Comp. Dissertation

Information Extraction in Scholarly Articles

By

Hon Hao Chen

Department of Computer Science

School of Computing

National University of Singapore

2021/22

B.Comp. Dissertation

Information Extraction in Scholarly Articles

By

Hon Hao Chen

Department of Computer Science

School of Computing

National University of Singapore

2021/22

Project No: H123456
Supervisor: A/P Kan Min-Yen
Advisor: Yang Yajing, Abhinav Ramesh Kashyap
Deliverables:
    Report: 1 Volume

Abstract

Natural Language Processing (NLP) tool is one of the methods to extract information from research papers without going through each paper one by one. In this project, we discuss the problem of Information Extraction (IE) in scientific scholarly documents to obtain structured information. We then attempt to build a relationship extractor tool in a supervised-model-learning setting.

Subject Descriptors:
I.2.7 Natural Language Processing
I.7 Document and Text Processing

Keywords:
Natural Language Processing, Scholarly Document Processing, SemEval 2017 Task 10 Extracting Keyphrases and Relations from Scientific Publications

Implementation Software and Hardware:
sciwing, pytorch

Acknowledgement

I would like to thank my supervisor, Prof Kan Min-yen for helping me to vet through this report. Also, special thanks to my partner, Huang Po-Wei for working together developing the SciWING toolkit.

List of Figures

List of Tables

Table of Contents

Chapter 1

Project Objectives

Natural Language Processing (NLP) has been increasingly gaining attention to model how human handles language data. Among many of the NLP applications, Scientific Document Processing (SDP) focuses on processing language information within scholarly documents. SDP tasks include content summarization, logical structure recovery, information extraction and etc. In particular, the process of Information Extraction (IE) helps to extract key meta-data information from a scholarly document. This process helps researchers to generate insights from a scholarly document without having to go through each of the document manually.

Information Extraction (IE) contains two subtasks, namely, Named Entity Recognition (NER) and Relationship Extraction (RE). One key point is that Named Entity Recognition often comes before Relation Extraction (Nasar et al., 2021). For instance, before we can infer the relationships between two or more named entities, we must first resolve which phrases in a document constitute to a named entity.

In this project, We first evaluate our existing NER model performance in the SciWING toolkit on the ScienceIE task. The ScienceIE task was chosen because it addresses both the Named Entity Recognition (NER) and Relation Extraction (RE) tasks. We then propose several improvement on the NER model using knowledge from transfer learning. Finally, we research to model NER and RE jointly. We believe by leveraging information from NER, the performance of RE can be improved significantly.

Chapter 2

Literature Review

Information Extraction (IE) in Natural Language Processing (NLP) is the extraction of structured data from unstructured data (Singh, 2018). It contains various tasks such as Parts-of-Speech (POS) tagging, Text Parsing, Named Entity Recognition (NER), Relation Extraction (RE), etc. Our interest here is Relation Extraction because it helps to capture relation information between two named entities. Sometimes, the RE task is broken down into the first stage of NER, and then the second stage of RE. We will discuss each of these tasks and their joint accomplishment over the following three sections.

2.1 Named Entity Recognition

Carreras et al., 2002 presented a binary AdaBoost NER system for the CoNLL-2002 shared task and achieved the best result. AdaBoost is a tree-based statistical classification meta-algorithm that requires manual intervention of input features such as syntax tagger, word properties and left predictions. Later in the CoNLL-2003 competition, Florian et al., 2003 presented a four classifier combination framework (Robust Linear Classifier, Maximum Entropy, Transformation-based Learning, and Hidden Markov Model) and scored the best in the German language.

Previous State-of-the-art (SOTA) NER systems often rely on hand-crafted features and domain-specific knowledge to perform supervised training on a limited number of labelled training data. However, Lample et al., 2016's work has reached a new SOTA by introducing a Recurrent Neural Network (RNN) architecture. The RNN is based on a bidirectional Long Short Term Memory (LSTM) and a Conditional Random Fields (CRF) that can learn the features automatically from vector representation of individual words. With the presence of RNNs, the focus of the NER model then shifted from the model architecture decisions to the vector representation of words – Word Embeddings or Embeddings from Language Model (ELMo).

Word Embeddings is fundamental for machine learning training. Although one-hot vectors can be generated trivially, it often does not capture any semantics and syntactic regularities among words. Pennington et al., 2014's GloVe embeddings exploit the benefit of transfer learning by training the embeddings with papers from Wikipedia 2014 and Gigaword 5.

Recently, transformers (Vaswani et al., 2017) is another model architecture that has increasingly gained population. In machine learning, BERT Transformer Devlin et al., 2019 was initially designed to solve two tasks, Masked Language Modeling (MLM) and Next Sentence Prediction. However, it has been proven to be able to model language more deeply than RNNs. RNNs tend to have the problem of vanishing gradients when the number of layers are large (Fei and Tan, 2018). In contrast, transformers resolved this issue and could generate word contextual embeddings that were deeply bidirectional. Additionally, domain-specific word embeddings are often preferred in Scholarly Document Processing (SDP) because global embeddings like GloVe might not necessarily contain words that occur in scientific terminologies. SciBERT (Beltagy et al., 2019) in particular, is a domain-specific transformer model trained on biomedical and computer science journal papers.

2.2 Relation Extraction

Relation extraction (RE) refers to the extraction of semantic relationships between two named entities. RE relies heavily on extracted named entities so that it can establish links between them by using rules-based methods or Neural Network (NN) based methods. Rules-based methods include RelEX (Fundel et al., 2006) which uses dependency parse trees and rules filtering to extract candidate relations in the biomedical domain. NN-based methods includes work from (Socher et al., 2012) which employed a Recursive Matrix-Vector Model that learns the embeddings of words in a tree structure. However, it does not extract relationships between two words but the composition of words resulting from the tree-based transition.

2.3 Joint Modelling NER and RE

Recent works tend to solve the tasks of NER and RE in a joint fashion. The intuition is that RE depends largely on extracted named entity, and NER could benefit from RE as well. In the ScienceIE (Augenstein et al., 2017) task, the top 3 performers all used NN-based machine learning techniques when evaluated on unlabelled test data. The top performer AI2 System (Ammar et al., 2017) used a joint modelling system with a BiLSTM-CRF NER model (word embeddings are formed by concatenating CNN parameterized character representation and GloVe word representation) and a RE model (a combination of feature based gazetteer bitmap and RNN based word encodings, passed to a softmax layer to predict relations). However, the AI2 System is not truly joint modelling as the NER model and RE model are trained separately. Nevertheless, the author has stated that training their model jointly might result in better performance, despite being placed on top of the competition.

Several Joint Model of NER and RE has been proposed in recent years. Giorgi et al., 2019 proposed a joint model with the NER portion modelled using Bert Transformer and the RE portion modelled using a Biaffine binary classifier. Luo et al., 2015 proposed a Jointly Entity Recognition and Linking (JERL) model to model entities linking task and capture the mutual dependency between them. The RE portion of JERL is a tree-based ranking system. Miwa and Bansal, 2016 used a similar approach by modelling bidirectional tree-structured LSTM-RNNs stacked on top of bi-directional sequential LSTM-RNNs.

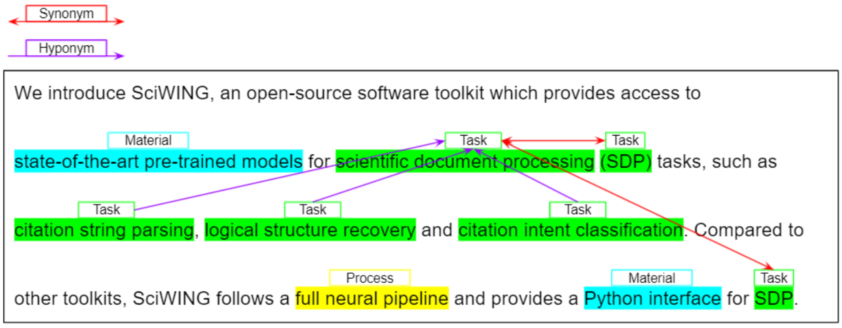
Bekoulis et al., 2018 employed yet a simpler approach by formulating RE as a multi-head selection problem. The NER model is a BiLSTM-CRF model. The generated word encodings from the BiLSTM together with the predicted labels from the CRF output layer are then concatenated. Subsequently, each concatenated instance would form a possible relationships with other concatenated instances to model possible relationships for each word-relation-word triplet in a sentence. The linear combination of each word pair is then passed into a softmax layer for relationship prediction. We would discuss more about this when we implement this in later chapter.

Chapter 3

Scientific Named Entity Recognition

3.1 Problem Statement

The ScienceIE task. The ScienceIE task consists of 500 scientific papers from the Computer Science, Material Sciences and Physics domains. We chose the ScienceIE task because it contains the NER and RE subtasks. The NER subtask requires participants to identify named entities in a document (ScienceIE subtask A) and classify the identified named entities to the Task label, Process label and Material label (ScienceIE subtask B). The RE subtask requires participants to classify a relationship among each pair of identified named entities (ScienceIE subtask C). It could be a Hyponym label, Synonym label or no label. An example of the task annotation is presented in figure 3.1.



We introduce SciWING, an open-source software toolkit which provides access to state-of-the-art pre-trained models for scientific document processing (SDP) tasks, such as citation string parsing, logical structure recovery and citation intent classification. Compared to other toolkits, SciWING follows a full neural pipeline and provides a Python interface for SDP.

Figure 3.1: ScienceIE Task Annotation Example

In the example, there are 9 named entities. For instance, the phrase "citation string parsing" has a Task label and has a Hyponym relationship with another Task label phrase "scientific document processing". More information regarding the ScienceIE task can be found at the ScienceIE website here.

We will solve the NER subtask together using the sequence labelling approach. Under the sequence labelling approach, each token is assigned with a class (Task, Process and Material) in a given input sentence.

The ScienceIE dataset consists of 3 labelled datasets responsible for training, testing and development purposes. We will use the train set to train our model and evaluate our model on the development set and test set. Since all the training and evaluation is based on labelled data, this report only addresses supervised training and testing. Corpus statistics can be found in Appendix A.

Problem Approach. We aim to investigate different embeddings architectures in helping the named entity recognition task. In particular, we compare the difference of using static embeddings (character embeddings + GloVe embeddings) with contextual embeddings (BERT). For contextual embeddings, we further discuss improvement using the SciBERT model which is trained on papers in the scientific domain as well as using domain-specific scientific vocabulary (SciBERT-sci).

The SciWING Toolkit. SciWING is well-built for the Information Extraction task but it only tackled the Named Entity Recognition (NER) subtask using the sequence labelling technique. Here, we present the general model structure regarding the way SciWING NER model handles the NER task. The SciWING toolkit divides the general model structure into several parts, the Embedding Layer, the Model Architecture Layer, the Engine module and the Inference module.

1. Embedding Layer: The Embedding Layer allows users to use any character or word embeddings when converting the words to vectors. In our NER baseline model, character embeddings and word embeddings are concatenated and used. This is recommended by Akbik et al., 2018 because word embeddings alone would have difficulty addressing Out of Vocabulary (OOV) words. The character embeddings is modelled also using a BiLSTM layer. Subsequently, we replace this character + GloVe embeddings layer with pre-trained BERT embeddings to evaluate its effectiveness.

2. Model Architecture: Following Chiu and Nichols, 2016's work, the NER model consists of a Bidirectional LSTM layer for encoding the character + word embeddings and a Conditional Random Field (CRF) layer for word tokens tagging.

3. Engine: The engine module allows users to train the module efficiently by specifying which datasets to use and the number of batches, training epochs, etc.

4. Inference Module: The Inference module takes in the trained model and performs inference on a user input basis.

3.2 Named Entity Recognition Baseline

The sequence labelling NER task is divided into 3 labels (Task, Process and Material) and is trained separately. For each label, BILUO tagging scheme (B-Begining, I-Inside, L-Last, U-Unit, O-None) were utilised to capture more fine-grained labels.

Baseline Performance on NER. Table 3.1 and 3.2 shows the performance of the SciWING NER model on the ScienceIE NER subtask. The hyperparameters used are 75 training epochs, 64 batch size, 0.001 learning rate, 0.1 dropout, character embeddings dimension 20, character BiLSTM encoding dimension 30 and word encoding dimension 350. The word embeddings used is pre-trained GloVe embeddings with 100 dimensions for each token. For OOV words, a randomly generated 100 dimensions embedding is used.

Class	Precision	Recall	F-score	support
Task	0.16	0.06	0.09	193
Process	0.41	0.16	0.23	954
Material	0.28	0.35	0.31	904
avg / total	0.30	0.23	0.26	2051

Table 3.1: NER Sequence Labelling Performance on Test data

Class	Precision	Recall	F-score	support
Task	0.22	0.09	0.13	137
Process	0.48	0.22	0.30	455
Material	0.47	0.46	0.46	562
avg / total	0.45	0.32	0.37	1154

Table 3.2: NER Sequence Labelling Performance on Development data

3.3 Transfer learning from Word Embeddings

We would like to investigate the use of pre-trained Bert embeddings in affecting the outcome of our BiLSTM-CRF tagger. We compare with the use of BERT embeddings, which is a pre-trained transformer model. BERT should provide a better sense of generating contextual embeddings from the sentence inputs as compared to static embeddings from GloVe. A major side benefit of using BERT is that tuning only requires a few epochs compared to using GloVe embeddings for the validation score to converge. In the experiment, using BERT output the best performance in about 8 epochs as compared to the previous Character + GloVe embeddings setting which requires about 60 epochs.

We first used the pre-trained Bert-base-uncased model which is the BERT model version with uncased(convert all character to lowercase) vocabs. Even though we convert all tokens into lowercase, the performance is better as compared to Character + GloVe embeddings. In table 3.3 and 3.4, the delta values show the absolute increment value compared to our NER baseline.

Class	Precision	Recall	F-score	support
Task	0.18	0.10	0.11	193
Process	0.34	0.24	0.28	954
Material	0.38	0.31	0.34	904
avg / total	0.35 (+Δ.05)	0.26 (+Δ.03)	0.30 (+Δ.04)	2051

Table 3.3: Bert-base-uncased Model Performance on Test data

Class	Precision	Recall	F-score	support
Task	0.34	0.21	0.26	137
Process	0.46	0.35	0.40	455
Material	0.62	0.46	0.53	562
avg / total	0.53 (+Δ.08)	0.38 (Δ.06)	0.44 (+Δ.07)	1154

Table 3.4: Bert-base-uncased Model Performance on Development data

Since named entity recognition is generally sensitive to word capitals, we hypothesize that cased vocabs should provide a better outcome in detecting named entities. Next, we also compare with the performance of SciBERT, which is a BERT variant yet pre-trained to perform in scientific tasks. As SciBERT has a better match with the vocabulary domain of the ScienceIE tasks, we hypothesize that SciBERT would then yield higher performance. Finally, pre-trained SciBERT comes with a second version trained with scientific vocabs, this contextual embeddings with domain-specific vocabulary and trained on scientific papers should be the best pre-trained embeddings among all. Table 3.5 and table 3.6 summarise our experiments results.

	Casing			**Domain-specific model**			**Domain-specific vocab**		
	Bert-base-cased			SciBert-base-cased			SciBert-sci-cased		
Class	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
Task	0.21	0.09	0.13	0.29	0.06	0.10	0.11	0.10	0.11
Process	0.37	0.31	0.34	0.44	0.27	0.33	0.39	0.25	0.30
Material	0.46	0.36	0.40	0.41	0.37	0.39	0.40	0.40	0.40
avg	0.40	0.31	**0.35** (+Δ.05)	0.42	0.29	**0.35**	0.36	0.30	**0.33**

Table 3.5: Model Comparison Performance on Test data

	Casing			**Domain-specific model**			**Domain-specific vocab**		
	Bert-base-cased			SciBert-base-cased			SciBert-sci-cased		
Class	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
Task	0.26	0.15	0.19	0.23	0.04	0.06	0.25	0.18	0.21
Process	0.49	0.39	0.43	0.54	0.36	0.43	0.45	0.18	0.40
Material	0.67	0.54	0.60	0.64	0.54	0.59	0.62	0.59	0.60
avg	0.56	0.44	**0.49** (+Δ.05)	0.59	0.41	**0.48**	0.52	0.45	**0.48**

Table 3.6: Model Comparison Performance on Development data

If we look at the results across the bolded average f1 measure, we see that only the cased vocab model fits our assumption. Let's first explain how BERT generates contextual word embeddings. In BERT, each token is tokenized into subwords, resulting in subword embeddings. This specialised way of tokenization helps BERT to handle any unseen vocabularies.

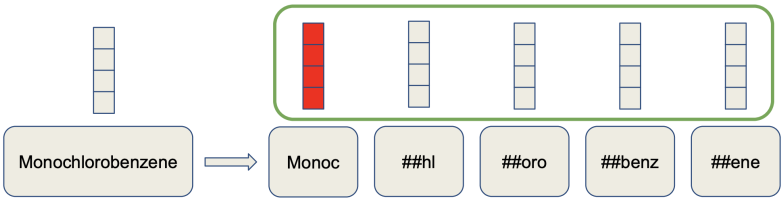


Figure 3.2: BERT Subword Embeddings Example

In figure 3.2, The token "Monochlorobenzene" is tokenized into subwords. Our previous experiment was using the first subword embeddings "Monoc" to represent the entire token embeddings. Perhaps a better strategy is to take into account the average of the generated subwords' embeddings to take account each subword's context. These averaged subwords embeddings should represent the token embedding better. We further test our hypothesis using the subword embeddings averaging strategy.

	Casing			**Domain-specific model**			**Domain-specific vocab**		
	Bert-base-cased			SciBert-base-cased			SciBert-sci-cased		
Class	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
Task	0.14	0.12	0.13	0.22	0.16	0.19	0.19	0.12	0.15
Process	0.37	0.32	0.34	0.36	0.36	0.36	0.37	0.34	0.35
Material	0.44	0.31	0.36	0.45	0.35	0.39	0.42	0.38	0.40
avg	0.37	0.29	**0.33** (+Δ.03)	0.38	0.34	**0.36**	0.38	0.34	**0.36**

Table 3.7: Comparison using Subwords Embeddings Average Strategy on Test data

The results now are much promising as we see improvements from using BERT model, SciBERT model to SciBERT model with scientific vocabulary. However, domain-specific vocabulary only provided minimal improvement on both our test data and development data. In our ScienceIE training dataset (please refer to Appendix A.4), the number of more than one subword produced is 51% when using base vocabularies and 43% when using scientific vocabularies. This

	Casing			**Domain-specific model**			**Domain-specific vocab**		
	Bert-base-cased			SciBert-base-cased			SciBert-sci-cased		
Class	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
Task	0.25	0.22	0.23	0.28	0.21	0.24	0.28	0.20	0.23
Process	0.46	0.39	0.42	0.47	0.49	0.48	0.47	0.47	0.47
Material	0.68	0.50	0.58	0.68	0.51	0.58	0.62	0.55	0.60
avg	0.53	0.42	**0.47** (+Δ.03)	0.54	0.46	**0.50**	0.54	0.58	**0.51**

Table 3.8: Comparison Subwords Embeddings Average Strategy on Development data

8% reduction is reasonable as scientific vocabularies are more resemblance to our training data. The minimal improvement is probably due to the large overlaps (42%) in the base and scientific vocabularies as addressed by Beltagy et al., 2019.

3.4 Relation Extraction Task

Relationship Extraction can be solved as a sequence labelling task. However, This work only for a bidirectional relationship such as the Synonym label where token A and token B correspond to the same meaning. For the Hyponym label, A is a Hyponym of B does not necessarily mean that B is a Hyponym of A. The sequence labelling approach therefore does not solve the problem of directional relation extraction. Regardless, we have done some experiments to gain insights on using the sequence labelling approach in Appendix B.

In the next chapter, we explore the multi-selection approach, which can capture directional relationships among tokens in a sentence. Following work from Bekoulis et al., 2018, we will model the relation extraction task with a joint modelling approach to take advantage of the outputs from the Named Entity Recognition subtask.

Chapter 4

End-to-end NER and RE

We aim to replicate Bekoulis et al., 2018's end-to-end model with pre-trained BERT embeddings as our initial modification. The end-to-end model is currently in development and the overall pipeline has been constructed. However, the model is not fully integrated with BERT yet. Further sanity check is required on the model for its correctness. However, initial experiments have provided promising results. We will explain NER and RE separately and discuss how they are joined.

Named Entity Recognition Model. We will be using the same BiLSTM-CRF architecture as presented in figure 4.1. A key difference in this NER model is that we are using BIO tagging scheme instead of the BILUO scheme used in the previous chapter. Additionally, we only identify named entities without classifying them into different labels. The named entities serve as inputs to the RE model.

Relation Extraction Model. The relationship extraction is modelled as a multi-head selection approach. Figure 4.2 demonstrates the simplified view of the multi-head selection procedure. It takes each word embeddings (together with the BIO label, concatenated) and dot product each relationship with each other token embeddings to form a word-relation-word triplet. Subsequently, this element-wise production is passed to a softmax layer to predict relationships. If a certain production along the lines in the figure passed the softmax threshold, a relationship is predicted. In our experiment, 0.5 threshold value is being used.

Finally, figure 4.3 shows the overall architecture.

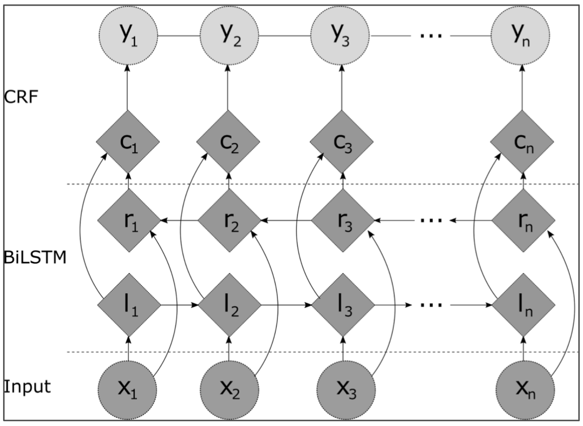


Figure 4.1: NER Model Architecture

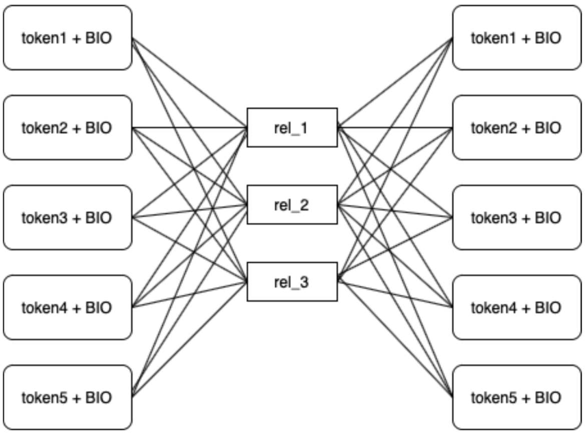


Figure 4.2: RE Model Architecture

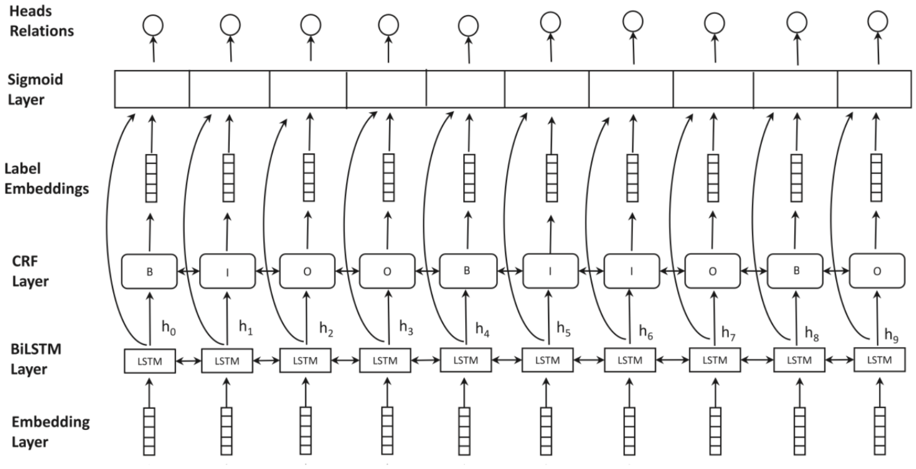


Figure 4.3: NER and RE Joint model

4.1 Influence of NER on RE

As we improve our model bit by bit, we also train our model to demonstrate the influence of NER on RE when both are presented in a joint fashion.

	NER			RE		
Model	Precision	Recall	F-score	Precision	Recall	F-score
Random Word Embeddings	0.57	0.52	**0.54**	0.27	0.05	**0.09**
GloVe Embeddings	0.56	0.56	**0.56**	0.46	0.15	**0.23**
Char+GloVe Embeddings	0.58	0.59	**0.58**	0.45	0.18	**0.25**

Table 4.1: Comparison with Different Improvement on the NER Model

What we can see in our experiment is that as we incorporate better feature representation of our word embeddings from random word embeddings, GloVe embeddings only, to the combination of character and GloVe embeddings, we improve the performance of our NER model. The conclusion made is that improvement on the NER model, in turn, influence our final Relation Extraction performance.

Limitation. A drawback with the ScienceIE dataset includes the reduction of the relationship as our joint model only consider relationships within a sentence. More detailed statistics can be found in Appendix A.3. Due to this reason, we have collected 2 more datasets which is much more designated to the relation extraction task.

1. CoNLL04: Contains the LOCATED_IN, WORK_FOR, ORGBASED_IN and KILL relationships. The vocabs are much more general and are often used in a daily context.
2. ADE_V2: Contains the drug AFFECTS and drug DOSAGE relationships. The vocabs are in the biomedical domain.

Notice that we collected datasets from other domains to show that our model can be easily used in other domains with minimal changes to our model. This is one of the main advantages of using pre-trained word embeddings that fits our use case. Our initial experiments have also shown that improvement on the Named Entity Recognition model influences our Relationship Extraction outputs. Training details can be found in Appendix C.

Chapter 5

Conclusion

We have demonstrated the ability of neural networks to learn and automatically extract features from word embeddings. In NER, pre-trained word embeddings tend to contain more knowledge for the model to learn better and quicker. The context within the subwords embeddings also play a role in representing each word embedding. Most importantly, RE relies on NER to capture key phrases before inferring relationships among them. Joint modelling NER and RE has been proven to be a better approach than using RE alone. Our future work consists of two parts:

1. Finalise our joint model with improvement on the NER model that is deemed to work. Specifically, we are going to include the Bert embeddings in our model. The joint model is separated from the SciWING toolkit due to some design issues and not all modules can be used. After Bert contextual embedding is included, we will do more fine tuning on the RE model and explore other ways to improve the multi-head selection problem.

2. While the model is well-trained, we can see that performance in a supervised setting still has its own limits. We aim to explore unsupervised or semi-supervised learning to leverage vastly available unlabelled data. Semi-supervised method such as Xie et al., 2020's Unsupervised Data Augmentation (UDA) has been proved to improve existing state-of-the-arts. The examination of semi-supervised learning framework is currently at the initial process and there are other frameworks such as Berthelot et al., 2019's MixMatch that is on our list of considerations.

References

Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. *Proceedings of the 27th International Conference on Computational Linguistics*, 1638–1649. https://aclanthology.org/C18-1139

Ammar, W., Peters, M. E., Bhagavatula, C., & Power, R. (2017). The AI2 system at SemEval-2017 task 10 (ScienceIE): Semi-supervised end-to-end entity and relation extraction. *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 592–596. https://doi.org/10.18653/v1/S17-2097

Augenstein, I., Das, M., Riedel, S., Vikraman, L., & McCallum, A. (2017). Semeval 2017 task 10: Scienceie - extracting keyphrases and relations from scientific publications. *CoRR*, *abs/1704.02853*. http://arxiv.org/abs/1704.02853

Bekoulis, G., Deleu, J., Demeester, T., & Develder, C. (2018). Joint entity recognition and relation extraction as a multi-head selection problem. *CoRR*, *abs/1804.07847*. http://arxiv.org/abs/1804.07847

Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3615–3620. https://doi.org/10.18653/v1/D19-1371

Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/1cd138d0499a68f4bb72bee04bbec2d7-Paper.pdf

Carreras, X., Màrquez, L., & Padró, L. (2002). Named entity extraction using adaboost. *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, 1–4. https://doi.org/10.3115/1118853.1118857

Chiu, J. P. C., & Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Fei, H., & Tan, F. (2018). Bidirectional grid long short-term memory (bigridlstm): A method to address context-sensitivity and vanishing gradient. *Algorithms*, *11*(11). https://doi.org/10.3390/a11110172

Florian, R., Ittycheriah, A., Jing, H., & Zhang, T. (2003). Named entity recognition through classifier combination. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, 168–171.

Fundel, K., Küffner, R., & Zimmer, R. (2006). RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, *23*(3), 365–371. https://doi.org/10.1093/bioinformatics/btl616

Giorgi, J. M., Wang, X., Sahar, N., Shin, W. Y., Bader, G. D., & Wang, B. (2019). End-to-end named entity recognition and relation extraction using pre-trained language models. *CoRR*, *abs/1912.13415*. http://arxiv.org/abs/1912.13415

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *CoRR*, *abs/1603.01360*. http://arxiv.org/abs/1603.01360

Luo, G., Huang, X., Lin, C.-Y., & Nie, Z. (2015). Joint entity recognition and disambiguation. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 879–888.

Miwa, M., & Bansal, M. (2016). End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Nasar, Z., Jaffry, S. W., & Malik, M. K. (2021). Named entity recognition and relation extraction: State-of-the-art. *ACM Comput. Surv.*, *54*(1). https://doi.org/10.1145/3445965

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Singh, S. (2018). Natural language processing for information extraction. *CoRR*, *abs/1807.02383*. http://arxiv.org/abs/1807.02383

Socher, R., Huval, B., Manning, C. D., & Ng, A. Y. (2012). Semantic compositionality through recursive matrix-vector spaces. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 1201–1211. https://aclanthology.org/D12-1110

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, u., & Polosukhin, I. (2017). Attention is all you need. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010.

Xie, Q., Dai, Z., Hovy, E., Luong, T., & Le, Q. (2020). Unsupervised data augmentation for consistency training. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *Advances in neural information processing systems* (pp. 6256–6268). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2020/file/44feb0096faa8326192570788b38c1d1-Paper.pdf

Appendix A

ScienceIE Dataset Statistics

Here we present non-exhaustive statistics of the ScienceIE datasets. We provide relevant statistics for each training, development and testing set.

A.1 Name Entities

The statistics for tokens being labelled as name entities are as follow.

	Training Set		Test Set		Development Set	
Criteria	Value	Precentage	Value	Precentage	Value	Precentage
Unique Token	8758	-	4743	-	2884	-
OOVs wrt Train Set	-	-	1758	-	905	-
# of Token	65457	-	22061	-	11253	-
# of Lowercase Token	50830	77.65%	16712	75.75%	8764	77.88%
# of Entity	19865	30.35%	5047	22.88%	3048	27.09%
# of Task	5710	8.72%	987	4.47%	739	6.57%
# of Process	9557	14.60%	2478	11.23%	1362	12.10%
# of Material	5927	9.05%	1885	8.54%	1088	9.67%

Table A.1: ScienceIE Vocabulary statistics, Entities

A.2 Relationships

The statistics for tokens being labelled as relationships are as follow.

	Training Set		Test Set		Development Set	
Criteria	Value	Precentage	Value	Precentage	Value	Precentage
# of Token	65457	-	22061	-	11253	-
# of Relation	3313	5.06%	863	3.91%	686	6.10%
# of Synonym	1355	2.07%	516	2.34%	204	1.81%
# of Hyponym	2081	3.18%	366	1.66%	502	4.46%

Table A.2: ScienceIE Vocabulary statistics, Relationships

A.3 Relationships in Joint Model

Since our joint model can only take in inputs sentence by sentence, relationships that span across sentences are removed. This results in the following statistics below.

	Training Set		Test Set		Development Set	
Criteria	Value	Precentage	Value	Precentage	Value	Precentage
# of Token	65457	-	22061	-	11253	-
# of Relation	820	1.25%	863	1.28%	686	1.61%
# of Synonym	468	0.71%	204	0.92%	88	0.78%
# of Hyponym	352	0.54%	79	0.36%	93	0.83%

Table A.3: ScienceIE Vocabulary statistics, Relationships in Joint Modelling

A.4 Bert Tokenizer

We compare the tokenization results of using base-uncased vocab tokenizer, base-cased vocab tokenizer and scientific-cased vocab tokenizer

	Training Set		Test Set		Development Set	
Criteria	Value	Precentage	Value	Precentage	Value	Precentage
Unique Token	8758	-	4743	-	2884	-
Subword > 1	3471	39.63%	1488	31.37%	776	26.91%
Subword > 5	152	1.74%	75	1.58%	42	1.46%
Max subword length	72	-	66	-	30	-

Table A.4: ScienceIE Vocabulary statistics using base-uncased vocab tokenizer

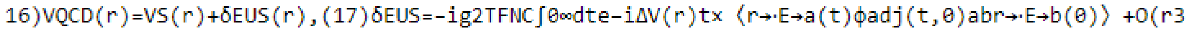
	Training Set		Test Set		Development Set	
Criteria	Value	Precentage	Value	Precentage	Value	Precentage
Unique Token	8758	-	4743	-	2884	-
Subword > 1	4024	45.95%	1720	36.26%	912	31.62%
Subword > 5	181	2.07%	87	1.83%	43	1.49%
Max subword length	28	-	47	-	31	-

Table A.5: ScienceIE Vocabulary statistics using base-cased vocab tokenizer

	Training Set		Test Set		Development Set	
Criteria	Value	Precentage	Value	Precentage	Value	Precentage
Unique Token	8758	-	4743	-	2884	-
Subword > 1	3270	37.34%	1329	28.02%	669	23.20%
Subword > 5	92	1.05%	51	1.08%	22	0.76%
Max subwords length	66	-	**71**	-	28	-

Table A.6: ScienceIE Vocabulary statistics using scientific-cased vocab tokenizer

Of all the subwords with max subwords length, they are either formulas or equations in the scientific domain. A particular example with subwords length 71 tokenized by scientific-cased vocab tokenizer is illustrated in figure A.1



```
16)VQCD(r)=VS(r)+δEUS(r),(17)δEUS=-ig2TFNC∫0∞dte-iΔV(r)tx ⟨r→·E→a(t)ϕadj(t,0)abr→·E→b(0)⟩ +O(r3
```

Figure A.1: Example of Token with 71 Subwords

Appendix B

RE - Sequence Labelling Approach

Here we address the ScienceIE RE subtask using sequence labelling. Each token is labelled as Synonym label or Hyponym label using the BILUO scheme. In the tables below, we only consider evaluations on test data. The O label is removed as our tokens with relationship labels are very few. From Appendix A, only about 5% of tokens contains relationship labels.

Class	Precision	Recall	F-score
B-Synonym	0.3333	0.4032	0.3649
I-Synonym	0.2714	0.4269	0.3318
L-Synonym	0.4133	0.5	0.3525
U-Synonym	0.65	0.4021	0.4968
macro avg	0.4170	0.4331	0.4115
weighted avg	0.4156	0.4341	0.4246

Table B.1: RE Sequence Labelling on Synonym Label using char+GloVe embeddings

Class	Precision	Recall	F-score
B-Hyponym	0.0594	0.3939	0.1032
I-Hyponym	0.0255	0.2617	0.0465
L-Hyponym	0.0591	0.398	0.1029
U-Hyponym	0.0833	0.1774	0.1134
macro avg	0.0568	0.3078	0.0915
weighted avg	0.0459	0.3197	0.0803

Table B.2: RE Sequence Labelling on Hyponym Label using char+GloVe embeddings

This only serves as a preliminary analysis on how relation among key phrases can be extracted under sequence labelling approach. Similarly, we can see improvements in terms of weighted f1 measure when using pre-trained BERT embeddings.

Class	Precision	Recall	F-score
B-Synonym	0.7667	0.371	0.5
I-Synonym	0.8272	0.3918	0.5317
L-Synonym	0.8167	0.3952	0.5327
U-Synonym	0.8154	0.5464	0.6543
macro avg	0.8065	0.4261	0.5547
weighted avg	0.8083	0.4167	**0.5499**

Table B.3: RE Sequence Labelling on Synonym Label using BERT embeddings

Class	Precision	Recall	F-score
B-Hyponym	0.2073	0.1717	0.1878
I-Hyponym	0.1667	0.1121	0.1341
L-Hyponym	0.1951	0.1633	0.1778
U-Hyponym	0.3143	0.1774	0.2268
macro avg	0.2209	0.1561	0.1816
weighted avg	0.2066	0.1530	**0.1758**

Table B.4: RE Sequence Labelling on Hyponym Label using BERT embeddings

Appendix C

Joint Model in Different Domains

The tables below show the performance increments when used different embeddings architectures at the NER model. Jointly modelling NER and RE allows information from NER to be used as a basis to differentiate relationships among named entities at the RE model.

	NER			RE		
Model	Precision	Recall	F-score	Precision	Recall	F-score
Random Word Embeddings	0.82	0.76	**0.79**	0.03	0.04	**0.03**
GloVe Embeddings	0.81	0.79	**0.80**	0.63	0.34	**0.45**
Char+GloVe Embeddings	0.88	0.87	**0.87**	0.61	0.43	**0.51**

Table C.1: Comparison with Different Improvement on the NER Model for the CoNLL04 dataset

	NER			RE		
Model	Precision	Recall	F-score	Precision	Recall	F-score
Random Word Embeddings	0.58	0.55	**0.56**	0.11	0.09	**0.10**
GloVe Embeddings	0.64	0.58	**0.61**	0.15	0.10	**0.12**
Char+GloVe Embeddings	0.65	0.55	**0.60**	0.14	0.10	**0.12**

Table C.2: Comparison with Different Improvement on the NER Model for the ADE dataset