A Deep Relation Extraction from Biomedical Texts with Attention Mechanisms and Domain-Specific Contextual Representations

Mr. Amarin Jettakul

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2018

การสกัดความสัมพันธ์เชิงลึกจากข้อความชีวเวชด้วยกลไกจุดสนใจและตัวแทนข้อมูลแบบพึ่งบริบทเฉพาะด้าน

นายอัมรินทร์ เจตน์ฐากูร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2561

Thesis Title	A Deep Relation Extraction from Biomedical Texts with Attention Mechanisms and Domain-Specific Contextual Representations
By	Mr. Amarin Jettakul
Field of Study	Computer Engineering
Thesis Advisor	Assistant Professor Dr. PEERAPON VATEEKUL
Thesis Co Advisor	Dr. DUANGDAO WICHADAKUL

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Engineering

..... Dean of the Faculty of Engineering
(Professor Dr. SUPOT TEACHAVORASINSKUN)

THESIS COMMITTEE

..... Chairman
(Professor Dr. BOONSERM KIJSIRIKUL)

..... Thesis Advisor
(Assistant Professor Dr. PEERAPON VATEEKUL)

..... Thesis Co-Advisor
(Dr. DUANGDAO WICHADAKUL)

..... Examiner
(Dr. Ekapol Chuangsuwanich)

..... External Examiner
(Dr. Sissades Tongsima)

อัมรินทร์ เจตน์ฐากูร : การสกัดความสัมพันธ์เชิงลึกจากข้อความชีวเวชด้วยกลไกจุดสนใจและตัวแทนข้อมูลแบบพึ่งบริบทเฉพาะด้าน. (A Deep Relation Extraction from Biomedical Texts with Attention Mechanisms and Domain-Specific Contextual Representations) อ.ที่ปรึกษาหลัก : ผศ. ดร.พีรพล เวทีกูล, อ.ที่ปรึกษาร่วม : อ. ดร.ดวงดาว วิชาดากุล

การสกัดความสัมพันธ์แบบชีวเวชเป็นงานที่ต้องการจะศึกษาความสัมพันธ์ระหว่างคำเฉพาะที่ถูกกำหนดไว้ออกจากเอกสารทางชีวเวชซึ่งถูกมองว่าเป็นพื้นฐานสำคัญทางด้านเทคโนโลยีชีวภาพ ตัวอย่างชุดข้อมูลของงานดังกล่าว ได้แก่ การศึกษาความสัมพันธ์ของแบคทีเรียกับแหล่งที่อยู่ และการศึกษาความสัมพันธ์ของชื่อยา วิธีที่ได้รับความนิยมเป็นอย่างมากในงานวิจัยที่ผ่านมา คือ การใช้โมเดลเรียนรู้จากลักษณะหรือโมเดลเรียนรู้เชิงลึก ประกอบกับการใช้โครงสร้างประโยคเชิงพึ่งพิงที่สั้นที่สุด ซึ่งถูกนำเสนอมาและแสดงให้เห็นว่าให้ผลลัพธ์ที่ดี แต่การเรียนรู้จากโครงสร้างประโยคเชิงพึ่งพิงจะมีข้อจำกัดที่จำเป็นจะต้องตัดคำบางคำในประโยคออกไป ซึ่งนำไปสู่การที่โมเดลเรียนรู้จากประโยคที่เหลือได้ไม่เพียงพอ และการแทนที่คำด้วยเวกเตอร์แบบไม่พึ่งบริบทซึ่งถูกใช้ในงานวิจัยก่อน ๆ ที่ผ่านมา อาจจะนำไปสู่ปัญหาความกำกวมของคำได้ งานวิจัยชิ้นนี้ต้องการจะนำเสนอการสกัดความสัมพันธ์เชิงชีวเวชด้วยโมเดลเรียนรู้เชิงลึก ซึ่งเป็นการใช้ลักษณะสำคัญทั้งโครงสร้างประโยคทั้งหมดและโครงสร้างประโยคเชิงพึ่งพิงแบบสั้นที่สุด ประกอบกับกลไกจุดสนใจ นอกจากนั้นยังมีการใช้การแทนที่คำและประโยคด้วยตัวแทนข้อมูลแบบพึ่งบริบทเฉพาะด้าน และงานวิจัยชิ้นนี้ต้องการที่จะวัดประสิทธิภาพโดยรวมของโมเดลด้วย โดยจะแสดงถึงความทนทานของประสิทธิภาพของโมเดลต่อการสุ่มเวกเตอร์ตั้งต้นของโมเดลในหลาย ๆ ครั้ง เพื่อการันตีประสิทธิภาพของโมเดลในการนำไปใช้งานจริงบนโปรแกรมสำเร็จประยุกต์ โดยที่เมื่อเปรียบเทียบกับงานวิจัยอื่น ๆ ที่เป็นมาตรฐานอยู่ในปัจจุบัน ผลการทดลองบนชุดข้อมูลความสัมพันธ์ของแบคทีเรียกับแหล่งที่อยู่แสดงให้เห็นว่าโมเดลที่ใช้วิธีการที่นำเสนอทั้งหมดให้ผลลัพธ์ที่ดีที่สุดบนค่าวัดประสิทธิภาพ (F score) ทั้งค่าที่มากที่สุดและค่าเฉลี่ยอยู่ที่ 60.77%และ 57.63% ตามลำดับ นอกจากนี้โมเดลที่นำเสนอยังให้ผลลัพธ์ที่ดีที่สุดบนชุดข้อมูลความสัมพันธ์ของชื่อยา ด้วยค่าวัดประสิทธิภาพ (F score) มากที่สุดและค่าเฉลี่ยอยู่ที่ 80.3% และ 77.7%ตามลำดับ งานวิจัยชิ้นนี้ได้แสดงให้เห็นว่าวิธีการที่ได้นำเสนอไปทั้งหมดสามารถสกัดคุณลักษณะที่สำคัญของโครงสร้างประโยคและเป็นประโยชน์ต่อการเรียนรู้โมเดลเชิงลึกได้เป็นอย่างดี นำไปสู่ประสิทธิภาพที่ดีที่สุดของโมเดลที่นำเสนอเมื่อเทียบกับโมเดลมาตรฐานต่าง ๆ นอกจากนี้ยังมีการวิเคราะห์ผลการทำนายที่ทั้งถูกและผิดของโมเดลที่นำเสนอ เพื่อนำไปสู่การวิเคราะห์ตัวแปรที่ส่งผลต่อประสิทธิภาพของโมเดล

สาขาวิชา	วิศวกรรมคอมพิวเตอร์	ลายมือชื่อนิสิต .....
ปีการศึกษา	2561	ลายมือชื่อ อ.ที่ปรึกษาหลัก .....
		ลายมือชื่อ อ.ที่ปรึกษาร่วม .....

6170333221 : MAJOR COMPUTER ENGINEERING

KEYWORD: Biomedical text mining, Relation extraction, Deep learning, Attention networks, Domain-specific contextual embeddings

Amarin Jettakul : A Deep Relation Extraction from Biomedical Texts with Attention Mechanisms and Domain-Specific Contextual Representations. Advisor: Asst. Prof. Dr. PEERAPON VATEEKUL Co-advisor: Dr. DUANGDAO WICHADAKUL

The biomedical relation extraction (RE) tasks aim to study the interaction between pre-defined entities from biomedical literature: Bacteria Biotope (BB) and Drug-Drug interactions (DDI) tasks. Some previous investigations have used feature-based models; others have presented deep-learning-based models such as convolutional and recurrent neural networks used with the shortest dependency paths (SDPs). Although SDPs contain valuable and concise information, sections of significant information necessary to define bacterial location relationships are often neglected. In addition, the traditional word embedding used in previous studies may suffer from word ambiguation across linguistic contexts. Here, we present a deep learning model for biomedical RE. The model incorporates feature combinations of SDPs and full sentences with various attention mechanisms. We also used pre-trained contextual representations based on domain-specific vocabularies. In order to assess the model's robustness, we introduced a mean F score on many models using different random seeds. The experiments were conducted on the BB corpus in BioNLP-ST'16 and the DDI corpus in BioNLP-ST'13. For the BB task, our experimental results revealed that our proposed model performed better (in terms of both maximum and average F scores; 60.77% and 57.63%, respectively) compared with other existing models. For the DDI task, our proposed model also gets state-of-the-art performance with a maximum F score of 80.3% and a mean F score of 77.7%. In conclusion, we demonstrated that our proposed contributions to this task can be used to extract rich lexical, syntactic, and semantic features that effectively boost the model's performance. Moreover, we analyzed the correct and incorrect predictions of our model to determine the related factors that affected the model's performance.

Field of Study:	Computer Engineering	Student's Signature .....
Academic Year:	2018	Advisor's Signature .....
		Co-advisor's Signature .....

ACKNOWLEDGEMENTS

A Master's dissertation is a long journey through unknown lands, over sky-high peaks, and through deep and dark troughs. The development of this dissertation cannot be done by myself alone. It is influenced, raised, and nourished by several groups of people. First, I would like to thank my supervisors Asst. Prof. Dr. Peerapon Vateekul and Dr. Duangdao Wichadakul not only for shaping my work from the very start but also for inspiring many of my thoughts. I also thank my dissertation examiners: Prof. Boonserm Kijsirikul, Dr. Ekapol Chuangsuwanich, and Dr. Sissades Tongsima.

I am very fortunate to have met many friendly colleagues at Data Mind Lab and 20th Floor community in the Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University. This dissertation is a direct result of discussion with others throughout the environment of a peer review system. I also thank the scholarship from Chula Computer Engineering Graduate Scholarship for CP Alumni for financial support.

I am grateful for all the amazing people at Tact Social Consulting in ensuring that the fire keeps burning and being there at times when I required motivation and propelling me on the growth mindsets and also for providing me the very hard works. Their support, encouragement and credible ideas have been great contributors in my beneficial life-long learning skills. I am looking forward to seeing you all at future events.

I also want to special thank every support of my family and friends for their unconditional understanding which nourishes me when I need it the most. Much of their serenity is sacrificed in the dark hours where I spent days and nights during this dissertation.

Amarin Jettakul

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

Page

1. INTRODUCTION

1.1. Motivation

Automatic relation extraction is a task which defines the interactions between the biomedical entities from biomedical texts, which is important and getting more attention from biomedical researchers these days. Due to the rapid development of computational and biological technology [1], the biomedical literature is expanding at an exponential rate leading to the difficulty in manually extracting the required information. As the primary shared workshops of various biomolecular event extractions from literature, the BioNLP Shared Task (BioNLP-ST) series represent a community trend in text mining for biology toward fine-grained information extraction. For instance, Drug-Drug Interactions (DDI) [2] and Bacteria-Biotope relations (BB) [3]. In addition, the first BioNLP-ST was organized in 2009, and the time of writing the recent series of this community-wide event was in 2016.

Over the past few years, great efforts have been made in challenging biomedical relation extraction. In general, automatic relation extraction from biomedical texts which can be considered as classification task is divided into two categories, binary and multi-class relation extractions. Firstly, BB task is an example task in binary relation extraction. As the fourth series of BioNLP Shared Task in 2016, this task [3] followed the general outline and goals of the previous tasks in 2011 [4] and 2013 [5]. It aims to investigate the interactions of bacteria: Bacteria entity, and its biotope: Habitats or Geographical entity, from genetic, phylogenetic, and ecology perspectives. It also involves Lives in relation, which is a mandatory relation between related arguments, the bacteria and the location where it lives. An example of the relation between bacteria and location in the biomedical text in BB task can be displayed in Figure 1. Furthermore, for multi-class relation extraction, DDI task [2] in BioNLP-ST'2013 concentrates effects on the novel aspects of the extraction of drug interactions. This task relies on the DDI corpus composed by texts from MedLine abstracts on drug-drug interactions as well as documents describing on drug-drug interactions from the DrugBank database. Figure 2 shows an example of drug mentions and their relation in this task. Several existing works [6-10] have been shown that feature-based methods can be successfully employed for both automatic relation extractions from biomedical texts. However, the feature-based methods are heavily dependent on feature engineering, and it is sometimes difficult to find a machine-learning researcher that has enough sufficient domain knowledge to extract features for the problem.

In recent advances, other than the features-based methods, deep learning (DL) methods have been found to be outstanding and receiving more attention due to its capability to achieve state-of-the-art performance in several NLP tasks. Additionally, DL models demand less feature engineering since they can automatically learn useful features from training data. The examples of popular DL models that have successfully been applied for biomedical relation extraction are Convolutional Neural Networks (CNNs) [11-14] and Recurrent Neural Networks (RNNs) [15-18]. There are some significant differences between RNNs and CNNs in the relation extraction tasks. RNN models, such as Gated Recurrent Unit (GRU) [19] and Long Short-Term Memory (LSTM) [20], are capable of learning some long-term dependency sequential features which are more suitable for handling long sentences. In contrast, CNN models can capture the local features from contexts based on convolution operations which are more appropriate for addressing short sentences in NLP problems.

Despite the success of DL models in previous studies, there are still several limitations to be considered. To begin with, although shortest dependency paths (SDPs) has been shown to contain valuable syntactic features that are important to the relation extraction models, it is also possible for these dependencies to miss some valuable information in the SDP. For example, for BB task, the word “detection”, which should play the key role to define the relationship between bacteria “E. coli” and biotope “CSF”, is not included in SDP as seen in Figure 1, because there is no dependency path between “detection” and the entity mentions. Similarly, for DDI task, Figure 2 shows that the word “cause” which should be important to extract drug “sumatriptan” and drug “mesylate” relation is excluded from SDP. On the other hand, there have been some researchers studying on using full sentences to extract biomedical relations from texts. However, by considering only full sentences which are usually long and complicated, it is very difficult for DL models to learn enough features from only full-sentence features. Additionally, attention networks have demonstrated success in a wide range of biomedical NLP tasks. For instance, Additive attention mechanism [21, 22] is used to focus on only parts of sentence inputs to achieve the state-of-the-art performance in BB task [23] and Entity-Oriented attention [15] is employed to determine which words of the sentences are the most influential for the relationship between a pair of entities. However, there are still some attention mechanisms that have not been explored and used together in biomedical relation extraction such as Multi-Head attention [24] which has been very successfully proposed in general-domain machine translation. Moreover, although traditional context-free word models, such as Skip-Gram [25] and GloVe [26], have been used for many biomedical relation extractions, they only allow a single context-independent representation for each word to learn the corresponding word vector.

This could lead to word sense ambiguation across various linguistic contexts. To alleviate this problem, contextual representation models [27, 28], which use language understanding models to generate word vectors based on their contexts, have been proposed and their pre-trained weights on general-purpose domain are freely available. However, based on the experimental evidence [29], some studies on the biomedical domain have been shown that the pre-trained word embedding model on a general-purpose corpus, such as Wikipedia, is not sufficient enough for biomedical tasks. Finally, estimating the accuracy of biomedical relation extraction model is significant not only to predict the new coming datasets but also to choose the successful method from a given set. For the reliable performance evaluation, we need an estimation method with low bias and low variance. To choose the proper method, the absolute maximum F score, which is usually used as the evaluation metric in the previous works, might be considered to be less important and the estimation methods such as cross-validation and bootstrap, which are more suitable for the biases and trends in real-world applications [30], should be used instead.

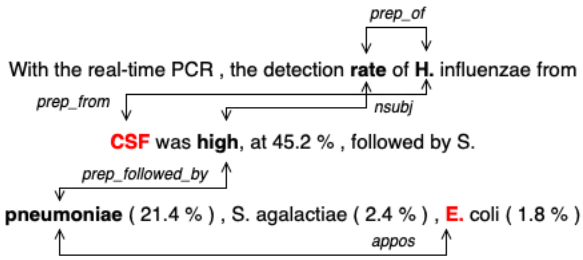

Figure 1. An example of the relation between bacteria and location in BB task. Bacteria and location mentions are in red bold texts; "E. coli" represents the bacteria mention; "CSF" represents the habitat mention; The words in SDP are shown in black bold.

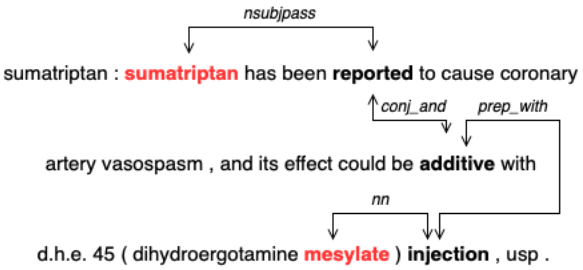

Figure 2. An example of drug mentions and their relation in DDI task. Two drug mentions are in red bold texts. "sumatriptan" and "mesylate" represent the drug mentions; The words in SDP are shown in black bold.

This study proposes a DL relation extraction for biomedical texts. We explore the effectiveness of the hybrid model between RNN model to extract full-sentence features from long and complicated sentences, and CNN model to capture SDP features which are shorter, more valuable, and more concise. We also exploit the integration of three types of attention layers: Additive attention, Entity- Oriented attention, and Multi-Head attention, to enhance the relation extraction task by sentence-level concentration, word-level concentration, and extracting features in multiple subspaces, respectively. Then, we integrate the domain-specific contextual word representation to provide a word sense disambiguation for this task, and contextual sentence representation to improve full-sentence RNN model by embedding sequence sentence information from pre-trained language understanding model. Finally, we compare our proposed model with other existing DL models by the low-bias estimation methods.

1.2. Research Objectives

This thesis studies the problem of automatically extracting relation from biomedical literature with deep learning-based methods. The main hypothesis of this thesis is following: *"Deep neural networks in relation extraction that leverage existing relevant information from related tasks outperform models not using this information across biomedical relation extraction tasks"*. In addition, we goal to evaluate our proposed model with the low-bias metrics to reliably measure the model performance. Overall, we lay out three objectives that will be addressed by the approaches proposed in this work:

- **To propose** a deep learning architecture to extract relations between biomedical entities.
- **To compare** the proposed model with the existing models using low-bias performance metrics.
- **To explore and analyze** the effectiveness of each proposed technique to our model.

1.3. Contributions

Throughout this thesis, we will focus on the five main contributions: a combination of full-sentence and SDP features, attention mechanisms, contextual representation, and low-bias model evaluation. Also, we will show how the contributions in this thesis related to the limitations from existing researches:

- We proposed to combine both full sentences and SDPs as input features of the model to overcome the lack of information in using only SDP of the previous state-of-the-art studies.
- We adapted the integration differential attention mechanisms into the proposed model.
- We presented contextual word and sentence representations that pre-trained on domain-relevant biomedical corpus.
- We evaluated the average model's performance which is less bias than the existing maximum score used in BioNLP challenges.
- We have provided the open-sourced our code.

1.4. Thesis outline

In **Chapter 2**, we provide an overview of background information that is relevant in order to understand the contents of this thesis. We review fundamentals of machine learning and deep learning approaches. We furthermore introduce the challenge BioNLP Shared Tasks (BioNLP-ST) and biomedical relation extraction tasks.

In **Chapter 3**, we review the related literature for novel techniques in NLP that outperform the state-of-the-art on benchmark tasks. In addition, we review the related works in the area of biomedical relation extraction in BioNLP-ST to be compared with our works.

Chapter 4 presents our preprocessing approaches and the proposed model architecture that can be separated into full-sentence model and SDP model. We also provide information about loss functions and optimization algorithms which will be explored in this work.

In **Chapter 5**, we focus on the experimental setups and result analysis of both binary and multi-class relation extraction. For experimental setups, data statistics and our model's hyper parameters used in our experiments are provided in detail. Then, we compare our proposed model with other existing models in the leaderboard. Apart from the comparison with others, each contribution of our model is explored and analyzed its effectiveness to the model's performance. Finally, error analysis is given to examine validation examples that our proposed model misclassified.

Chapter 6 finally contains the conclusions where we summarize our findings and provide the future directions.

1.5. Research schedule

This is section, we provide the following Gantt chart to list all of our project's activities and the duration of each activity:

Table 1. Gantt Chart of the project's activities.

Stage	Task	Activity by Months from the start of the project												
		Jun 18	Jul	Aug	Sep	Oct	Nov	Dec	Jan 19	Feb	Mar	Apr	May	Jun
1	Select research topic													
	Review literature													
2	Design model architecture													
	Implement code													
	Perform experiments #1													
	Analyze results #1													
3	Write paper													
	Submit paper to journal													
4	Write proposal document													
5	Revise model architecture													
	Reimplement code													
	Perform experiments #2													
	Analyze results #2													
6	Write dissertation document													

1.6. Publications

The work in this thesis primarily relates to the following the peer-reviewed article:

1. A. Jettakul, D. Wichadakul, and P. Vateekul, “Relation extraction between bacteria and biotopes from biomedical texts with attention mechanisms and contextual representations”, BMC Bioinformatics (2019).

2. BACKGROUND

This chapter provides background knowledge related to the project is presented to set the stage for novel methods that have been proposed for various NLP tasks. It reviews fundamental knowledge of machine learning (§2.1), BioNLP shared tasks (§2.2), and biomedical relation extraction (§2.3). In addition, it provides fundamental approaches for biomedical relation extraction: SDPs (§2.4) and input representations for NLP models (§2.5). It then introduces the reader to other state-of-the-art techniques for most of NLP tasks: imbalance handling techniques (§2.6), attention mechanisms (§2.7), and contextual representations (§2.8), that would be adapted for biomedical relation extraction in this thesis.

2.1. Machine learning

In this section, we introduce the reader to fundamental knowledge of machine learning, which builds mathematical models from data. In machine learning, each input is typically represented as a vector $\boldsymbol{x} \in \mathbb{R}^d$ of $\boldsymbol{d}$ *features*, where each feature contains the value for a particular attribute of the data and each example is assumed to be drawn independently from the data generating distribution $\hat{p}_{data}$. An entire dataset can be seen as a matrix $X \in \mathbb{R}^{n\,x\,d}$ containing $\boldsymbol{n}$ examples, one example in each row.

In supervised learning, for every input $\boldsymbol{x_i}$, the output is typically a separate label $\boldsymbol{y_i}$, which can be arranged as a vector of labels $\boldsymbol{y}$ for the entire dataset. In unsupervised learning, no designated labels are available. Two common categories of machine learning tasks are classification and regression. In classification, the label $\boldsymbol{y_i}$ belongs to one of a predefined number of classes or categories. In regression, $\boldsymbol{y_i}$ is a continuous number.

Classification further subsumes under binary classification, multi-class classification, and multi-label classification. Binary classification only deals with two classes, while multi-class classification deals with more than two classes. Typically, every example $\boldsymbol{x_i}$ only has one correct $\boldsymbol{y_i}$ label.

2.1.1. Maximum likelihood estimation

The most common way to design a machine learning algorithm is to use the principle of maximum likelihood estimation (MLE). An MLE model is defined as a function $p_{model}(x; \theta)$ that maps an input $\boldsymbol{x}$ to a probability using a set of parameters $\boldsymbol{\theta}$. As the true probability $p(x)$ of an

example x is unknown, we approximate the true probability $p(x)$ with the probability $\hat{p}_{data}(x)$ under the empirical or data generating distribution.

The objective of MLE then is to bring the probability of our model $p_{model}(x;\theta)$ as close as possible to the empirical probability of the input $\hat{p}_{data}(x)$. In other words, MLE seeks to maximize the likelihood or probability of the data under the configuration of the model. The maximum likelihood estimator is defined as

$$\hat{\theta}_{MLE} = argmax_{\theta} \prod_{i=1}^{n} p_{model}(x_i;\theta) \tag{1}$$

In practice, many of the probabilities in the product can be small, leading to underflow. Taking the logarithm does not change the arg max, but transforms the product into a sum, which results in a more convenient optimization problem:

$$\hat{\theta}_{MLE} = argmax_{\theta} \sum_{i=1}^{n} log(p_{model}(x_i;\theta)) \tag{2}$$

2.1.2. Generalization

The goal of machine learning is generalization, training a model that performs well on new and previously unseen inputs. To this end, the available data x is typically split into a part that is used for training, the training set and a second part reserved for evaluating the model, the test set. Performance on the test set is then used as a proxy for the model's ability to generalize to new inputs.

This measure is responsible for the main tension in machine learning: During training, we compute the training error, the error of the model on the training set, which we try to minimize. The actual measure of interest, however, is the generalization error or test error, the model's performance on the test set, which it has never seen before. This is also the main difference to optimization: While optimization seeks to find the minimum that minimizes the training error, machine learning aims to minimize generalization error.

Train and test sets are typically assumed to be *i.i.d.*: Examples in each dataset are *independent* from each other and train and tests sets are *identically distributed*, i.e. drawn from the same probability distribution.

In bias-variance trade-off, the goal to minimize a model's generalization error gives rise to two desiderata: to minimize the training error and to minimize the gap between training and test error. This dichotomy is also known as bias-variance trade-off. If the model is not able to obtain a low error on the training set, it is said to have high bias. This is typically the result of erroneous assumptions in the learning algorithm that cause it to miss relevant relations in the data. On the other hand, if the gap between the training error and test error is too large, the model has high variance. It is sensitive to small fluctuations and models random noise in the training data rather than the true underlying distribution.

If a model has high bias it is also said to be *underfitting*. If a model has high variance, it is known to be *overfitting*. A key factor that determines whether a model underfits or overfits is its capacity, which is its ability to fit a variety of functions. One way to control a model's capacity is to choose an appropriate hypothesis space, the set of functions it can choose from to find the solution. A machine learning model performs best when its capacity is appropriate for the task it is required to solve. A commonly used heuristic is expressed by Occam's razor, which states that among competing hypotheses that explain known observations equally well, one should choose the simplest, which in this context refers to the model with the lowest capacity. However, while simpler functions are more likely to generalize, we still require a hypothesis that is sufficiently complex to achieve low training error.

In practice, a validation set is often used in addition to tune different settings of the model, its hyper-parameters, such as the degree of the polynomial in logistic regression. If the test set is too small, another technique called cross-validation is typically used. Cross-validation repeats the training and test computations on different randomly chosen splits of the data and averages the test error over these splits. The most common variation is k-fold cross-validation, which splits the data into k subsets of equal size and repeats training and evaluation k times, using k−1 splits for training and the remaining one for testing. But in some case that the test set is unseen, cross-validation cannot be used. Then, other low-bias evaluation techniques, such as a bootstrapping method and a measure of average performance, are applied instead.

2.1.3. Regularization

Another way to modify a model's capacity is to encourage the model to prefer certain functions in its hypothesis space over others. The most common way to achieve this is by adding a regularization term $\sum \theta$ to the cost function $J(\theta)$:

$$J(\theta) = MSE + \lambda \sum(\theta) \tag{3}$$

where λ controls the strength of the regularization. If λ = 0, we impose no restriction. As λ grows larger, the preference that we impose on the algorithm becomes more prominent.

The most popular forms of regularization leverage common vector norms. ℓ_1 regularization places a penalty on the ℓ_1 norm, i.e. the sum of the absolute values of the weights and is defined as follows:

$$\sum(\theta) = ||\theta||_1 = \sum_i |\theta_i| \tag{4}$$

where $\theta_i \in \mathbb{R}$. ℓ_1 regularization is also known as lasso (least absolute shrinkage and selection operator) and is the most common way to induce sparsity in a solution as the ℓ_1 norm will encourage most weights to become 0.

ℓ_2 regularization is defined as:

$$\sum(\theta) = ||\theta||_2^2 \tag{5}$$

where $||\theta||_2 = \sqrt{\sum_i \theta_i^2}$ is the Euclidean norm or ℓ_2 norm. Somewhat counter-intuitively, ℓ_2 regularization thus seeks to minimize the squared ℓ_2 norm as in practice, the squared ℓ_2 norm is often more computationally convenient to work with than the ℓ_2 norm. For instance, derivatives of the squared ℓ_2 norm with respect to each element of θ depend only on the corresponding element, while derivatives of the ℓ_2 norm depend on the entire vector. ℓ_2 regularization is also

known as Tikhonov regularization, ridge regression, and weight decay. ℓ_2 regularization expresses a preference for smaller weights in a model.

Different forms of regularization may also be combined. The combination of ℓ_1 and ℓ_2 regularization is also known as elastic net regularization. It uses an α parameter to balance the contributions of both regularizers:

$$\sum(\theta) = \alpha||\theta||_1 + (1-\alpha)||\theta||_2^2 \quad (6)$$

2.1.4. Neural networks

In recent years, neural networks have become the tool of choice in natural language processing (NLP). In this section, we will give an overview of the fundamental building blocks used in neural networks. Neural networks can be seen as compositions of functions. In fact, we can view the basic machine learning models, such as linear regression and logistic regression, as simple instances of a neural network.

2.1.4.1. Multilayer perceptron (MLP)

A neural network is a composition of multiple such affine functions interleaved with non-linear activation functions. The softmax and sigmoid functions are common functions used at the final or *output layer* of a neural network to obtain a categorical and Bernoulli distribution respectively. Non-output layers are referred to as *hidden layers*. Linear regression can be seen as a neural network without a hidden layer and a linear activation function—the identity function—while logistic regression employs a non-linear activation function. Neural networks are typically named according to the number of hidden layers. A model with one hidden layer is known as a one-layer feed-forward neural network, which is also known as a multilayer perceptron:

$$h = \sigma_1(W_1 x + b_1) \quad (7)$$

$$y = softmax(W_2 h + b_2) \quad (8)$$

where σ_1 is the activation function of the first hidden layer. Note that each layer is parameterized with its own weight matrix W and bias vector b.

Computing the output of one layer, e.g. h that is fed as input to subsequent layers, which eventually produce the output of the entire network y is known as forward propagation. As a

composition of linear functions can be expressed as another linear function, the expressiveness of deep neural networks mainly comes from its non-linear activation functions. Besides the sigmoid and softmax functions that are mainly used at output layers, a common activation function for hidden layers is the rectified linear unit (ReLU), which is defined as:

$$\sigma(x) = max(0, x) \tag{9}$$

2.1.4.2. Recurrent neural networks (RNNs)

As text is sequential, we will be using models that can process a sequence of inputs. The most elementary neural network for sequential input is the recurrent neural networks (RNNs). Unlike feedforward networks, RNNs have cyclical connections and are more suitable for NLP tasks where the meaning of a text segment is naturally dependent on what occurred in the narrative before it. RNNs recurrently compose input vectors of a sentence from left to right, effectively letting information persist from the history of the previously seen words. There is usually an input layer, a hidden layer, and an output layer. The output of hidden layer is fed back to itself at consecutive time steps which are many times as there are words in the sentence and the output at any time step is generally the recurrent composition of information until that point. Apart from only the previously seen word information, to exploit signals come from the future part of a sentence in interpreting the current word, running the RNNs from right to left over the input texts can yield additional contextual information. This resulted in bi-directional RNNs (BRNNs) [31] which essentially have two separate RNNs, each with its own parameters, capturing the context at each position from both directions. Then, the output at each time step is a combination of output vectors from both RNNs by concatenation. To handle the problem of vanishing gradients in regular RNN, Long Short-Term Memory networks (LSTMs) has become popular. The state representation of LSTM unit includes an explicit memory cell which is controlled through the gates. These gates control the flow of information based on the previous output and cell state. Given a full sentence of M tokens, (z_1, z_2, ..., z_M), at time step t-th, BLSTM takes the current input representation (z_i), previous hidden state (h_{t-1}), and previous memory cell (c_{t-1}) as its inputs to generate the current hidden state (h_i) and memory cell (c_i). To solve the vanishing gradient problem, LSTM has been proposed based on the gate mechanisms, which we can compute the input gate (i_t), forget gate (f_t), output gate (o_t), and new candidate vectors ($\tilde{c}_t$) as follows:

$$i_t = sigmoid(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{10}$$

$$f_t = sigmoid(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{11}$$

$$o_t = sigmoid(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{12}$$

$$\widetilde{C_t} = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (13)$$

where W_* and b_* are weight matrices and bias vectors, respectively. Then, at time step t-th, the memory cell (C_t) and hidden state (h_t) are calculated with equations (14) and (15) where $\odot$ denotes element-wise multiplication.

$$C_t = (f_t \odot C_{t-1}) + (i_t \odot \widetilde{C_t}) \quad (14)$$

$$h_t = o_t \odot tanh(C_t) \quad (15)$$

For BLSTM [32], the forward LSTM output (h_k^f) and the backward LSTM output (h_k^b) are concatenated into $h_k = h_k^f \oplus h_k^b$.

2.1.4.3. Convolutional neural networks (CNNs)

Another commonly used neural network in NLP tasks is the convolutional neural network (CNN). CNNs utilize layers with convolving filters that are applied to local features. Originally invented for computer vision, CNNs have subsequently been shown to be effective and have achieved excellent results in many NLP tasks. For a given SDP sequence of N tokens, (z_1, z_2, ..., z_N), let $z_i \in \mathbb{R}^k$ be the k-dimensional input embedding vector corresponding to the i-th word in the sequence. The input sequence of length N (padded where necessary) is represented as:

$$z_{1:N} = z_1 \oplus z_2 \oplus \ldots \oplus z_N \quad (16)$$

where $\oplus$ is the concatenation operator. Let $z_{i:i+j}$ refers to the concatenation of words between z_i and z_{i+j}. CNN model is constructed by feature maps which can contain multiple filters. Each convolutional filter is applied to a window of h words to produce a new feature. For example, a feature c_i from a window $z_{i:i+h-1}$ is computed as follows:

$$c_i = f(w \cdot x_{i:i+h-1} + b) \quad (17)$$

where b is a bias parameter, and f is a non-linear function. The concatenation of each feature map (c_i) of possible windows ($z_{1:h}$, $z_{2:h+1}$, ..., $z_{N-h+1:N}$) is used as the whole feature map (c) with the equation (18).

$$c = [c_i ; c_2 ; \ldots ; c_{N-h+1}] \quad (18)$$

After that, a max pooling operation is applied over the feature map to choose the maximum value $\hat{c} = max(c)$. This maximum value represents the feature representation corresponding to

this particular filter, which can effectively constitute the most important feature. For multiple-filter-widths CNN model [33], each filter width produces the maximum feature $\hat{c}$, then the multiple-filter-widths CNN output is a combination of the maximum features from every filter width. For given d filter widths (f_1, f_2, . . ., f_d), the output (c_{out}) is computed as follows:

$$c_{out} = [\hat{c}_1 ; \hat{c}_2 ; \ldots ; \hat{c}_d] \quad (19)$$

2.2. BioNLP shared tasks (BioNLP-ST)

The BioNLP Shared Task (BioNLP-ST, hereafter) series represents a community-wide move toward fine-grained information extraction (IE), in particular biomolecular event extraction. The series is complementary to BioCreative [34]; while BioCreative emphasizes the short-term *applicability* of introduced IE methods for tasks such as database curation, BioNLP-ST places more emphasis on the *measurability* of the state-of-the-art and *traceability* of challenges in extraction through an approach more closely tied to text. The recent workshop of these series is BioNLP open shared task (BioNLP-OST) 2019 which is a continuation of the previous efforts organized around the BioNLP-ST workshop series from 2009 to 2016.

2.3. Biomedical relation extraction

Determining the relationships among biomedical entities is the key point in relation extraction in biomedical domain [35]. The ultimate goal is to locate the occurrence of a specific relationship type between given two entities. There are lots of extraction format available in biomedical domain such as RDF and XML format which is widely used. Relation extraction is usually integrated with the similar challenges as NER, such as creation of high-quality annotated data for training and assessing the performance of relation extraction systems.

There are lots of ongoing research in biomedical relation extraction due to critical roles of biomedical interactions in different biological processes. Many different approaches for biomedical relation extraction have been proposed which can be simple systems that only rely on co-occurrence statistics to complex ones which use syntactic analysis and neural networks. The *co-occurrence* technique is considered as the most straightforward techniques which is based on the fact that if they are mentioned together more frequently, there is a chance that they might be related together in some way. For example, [36] introduce a co-occurrence statistics method to calculate and evaluate the degree of association between disease and relevant drugs from clinical narratives and biomedical literature.

Another approach in this area is *rule-based* approach. In this technique, a set of predefined methods used for biomedical relation extraction. Usually, rules are defined manually by domain experts [37]. Having faced the increasing growth of biomedical data, many approaches utilized machine learning techniques to extract useful information from syntactic structures rather than applying manually derived rules. In machine learning, considered as *classification-based* methods, these rules are automatically generated by using machine learning methods from an annotated corpus.

Relation extraction methods can be improved fundamentally by considering the syntactic and semantic structures. Specifically, syntactic parsing methods such as dependency trees are able to produce syntactic information about biomedical texts which reveal grammatical relations between words or phrases. Hence, most relation extraction systems [8-10, 23, 37-41] rely on a shortest dependency path between two entities which has been shown to be a key feature in their models.

2.4. Shortest dependency paths (SDPs)

The key challenge of biomedical relation extraction is the accurate classification of biomedical interactions in the complicated sentences. Some sentences in the biomedical literature may contain several clauses and words. Therefore, it is very difficult for the machine learning models to capture or learn enough lexical and syntactic features based on only raw full sentences. Many previous studies on biomedical relation extraction suggested that the shortest dependency paths (SDPs) contain the valuable syntactic features that are very important for extracting relations. Examples of SDPs in the sentences from BB and DDI tasks are given in Figure 1 and 2, respectively. In Figure 1, “E. coli” is bacteria mention and “CSF” is its location mention that appears in the same sentence. The dependency path is generated by syntactic tools and can effectively represent the syntactic dependency relations of the sentence. For example, “rate” and “H.” are two words in the sentence, and “prep_of” denotes that the dependency relation between “rate” and “H.” is the “prep_of” type which is a prepositional modifier of the word. Such syntactic information is helpful for identifying the relation between “E. coli” and “CSF” entities. As seen in Figure 1, all of the words associated with two entities, such as “rate”, “H.”, “high”, and “pneumoniae” are shown in bold texts which means these words are on the SDP.

2.5. Input representations for NLP models

Choosing a suitable representation of the input data is a vital part of deep learning tasks because neural networks tend to be relatively robust to the choice of input representation. The only requirements for neural network input representations are that they are complete (in sense of

containing all information required to effectively predict the outputs) and reasonably compact. Although irrelevant inputs are not as much of a problem as they are for algorithms suffering from the so-called curse of dimensionality [42], having a very high dimensional input space leads to an excessive number of input weights and poor generalization. Beyond that, the choice of input representation is something black art, whose aim is to make the reasonable relationship between the inputs and the targets as simple as possible. In biomedical relation extraction tasks, there are many interesting ideas of input representations in the following paragraphs.

2.5.1 Word representation

Since the beginnings of NLP, word representation learning has been one of the main research areas. Distributed semantic representation have been proved to be effective and flexible keepers of prior knowledge to be integrated into downstream applications. Recently, neural-network-based approaches which process massive amounts of textual data to embed word semantics into low-dimensional vectors, the so-called word embeddings. Techniques such as Skip-Gram and Continuous Bag-of-Words have been shown to be effective in storing valuable syntactic and semantic information [25]. Another prominent word embedding architecture is GloVe [26] which combines global matrix factorization and local context window methods through a bi-linear regression model. More importantly, these word embeddings can also be used as pre-trained weights for downstream NLP applications. There are many pre-trained word weights that are freely available in multiple domains, such as general-purpose [26, 43, 44] and biomedical [45] domains.

2.5.2. Part-of-speech representation

The motivation of part-of-speech (POS) representation is like word representation. POS tags are categories of words which have grammatical properties. Words that are assigned to the same POS tag display similar behavior in terms of syntax. In general, POS representation is used to distinguish the semantic meaning in different sentences.

2.5.3 Distance representation

In the recent study for relation extraction, distance embedding [13] has been proposed by the assumption that each entity pair in the relation extraction task tends to constitute a relation if the distance between entities is short. The distance is derived from the relative distances of the current word to both biomedical entities in a sentence. For example, in Figure 1, the relative

distances of the word “high” to bacteria “E.” and location “CSF” are 22 and −2, respectively. To construct the distance embedding $D(l)$, each dimension $d(l)$of the distance embedding is initialized with equation (20) where l is the relative distance, s refers to the maximum of the relative distances in the dataset. The distance vectors $dist_1$ and $dist_2$ represent the distance embeddings $D(l)$ of the current word to both biomedical entities, respectively, which are bacteria and location mentions in the previous example.

$$d(l) = \tanh\left(\frac{l}{s}\right) \tag{20}$$

2.5.4 Positional encoding

Since no-recurrence models such as CNNs cannot make use of the order of sequence like RNNs. Positional encoding has been successfully used in the machine translation task [24]. In addition, positional encoding can be used for CNN model to inject some information about the absolute position of the words in the sentence. The positional encodings usually have the same dimension as the word embeddings so that we can sum two of these dimensions. These positional encoding (PE) vectors are initialized by sine and cosine functions of different frequencies to easily learn to attend by relative position with equation 21 and 22, where pos is the position, i is the dimension, and d_{PE} is the size of positional encoding.

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{PE}}}}\right) \tag{21}$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{PE}}}}\right) \tag{22}$$

2.6. Handling imbalanced data

Learning from imbalanced data has emerged as a new challenge to the machine learning and text mining communities. The data imbalance problem often occurs in classification scenarios when a portion of the classes possesses many more examples than others. When standard classification algorithms are applied to such skewed data, they tend to be overwhelmed by the major categories and ignore the minor ones. There have been several endeavors in handling

imbalanced datasets for NLP tasks. Here, we only focus on sampling strategy [46] and focal loss [47].

2.6.1. Under-sampling strategy

Under-sampling seeks to reduce the number of majority class members in the training set [46]. As a result, the overall number of labels in the training set is greatly reduced. This means that during classification, training time is also greatly reduced. Since we are dealing with very high dimensional datasets, there is a significant savings in memory as well. However, because we are eliminating members from the majority class, it is possible that we will lose a lot of valuable information if we eliminate documents that could be useful to our classifier in building an accurate model. Random under-sampling is a simple approach to resampling. Majority class documents in the training set are randomly eliminated until the ratio between the minority and majority class is at the desired level. Theoretically, one of the problems with random under-sampling is that one cannot control what information about the majority class is thrown away. In particular, very important information about the decision boundary between the minority and majority class may be eliminated. Despite its simplicity, random under-sampling has empirically been shown to be one of the most effective resampling methods.

2.6.2. Focal loss

Initial goal of focal loss function proposed by [47] is to address the problem of extreme balance between foreground and background classes during training in object detection scenarios. Focal loss is mainly used for object detection, some studies [48] also show that its sparse-specific characteristics are also applicable for NLP classification problem with imbalanced dataset.

The starting point of focal loss is the cross-entropy loss function [49] for binary classification, defined as:

$$CE(p,y) = \begin{cases} -log(p), & if\ y = 1, \\ -log(1-p), & otherwise, \end{cases} \tag{23}$$

in which $y \in (-1,1)$ denotes the ground truth for negative and positive classes, respectively, and $p \in (0,1)$ indicates the model's estimated probability for the class with label y = 1. Cross-entropy loss exhibits a loss with nontrivial magnitude even with easily classified samples. Therefore, these small loss values, accumulated with a large number of easy samples, can easily surpass the rare class.

For simplicity, let

$$p_t = \begin{cases} p, & if\ y = 1, \\ 1 - p, & otherwise, \end{cases} \quad (24)$$

In order to balance the importance of positive and negative samples, a weighting factor $\alpha \in [0,1]$ is introduced in a similar notation:

$$\alpha_t = \begin{cases} \alpha, & if\ y = 1, \\ 1 - \alpha, & otherwise, \end{cases} \quad (25)$$

For reducing the loss contribution from easily classified samples, a modulating factor m with a tunable focusing parameter $\gamma \geq 0$ is introduced to the cross-entropy loss:

$$m = (1 - p_t)^{\gamma}. \quad (26)$$

Taking these two new factors into equation (23), the focal loss function becomes:

$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} log(p). \quad (27)$$

Note that α and γ are two parameters indicating how sensitive it is to the easily classified samples. In this work, we propose to apply this focal loss function to the end of our proposed model architecture for multi-class relation extraction task.

2.7. Attention mechanisms

In recent advances, attention mechanisms have been successfully applied to biomedical relation extraction task since they are able to learn a vector of important weights for each word in a sentence to reflect its level of effect on the final result. These mechanisms encourage the model to use only parts of the input where the most relevant information is concentrated instead of the entire sentence. In the biomedical relation extraction, many previous studies [15, 23] have found that different words in each sentence should have different influences on the biomedical relations. For instance, in Figure 1, the word “detection” is more important to define the interaction between “E.” and “CSF” than the word “high” in bacteria-biotope relation extraction. Thus, attention mechanisms can improve the performance of relation extraction model.

2.8. Contextual representations

The choice of how to represent words or sentences poses a fundamental challenge for NLP communities. Most importantly, words have different meanings in different contexts. At a coarse-grained level, this was captured by experts in crafting WordNet, in which, for example, *get* is mapped to over thirty different meanings (or sense). It is difficult to obtain widespread agreement on how many senses should be allocated to different words, or on the boundaries between one sense and another; word senses may be fluid. For example, the word *bank* can refer to the side of a river or to a financial institution. When used to refer to a blood bank, we can debate whether the second sense is evoked or a third. Indeed, in many NLP models based on neural networks, the very first thing that happens is that each word vector is passed into a function that transforms it based on the word in its nearby context, giving a new version of the word vector, now specific to the token in its particular context.

With hindsight, we can now see that be representing word independent of context, we were solving a problem that was harder than it needed to be. Because words mean different things in different contexts, we were requiring that type representations capture *all* of the possibilities (e.g., the thirty meanings of *get*). To simplifies things, asking the word representation to capture only what a word means *in this context*. For the same reasons that the collection of contexts provides clues about its meanings, a particular token's context can provide clues about its specific meaning. For instance, you may not know what the word *blicket* means, but if you are told that "I ate a strawberry blicket for dessert", you likely to have a good guess.

Based on recent researches, adding contextual vectors into various NLP tasks such as named entity extraction, sentiment analysis, and question answering, has shown to massively improve the state-of-the-art results. These contextual vectors are the output of pre-trained language models. Details on these pre-trained language models will be discuss in the following paragraphs.

ELMo [28], which stands for *embeddings from language models*, brought a powerful advance in the form of word vectors—i.e., vectors for words in context, or *contextual word vectors*— that are pretrained on large corpus. The important insight behind ELMo is that if every word token is going to have its own vector, then the vector should depend on an arbitrarily long context of nearby words. To obtain a *context vector*, we start with word vectors, and pass them through a neural network that can transform arbitrary-length sequences of left- and/or right- context word vectors into a single fixed-length vector. Unlike context-free word vectors, which are essentially lookup tables, contextual word vectors include both type-level vectors and neural network

parameters that *contextualize* each word. ELMo trains one RNN model for left contexts (going back to the beginning of the sentence a token appears in) and another RNN model for right contexts (up to the end of the sentence). Longer contexts, beyond sentence boundaries, are in principle possible as well. In Figure 3, for example, the word *"play"* in the sentence using context-free word embeddings encodes multiple meanings such as the verb *"to play"* or in the case of the sentence a theatre production. In context-free word embeddings such as Glove or Word2Vec each instance of the word ***play*** would have the same representation which cause word-sense ambiguation problem. Thus, ELMo enables NLP models to better disambiguate between the correct sense of a given word. Whether this development completely solves the challenge of words with different meanings remains to be seen, but ELMo was shown to be extremely beneficial in various NLP tasks.

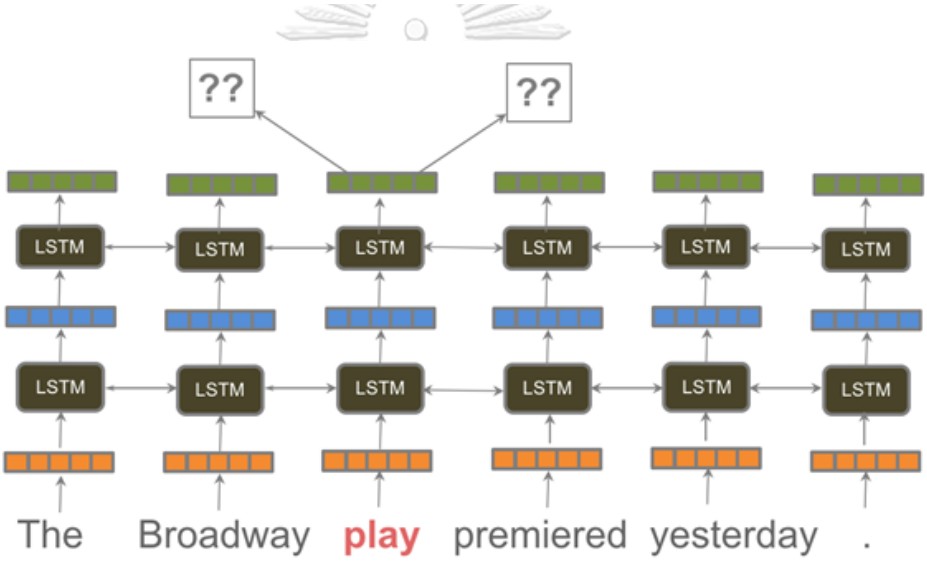

Figure 3. An example of how ELMo generates a contextual word representation.

BERT [27] (*bidirectional encoder representations from transformers*), unlike ELMo that uses language modeling (LM) as its pre-training task, instead replaces language modeling with a modified objective, so-called masked language modeling. In this model, words in a sentence are randomly erased and replaced with a special token *"masked"* with some small probability, 15%. Then a Transformer, instead of RNNs, is used to generate a prediction for the masked word based on the unmasked words surrounding it, both to the left and right. Apart from masked language modeling to learn relationships between words in different contexts, BERT is also pre-trained on next-sentence-prediction task to learn relationships between sentences. Using these objectives, BERT is able to achieve state-of-the-art performance on a variety of NLP tasks.

3. RELATED WORKS

In recent years, there are many efforts to study biomedical relation extraction according to BioNLP-ST. This work will focus on binary relation extraction (e.g. bacteria-biotope task) and multi-class relation extraction (e.g. drug-drug interaction task). In biomedical relation extraction, the F score, precision and recall are widely used as major evaluation metrics. Compared to precision and recall, the F score provides a more reasonable combination of both precision and recall and can be used to evaluate the overall performance. Previous works on our biomedical relation extraction tasks will be discussed in the following paragraphs.

Table 2 lists all existing works for the BB task. TEES [8] was the best-performed system in BioNLP-ST'13 which adopted the support vector machine (SVM) with a variety of features based on SDPs to get F score of 42%. VERSE team [10] proposed a utilized SVM with complex-designed features and minimum spanning dependency tree (MST) achieving 55.8% F score, which was the first place in BioNLP-ST'16. Other than the features-based methods for the BB task, several previous studies using deep learning approaches have significantly outperformed the traditional SVM approach. For example, in BioNLP-ST'16, DUTIR [41] utilized CNN models to achieve 47.8% F score and TurkuNLP [40] proposed multiple-LSTM with SPDs to achieve F score of 52.1% ranked second place in the competition. Afterward, DET-BLSTM [39] applied LSTM with a dynamic extended tree (DET) adapted from SDPs achieving the F score of 57.14%. Recently, BGRU-Attn [23] have proposed bidirectional-GRU (BGRU) with additive attention and domain-oriented distributed word representation to become the state-of-the-art deep learning system without hand-designed features for the BB task with 57.42% F score.

Similar to the BB task, existing works of the DDI task have been employed by either feature-based or deep-learning-based methods as shown in Table 3. Uturku [9] proposed SVM with information from domain resources such as DrugBank and dependency features. FBK-irst [7] combined different characteristics of three kernels to rank first in BioNLP-ST'13 with 65.1% F score. RAIHANI [6] utilized SVM with many rules and features such as chunk, trigger words, and filtering negative sentence to achieve F score of 71.1%. For CNN-based systems, Liu-CNN [11], MCCNN [12], and TEES-CNN [38] used CNN model with syntactic features, CNN model with multichannel word embeddings, and multiple-filter-widths CNN model with different vector embeddings to achieve F score of 69.8%, 70.2%, and 73.5%, respectively. Some examples of RNN-based systems are Joint AB-LSTM [16] which employed two LSTM networks, one of which exploited the pooling attention, to get 71.5% F score and Char-RNNs [18] which proposed the combination of character-level and

word-level representations to achieve F score of 72.1%. In addition, the best system for DDI task is Attn-BLSTM [15] that introduced the integration of BLSTM and Entity-Oriented attention and achieved the highest F score of 77.3%.

Table 2. Performance comparison of existing models on BB corpus.

Model		Precision	Recall	F score
Feature-based	TEES [8]	61.6	38.4	42.3
	VERSE [10]	51.0	61.5	55.8
CNN-based	DUTIR [41]	60.0	39.7	47.8
RNN-based	TurkuNLP [40]	**62.3**	44.8	52.1
	DET-BLSTM [39]	56.3	58.0	57.1
	BGRU-Attn [23]	48.8	**69.8**	**57.4**

Table 3. Performance comparison of existing models on DDI corpus.

Model		Precision	Recall	F score
Feature-based	Uturku [9]	73.2	49.9	59.4
	FBK-irst [7]	64.6	65.6	65.1
	RAIHANI [6]	73.7	68.7	71.1
CNN-based	Liu-CNN [11]	75.7	64.7	69.8
	MCCNN [12]	76.0	65.3	70.2
	TEES-CNN [38]	**80.5**	67.6	73.5
RNN-based	Joint AB-LSTM [16]	74.5	65.0	71.5
	Char-RNNs [18]	80.0	65.9	72.1
	Hierarchy RNN [17]	74.1	71.8	72.9
	Recursive NN [50]	77.8	69.6	73.5
	Attn-BLSTM [15]	78.4	**76.8**	**77.3**

As mentioned in the motivation section (§1.1), although these models have been shown to well perform in both tasks, there are some limitations that can be tackled to improve the model's performance in this work. The first constraint is the fact that SDP can miss some important information for biomedical relation extraction as shown in Figure 1 and 2, whereas learning feature from the only full sentence is not sufficient. In addition, there are some attention

networks which have never been explored in biomedical tasks. Finally, traditional word embedding models, which are used in all existing methods, suffer from word sense ambiguation across various linguistic contexts.

4. METHODOLOGY

In this section, we describe the proposed methods to extract relation relations from the biomedical literature.

4.1. Text preprocessing

We used the TEES system [8, 38, 40] to run the pipeline of the text preprocessing steps. Tokenization and part-of-speech (POS) tagging for each word in a sentence were generated using the BLLIP parser [51] with the biomedical-domain model. The dependency grammar resulted from the BLLIP was further processed using the Stanford conversion tool [52] to obtain the Stanford Dependencies (SD) graph.

We then used Dijkstra's algorithm to determine the SDPs between each pair of entities. The SDPs represented the most relevant information and diminished noises by undirected graph (Figures 1 and 2). An entity pair was neglected if there was no SDP between the entities. While the dependency paths only connect a single word to others within the same sentence (intra-sentence), there are some cross-sentence (inter-sentence) associations that can be very challenging in terms of the extraction task. In order to compare with other existing works, only intra-sentence relations were considered.

To ensure the generalization of machine learning-based models, we followed the protocol of previous studies [15, 23] that blinded the entities in a sentence. Two entities were replaced by "entity_1" and "entity_2," respectively. For example, as shown in Table 3, we can generate two BB relation candidates (termed "instances") from a sentence "Long-term ***Helicobacter pylori*** infection and the development of atrophic *gastritis* and gastric cancer in *Japan*.", where the bacteria and location mentions are highlighted in bold italics and italics, respectively. After entity blinding, we converted all words to lowercase.

Table 4. Relation candidates (instances) in a sentence after entity blinding.

Entity pair	Relation candidates after entity binding
(Helicobacter pylori, gastric)	Long-term **entity_1** infection and the development of atrophic gastritis and ***entity_2*** cancer in Japan.
(Helicobacter pylori, Japan)	Long-term **entity_1** infection and the development of atrophic gastritis and gastric cancer in ***entity_2***.

4.2. Input embeddings

The input representations used in our model were divided into full-sentence and SDP features. Let $(w_1, w_2, ..., w_m)$ and $(s_1, s_2, ..., s_n)$ denote the full sentence and SDPs of a sentence that are represented by different embeddings. Each word w_i in a full sentence was represented by word vector, POS, and distance embeddings. Each word s_j in the SDP was represented by word vector, POS, and distance embeddings together with positional encoding (PE). The detailed embeddings used in our model are explained below.

4.2.1. Word embedding

we use either context-free word embedding ($2.5.1) which was a 200-dimensional word vector that built from a combination of PubMed and PMC texts [45] or contextual word embedding from ELMo ($2.8) that pre-trained on PubMed.

The contextual word vector used in our proposed model was generated by ELMo [28]. ELMo learned word representations from the internal states of a bi-directional language model (biLM). It was shown to improve the state-of-the-art models for several challenging NLP tasks. Context-free models such as Skip-gram [25] and GloVe [26] generate a single word representation for each word in their vocabulary. For instance, the word “cold” would have the same representation in “common cold” and “cold sensation” [53]. On the other hand, contextual models will generate a representation of the word “cold” differently based on context. This representation can be easily added to our proposed model by reconstituting the 200-dimensional word vectors with the new pre-trained contextual word vectors. Currently, the ELMo model, pre-trained on a large general-purpose corpus (5.5 billion tokens), is freely available to use [28]. However, [29, 54] showed that domain-irrelevant, pre-trained word embedding models on large, general-purpose collections of texts are not sufficient for biomedical-domain tasks. Therefore, we present a pre-trained, domain-specific, contextual, word-embedding model based on a bacterial-relevant corpus. Inspired by the relevance-based word embedding [55], the corpus to pre-train our proposed contextual word embedding model included relevance-based abstracts downloaded from PubMed. For example, for BB task, the pre-trained corpus contains only sentences with bacterial scientific names from the task (118 million tokens). To evaluate the effectiveness of the domain-specific, contextual, word-embedding model, we compared it with the contextual model pre-trained from randomly selected abstracts from PubMed with the same number of tokens. All of the pre-trained models were fine-tuned with the particular dataset in order to transfer learned features from the pre-trained models to our task.

4.2.2. Part-of-speech embedding

Part-of-speech embedding ($2.5.2) was initialized randomly at the beginning of the training phase.

4.2.3. Distance embedding

Distance embedding ($2.5.3) is derived from the relative distances of the current word to the particular entity.

4.2.4. Positional encoding (PE)

For SDP in the CNN model, we further used PE ($2.5.4) to inject some information about the absolute position of the words in the sentence. The PE vectors were initialized by sine and cosine functions of different frequencies; these functions embed information based on their relative position. Because PE has the same dimension as the word embedding, we can sum these two vectors.

In summary, the overall input embedding representation for a word w_i in full sentences is z_i = $[w_i^{word} ; w_i^{pos} ; w_i^{dist_1} ; w_i^{dist_2}]$. Similarly, for a given word s_j on the SDP the overall input embedding representation is $z_i = [w_i^{word} + w_i^{PE} ; w_i^{pos} ; w_i^{dist_1} ; w_i^{dist_2}]$.

4.3. Model architecture

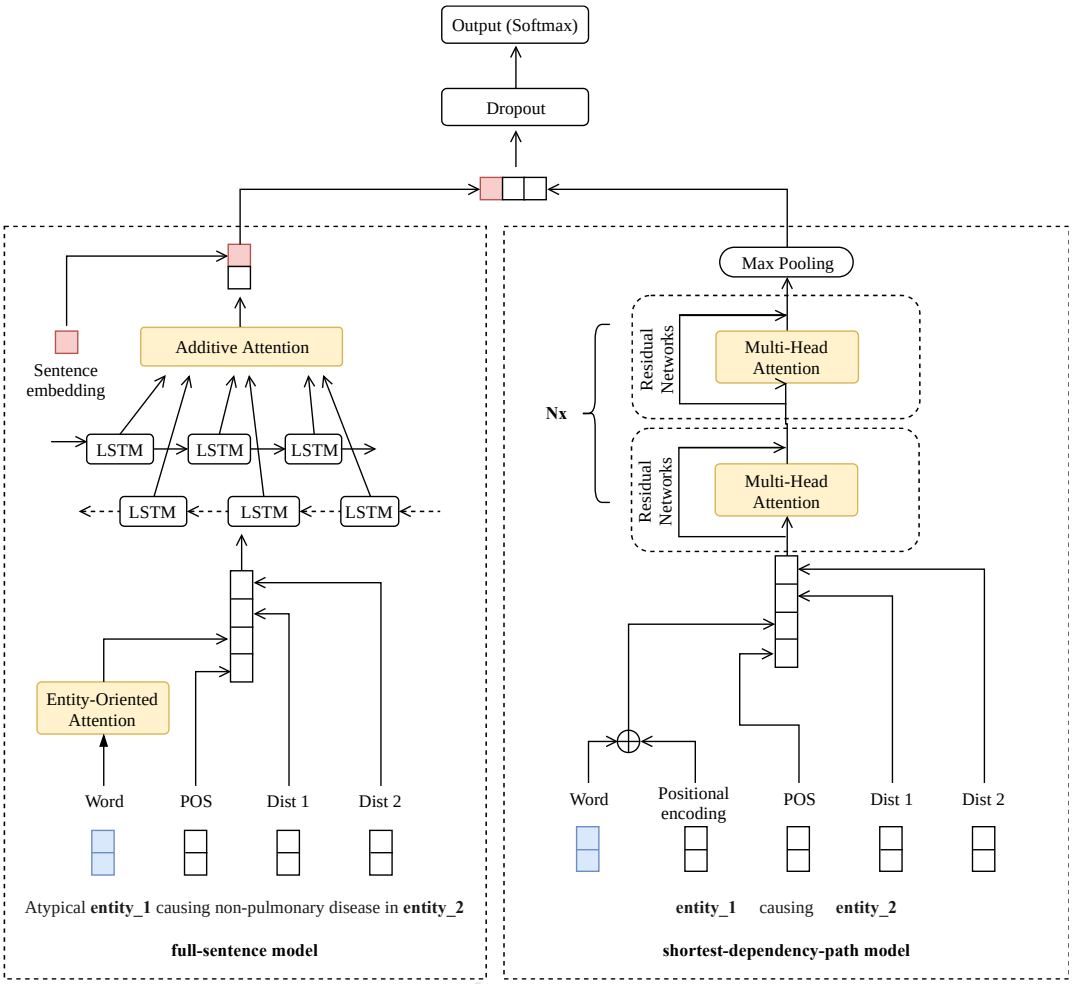

Figure 4. The overall architecture of our proposed model.
with the combined full-sentence and SDP models, together with several attention mechanisms.

In order to employ the advantages of full-sentence and SDP features, we proposed a deep learning model for the biomedical relation extraction based on full sentences and SDPs. Generally, RNNs have sequence architectures which are more powerful at capturing dependency features over long and complex sentences, while CNNs have hierarchical architectures which are good at learning the local semantic and syntactic features and more suitable for capturing the features of short and simple sentences. Figure 3 shows an overview of our proposed model which consists of full-sentence model and SDP model.

4.3.1. Full-sentence model

For the full-sentence model, we employ BLSTM to capture the global syntactic and semantic features from full sentences. Because full sentences are long and complex, we then integrate Additive [22] and Entity-Oriented [15] attention mechanisms to encourage our full-sentence model to use only parts of the input where the most relevant information is concentrated instead of the entire sentence. Furthermore, we introduce a contextual sentence embedding generated from BERT ($2.8) which can encourage our full-sentence model to understand the full sentences better.

4.3.1.1. Additive attention

The Additive attention focuses on sentence-level information. The idea of Additive attention is to consider all hidden states with different attention weights when deriving the context vector. The context vector depends on the sequence of LSTM hidden states ($\boldsymbol{h_1}, \boldsymbol{h_2}, \ldots, \boldsymbol{h_k}$). Each hidden state contains information about the whole input sequence with a strong focus on the parts surrounding the $\boldsymbol{i}$-th word. The context vector ($\boldsymbol{c_i}$) was computed as a weighted sum of these hidden states ($\boldsymbol{h_i}$) using equation (28). The attention weight ($\boldsymbol{a_i}$) of each hidden state ($\boldsymbol{h_j}$) was then computed using equation (29). The additive attention assigned a score ($\boldsymbol{a_i}$) to the pair of input at position $\boldsymbol{i}$, which was parameterized using a feed-forward network with a single hidden layer. The model was then jointly trained with other parts of the model. The attention score function is shown in equation (30), where $\boldsymbol{v_a}$ is the weight matrix to be learned.

$$c = \sum_{i=1}^{k} a_i h_i \tag{28}$$

$$a_i = softmax(score(h_i)) \tag{29}$$

$$score(h_i) = v_a^{\mathrm{T}} \tanh(h_i) \tag{30}$$

4.3.1.2. Entity-Oriented attention

Entity-Oriented attention focuses on word-level information. This attention was used to determine which words in a sentence most influence the relationship between a pair of entities. This attention mechanism was applied after our word-embedding layer to quantify the concentration of word-level information. We exploited two attention weights ($\boldsymbol{a_j}$), $\boldsymbol{j} \in [1,2]$,

which denoted the relevance degree of each word (w_i) of a sentence with respect to the j-th entity mention (e_j). The attention score (a_i^j) was calculated using the dot product operation ($\cdot$) of the current word embedding vector (u_{w_i}) and the j-th entity word-embedding vector (u_{e_j}). The score was then normalized by the dimensionality of word embedding vector (m) using equation (31). The attention weight (a_i) of word (w_i) to both entities was computed using equation (32).

$$a_i^j = softmax\left(\frac{u_{w_i} \cdot u_{e_j}}{m}\right) \tag{31}$$

$$a_i = \frac{a_i^1 + a_i^2}{2} \tag{32}$$

4.3.1.3. Contextual sentence representation

Our contextual sentence embedding was constructed by BERT model [27] which represents words based on a bi-directional approach and learns relationships between sentences. Hence, BERT representation unambiguously represents both words and sentences. However, due to the limited computational resource to pre-train BERT using our biomedical corpus the available pre-trained, general-purpose BERT was adopted and fine-tuned with the particular task (either BB or DDI task).

4.3.2. SDP model

Compared with the full sentences, SDPs are very shorter and more concise. We utilize a stack of Multi-Head attention networks [24] as our SDP model to learn valuable and concise local features. The SDP model is expected to be better at extracting high-quality features from SDPs compared with BLSTM. Since Multi-Head attention can be considered as CNNs with multiple attention linear transformations. In the experiments throughout this thesis (\$5.1.3 and \$5.2.3), we have shown that using Multi-Head attention can achieve the better performance than using CNNs so that Multi-Head attention is used as our SDP model in the final model architecture. Figure 3 shows the SDP model architecture of either CNN or Multi-Head attention.

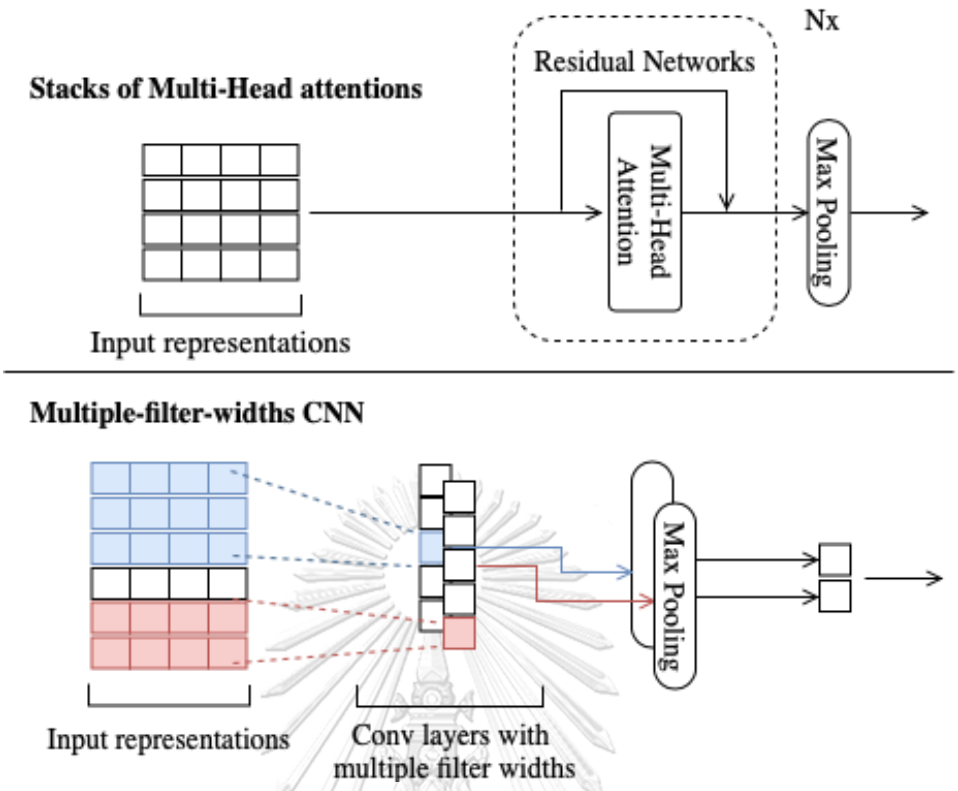

Figure 5. Illustration of the SDP model architecture.
to extract local features based on a CNN model or N stacks of Multi-Head attentions.

4.3.2.1. Multi-Head attention

The Multi-Head attention focuses on extracting local features in multiple subspaces. These authors showed that the model jointly attended to information from different representation subspaces at different positions. Instead of performing a single scale dot-product attention, [24] found to be beneficial if the queries (q), keys (k), and values (v) of the same dimension (d_k) were linearly projected h times with different learned projections. For each head, we computed the dot products ($\cdot$) of the queries of length (n) with all keys. We then divided each by $\sqrt{d_k}$ and applied the softmax function to obtain the attention score using equation (33). The context weight (c_i) on the values was computed using equation (34). To obtain Multi-Head feature representations, the context weights from h heads were concatenated using equation (35). Compared with CNNs, Multi-Head feature representation (h_{out}) uses h parallel attention mechanisms with different low-dimensional projection instead of a fixed-width convolutional filter. Inspired by the transformer network [24], stacks of Multi-Head attentions were employed in our model with residual connections and PE. Figure 5 shows the overview architecture of the CNN model and Multi-Head attentions as the SPD model.

$$score(h\_i) = softmax(\frac{q_i \cdot k_i^T}{\sqrt{d_k}}) \quad (33)$$

$$c\_i = \sum_{i=1}^{n} v_i score(h_i) \quad (34)$$

$$h_{out} = [c_1 ; c_2 ; \ldots ; c_h] \quad (35)$$

4.3.3. Output layer

The output layer used the softmax function [56] to classify the relationship between pairs of bacteria and biotope mentions. The softmax layer takes the output of Bi-LSTM for full-sentences, the output of Multi-Head attention networks for SDPs, and the sentence embedding from BERT as its inputs (Figure 4). These inputs are fed into a fully connected neural network. The softmax layer's output was the categorical probability distribution over each class type (c) (equation 36).

$$p(c|s) = softmax(W_0 \cdot s + b_0) \quad (36)$$

where W_0 and b_0 are weight parameters and s is the feature representation of sentences.

For binary classification, we used the cross-entropy cost function $J(\theta)$ as the training objective (equation 37) where y is the binary indicator (0 or 1) if the class label is correct for each predicted sentence and p is the predicted probability.

$$J(\theta) = -(y\log(p) + (1-y)\log(1-p)) \quad (37)$$

For multi-class classification, we used focal loss ($2.6.2) as the training objective.

5. EXPERIMENTS AND RESULTS

This section will discuss the experimental setups, comparison with state-of-the-art models, our contribution analysis, and error analysis of both binary and multi-class relation extraction.

As discussed above ($1.1), a large number of local minima in DL models can lead to larger parameter spaces. Evaluating a single model several times tends to result in performance convergence under different parameter initializations (or random seeds). To alleviate this problem, we reported the mean F1 score instead of only the maximum F score which used as main performance measurement in the BioNLP-ST challenge. To calculate the mean F score, we built 30 models as suggested by [57]. These models were trained using the same architecture but with different random seeds. Then, we evaluated the F score of each model on the same test set using an online evaluation service. With these F scores, we then calculated the minimum, maximum, mean, and standard deviation to assess the robustness of the model. In this study, we used the mean F score as the main evaluation metric; the maximum F score was still used to compare with other previously used models.

5.1. Binary relation extraction

For binary relation extraction, we used **bacteria-biotope (BB)** task [3] in BioNLP-ST'16 which aims to extract a critical information for studying the interaction mechanisms of the bacteria with their environment from genetic, phylogenetic and ecology perspectives.

5.1.1. Experimental setups

5.1.1.1. Training and test datasets

The dataset provided by the BB task consists of titles and abstracts from PubMed with respect to reference knowledge sources (NCBI taxonomy and OntoBiotope ontology). All entity mentions—*Bacteria*, *Habitat*, and *Geographical*—and their interactions were manually annotated from diverse-backgrounds annotators. Each bacteria-biotope pair was annotated as either a negative or positive *Lives_in* relation. The relations can be defined as inter-sentence and intra-sentence. In our study, we also followed previous studies [8, 23, 39-41] in simply excluding inter-sentence instances from the dataset. This procedure resulted in the removal of 107 and 64 annotated instances from the training data and development data, respectively. Table 4 lists the statistics of the preprocessed BB dataset used in our experiments.

Table 5. The statistics of BB dataset.

Instance	Training	Validation	Test
Positive relations	248	173	-
Negative relations	275	332	-
Total relations	523	505	532

5.1.1.2. The pre-training corpus for contextual word representations

In order to get the proposed domain-specific word embeddings (specific-PubMed ELMo), we pre-trained ELMo on the bacterial-specific abstracts downloaded from the PubMed database before November 2018. These specific abstracts contain roughly 118 million words that use all of the bacteria names that are noted in the BB dataset as keywords. An example keyword is the bacteria mention “E. coli” (Figure 1). Furthermore, we pre-trained another domain-general, biomedical, contextual, word-embedding model (random-PubMed ELMo) on randomly selected PubMed abstracts with a similar corpus size to evaluate the performance of the domain-specific model. To reduce the memory requirement of the pre-training models, we only used the words in the training, validation, and test sets to construct the vocabularies.

5.1.1.3. Hyper parameters

We used the Pytorch library [58] to implement the model and empirically tuned the hyper-parameters using 3-fold cross-validation on the training and validation data. After tuning, the dimensions of the contextual, word-embedding (ELMo), context-free word embedding, POS embedding, distance embedding, and sentence embedding (BERT) were 400, 200, 100, 300, and 768, respectively. The dimension of PE was set to either 200 or 400 for the context-free or contextual word embeddings, respectively. The hidden unit numbers of Bi-LSTM and the filter numbers of CNN were 64. The convolutional window sizes were 3, 5, and 7. For the Multi-Head attention mechanism, we used three stacks of Multi-Head attentions with respect to the residual connections; the number of heads for each stack was 2. Before the output layer, we applied a dropout rate of 0.5 to the concatenation of full-sentence, SDP, and sentence-embedding features. The mini-batch was set to 4, and a rectified linear unit (ReLU) was used as our activation functions. We set the learning rate to 0.001 for Adam optimization. Despite the underfitting problem, we used different hyper-parameters of the only full-sentences model, denoted as “full-sentence” in the section of influence of full-sentence and SDP features ($5.1.3.1). The dropout rate was set to 0.1, and the hidden unit number of LSTM was 32.

5.1.2. Binary Relation Extraction Results

Here, we discuss the overall experimental results of this proposed model on binary relation extraction. We assessed the performance of our model as follows. First, we compared our model with existing models in terms of maximum and mean F scores. Then, we evaluated the effectiveness of each contribution used by the model: feature combination between full sentences and SDP, attention mechanisms, contextual word representation, and contextual sentence representation.

5.1.2.1. Maximum F score comparison (Leaderboard)

Table 6 lists the maximum F score of our model compared with those of prior studies. In the BB task [3], each team evaluated the model on the test set using an online evaluation service. Most of the existing systems were based either on SVM or DL models. The SVM-based baseline [8] was a pipeline framework using SVMs on SDPs with an F score of 42.27%. Similarly, [10] proposed a utilized SVM with rich feature selection that yielded an F score of 55.80%. Compared with SVM-based models, DL-based models automatically learn feature representations from sentences and achieve state-of-the-art performance. For example, DUTIR [41] utilized a multiple-filter-widths CNN to achieve an F score of 45.60%. TurkuNLP [40] employed a combination of several LSTMs on the shortest dependency graphs to obtain the highest precision of 62.60% and an F score of 52.20%. BGRU-Attn [23] proposed a bidirectional GRU with the attention mechanism and biomedical-domain-oriented word embedding to achieve the highest recall of 69.82% and an F score of 57.42%. These results reveal that our proposed model achieved the best performance in the official evaluation (i.e., the highest F score: 60.77%). In contrast with the previous state-of-the-art model (BGRU-Attn), our model achieved more balanced precision (56.85%) and recall (65.38%). The results revealed that our model could leverage both full-sentence and SDP models along with contextual representations to capture the vital lexical and syntactic features of given sentences. Therefore, our model can combine the advantages of all contributions to achieve a good trade-off between moderate precision and high recall, which resulted in its superior performance in the BB corpus.

Table 6. Performance comparison (maximum F score) with existing models for the BB task. The highest scores are highlighted in bold.

Model		Precision	Recall	F score
SVM-based	TEES[8]	61.61	38.35	42.27
	VERSE [10]	51.00	61.50	55.80
DL-based	DUTIR [41]	56.60	38.20	45.60
	TurkuNLP [40]	62.30	44.80	52.10
	DET-BLSTM [39]	56.32	57.99	57.14
	BGRU-Attn [23]	48.76	69.82	57.42
	Our model	**47.93**	**71.89**	**57.51**

5.1.2.2. Mean F score comparison

Table 7 lists the results of our model compared with other models: TurkuNLP [40] and BGRU-Attn [23] based on mean F scores. Our model achieved the highest mean F score and the lowest standard deviation. This finding indicates that our model is more robust to randomness and highly consistent in its performance. Since the proposed domain-specific contextual word representation and contextual sentence representation are vector representations which can be integrated into every neural network model, we add these representations into the two reimplemented models to fairly compare them with our original model architecture, as shown in Table 7. To provide a statistically significant comparison of our model's performance, we also performed a two-sample *t*-test with the hypothesis that two populations (our model and a compared model) were equal in terms of their mean F scores (null hypothesis H_0). The results revealed that we rejected the null hypothesis with a p-value less than 0.001 (or more than 99.9% confidence). This fact implied that our model's mean F score was significantly better than that of other models.

Table 7. Performance comparison (mean F score) with existing models for the BB task. These results, from existing models, derive from the model reimplementation. The highest scores are highlighted in bold except for the standard deviation (SD). The p-value was calculated using the two-sample t-test for the difference of mean F score.

Model	F score				p-value
	Min	Max	Mean	Sd	
TurkuNLP	35.07	51.99	46.18	4.56	< 0.001
TurkuNLP + domain-specific ELMo	45.11	56.30	51.48	3.27	< 0.001
TurkuNLP + domain-specific ELMo + BERT	38.82	54.60	47.74	4.62	< 0.001
BGRU-Attn	43.05	55.54	50.24	2.70	< 0.001
BGRU-Attn + domain-specific ELMo	45.38	56.67	53.35	2.90	< 0.001
BGRU-Attn + domain-specific ELMo + BERT	46.37	52.58	49.84	1.73	< 0.001
Our model	**54.52**	**57.51**	**56.55**	**0.62**	**-**

5.1.3. Contribution analysis

In the following sections, we evaluate the effectiveness of each model contribution: combined full-sentence and SDP models, attention mechanisms, contextual word representation, and contextual sentence representation (Tables 8—11). Because test set is unknown in BB task, we cannot use highly reliable evaluation metrics such as cross-validation and bootstrapping method. To overcome the variant problem in model evaluation, each experiment used the **mean F score** for model selection and evaluation.

5.1.3.1. Influence of combined full-sentence and SDP features

Table 8 lists the mean F score of 30 DL models with different random seeds. The mean F score obtained from the experiment indicated that the use of full-sentence and SDP models together outperformed the separated models. The data in Table 8 also demonstrate that CNN achieved better performances than BLSTM when BLSTM and CNN were separately applied to the full sentences and SDPs. This result suggests that our model effectively combines the SDP and full-sentence models to extract more valuable lexical and syntactic features. These features were generated not only from two different sequences (full sentences and SDPs) but also two different neural network structures (BLSTM and CNN).

Table 8. The effectiveness of the application of full-sentence and SDP features for the BB task according to the mean F score of 30 different random seeds. All of the highest scores are highlighted in bold except for the standard deviation (SD).

Our model		F score			
Full sentence	**SDP**	**Min**	**Max**	**Mean**	**Sd**
BLSTM	-	12.82	49.93	41.22	14.49
-	CNN	37.02	51.82	43.79	3.39
BLSTM	**CNN**	**42.09**	**52.19**	**45.96**	**2.87**

5.1.3.2. Influence of attention mechanisms

After we measured the effectiveness of the full-sentence and SDP features, we additionally explored the effects of the Additive, Entity-Oriented, and Multi-Head attention mechanisms. The attention mechanisms were applied to concentrate the most relevant input representation instead of focusing on entire sentences. Table 9 lists the productiveness of each attention mechanism integrated into our full-sentence and SDP models. According to [24], Multi-Head attention networks were first proposed with the use of PE to insert valuable locality information. Because Multi-Head attention networks were employed with PE, we applied PE to CNN in order to fairly compare the effectiveness of Multi-Head attention. The use of the Additive attention mechanism improved the mean F score by 0.53%. Entity-Oriented attention improved the average F score from 49.02 to 50.24%. These results show that attention mechanisms might highlight influential words for the annotated relations and help reveal semantic relationships between each entity. This approach improved the overall performance of our model. Finally, the stacks of Multi-Head attention networks were the primary contributor to our model. The experimental results revealed that the proposed model using Multi-Head attention together with SDPs increased the mean F score by 3.18% compared with the proposed model using CNN. Our proposed model used stacks of Multi-Head attentions with residual connections instead of CNN, as shown in the overall architecture of Figure 4.

Table 9. The effectiveness of the integrated attention mechanisms for the BB task according to mean F score for 30 different random seeds. All of the highest scores are highlighted in bold except for the standard deviation (SD). The first-row results derive from the best results of previous experiments (i.e., the last row in Table 8). Note: "PE" denotes positional encoding, "Attn" denotes Additive attention, "EAttn" denotes Entity-Oriented attention, and "MAttn" denotes Multi-Head attention.

Our model		PE	F score			
Full sentence	SDPs		Min	Max	Mean	Sd
BLSTM	CNN	✗	42.09	52.19	45.96	2.87
BLSTM	CNN	✓	38.75	55.40	48.49	4.76
BLSTM + Attn	CNN	✓	42.03	56.51	49.02	3.62
BLSTM + Attn + EAttn	CNN	✓	43.14	55.72	50.24	3.72
BLSTM + Attn + EAttn	**MAttn**	✓	**46.67**	**56.7**	**53.42**	**2.51**

5.1.3.3. Influence of domain-specific contextual word representation

Table 10 lists the effectiveness of our domain-specific, contextual word representation to our model after previous contributions (combined features and attention mechanisms). The contextual word representation (ELMo) was proposed to provide word sense disambiguation across various linguistic contexts and handle out-of-vocabulary (OOV) words using a character-based approach. The results in Table 10 reveal that every ELMo model outperformed the traditional word2vec model. One possible explanation for this finding is that the ELMo model uses a character-based method to handle OOV words while word2vec initializes these OOV word representations randomly. The ELMo model can also efficiently encode different types of syntactic and semantic information about words in context and therefore improve over- all performance. The use of our proposed contextual word model with a domain-specific corpus (specific-PubMed ELMo) achieved the highest average F score of 55.91%. This score represented an improvement by 2.49%, 1.61%, and 2.10% compared with the score deriving from the use of PubMed word2vec, general- purpose ELMo, and random-PubMed ELMo, respectively. These improvements reveal the importance of taking relevant information into account when training contextual embedding vectors. We also noted that the general-purpose ELMo achieved slightly better performance compared with the random-PubMed ELMo. However, the latter was pre-trained on a biomedical-domain corpus; the size of the pre-trained corpus of the former (5.5

billion tokens) is significantly larger than that of the latter (118 million tokens), which resulted in the higher-quality word embeddings and better semantic representations.

Table 10. The effectiveness of domain-specific contextual word representation for the BB task according to mean F score of 30 different random seeds. All of the highest scores are highlighted in bold except for the standard deviation (SD). The first-row results derive from the best results of previous experiments (i.e., the last row in Table 9). Note: "PubMed word2vec" denotes the context-free word model, "general-purpose ELMo" denotes the general-purpose contextual word model, "random-PubMed ELMo" denotes the domain-general contextual word model based on 118 million randomly selected tokens from PubMed, and "specific-PubMed ELMo" denotes the domain-specific contextual word model based on 118 million bacterial-relevant tokens from PubMed.

Pre-trained word model	F score			
	Min	Max	Mean	Sd
PubMed word2vec	46.67	56.70	53.42	2.51
general-purpose ELMo	42.76	56.51	54.30	3.61
random-PubMed ELMo	38.89	57.01	53.81	3.65
specific-PubMed ELMo	**51.24**	**57.48**	**55.91**	**1.49**

5.1.3.4. Influence of contextual sentence representation

In order to use sentence embeddings as fixed features from the pre-trained BERT, [27] suggested that the best-performing method involved concatenating the feature representations from the top four 768- dimensional BLSTM hidden layers of the pre-trained model. However, we found that it was better to sum up the last four 768-dimensional hidden layers into the 768-dimension sentence embedding. This situation may have been due to the small training dataset. The addition of contextual sentence representation from the fine-tuned BERT model improved the mean F1 score by 1.68% (Table 11). The results suggest that the fine-tuned BERT model could enhance the full- sentence model to encode crucial contextual representations of long and complicated sentences.

Table 11. The effectiveness of the contextual sentence representation for the BB task according to mean F scores of 30 different random seeds. All of the highest scores are highlighted in bold except for the standard deviation (SD). The first-row results derive from the best results of previous experiments (i.e., the last row in Table 10).

Sentence representation	F score			
	Min	Max	Mean	Sd
without	51.24	57.48	55.91	1.49
with	**54.41**	**60.77**	**57.63**	**1.15**

5.1.4. Error analysis

To determine the factors that adversely affected the performance of our proposed model for binary relation extraction task, we manually analyzed the correct and incorrect predictions from a development set compared with other existing models. We found that the proposed model could detect *true negatives (TNs)* better than other reimplemented models. This finding arose mainly because full-sentence features boosted the model's ability to predict an entity pair as a false relation. For example, the sentence:

- "*Rickettsia felis* was the only **entity_1** found infecting fleas, whereas *Rickettsia bellii* was the only agent infecting ticks, but no animal or human **entity_2** was **shown** to contain rickettsial DNA."

where SDP are shown in bold, was predicted to be a false relation by our model. Other models predicted this sentence to be a true relation because of the word "shown" in the SDP.

In addition, we found that *false positives (FPs)* were generally caused by the complex and coordinate structures of full sentences. A complicated sentence and a long distance between two entities can lead to relation classification failures. Examples of these adverse effects can be seen in the following sentences which the SDPs are highlighted in bold:

- "The 210 isolates with typical LPS patterns (119 Ara- clinical, 13 Ara- soil, 70 **entity_1 entity_2**, and 8 reference National Type Culture Collection strains) also

exhibited similar immunoblot profiles against pooled sera from patients with melioidosis and hyperimmune mouse sera."

- "Testing animal and human sera by indirect immunofluorescence assay against four rickettsia antigens (*R. rickettsii*, *R. parkeri*, *R. felis*, and *R. bellii*), some opossum, **entity_2**, horse, and human sera reacted to **entity_1** with titers at least four-fold higher than to the other three rickettsial antigens."

5.1.5. Discussion

Our proposed model can take advantage of the proposed contributions in order to construct rich syntactic and semantic feature representations. Our model significantly outperforms other existing models in terms of both mean F score (57.63%; SD = 1.15%) and maximum F score (60.77%). The mechanisms that largely support stable performance include the Multi-Head attentions and domain-specific contextual word representation, which are responsible for mean F score increases of 3.18% and 2.49%, respectively. A possible advantage of Multi-Head attention compared with CNN is the ability to determine the most relevant local feature representations from multiple subspaces to the BB task based on attention weights. In addition, domain-specific contextual word representation is beneficial to the proposed model for capturing contextual embeddings from a bacterial-relevant corpus. The box-and-whisker plot in Figure 6 shows the mean F score distribution of the existing DL models and our final proposed model (blue boxes). The boxplot illustrates the performance of our model after incrementally adding each of the main contributions (grey boxes). The mean F score of each model is shown as a line. The blue boxes indicate the comparison of our final model and two reimplemented TurkuNLP and BGRU-Attn. The mean F score of our model was 57.63%, which exceeds that of the TurkuNLP and BGRU-Attn models by 11.45% and 7.41%, respectively. In other words, our proposed model generally achieves better performance in terms of both mean and maximum F scores. Furthermore, the inter-quartile range of our proposed model is much smaller than that of other DL models. This finding demonstrates that the performance of our model is more robust and suitable for real-world applications.

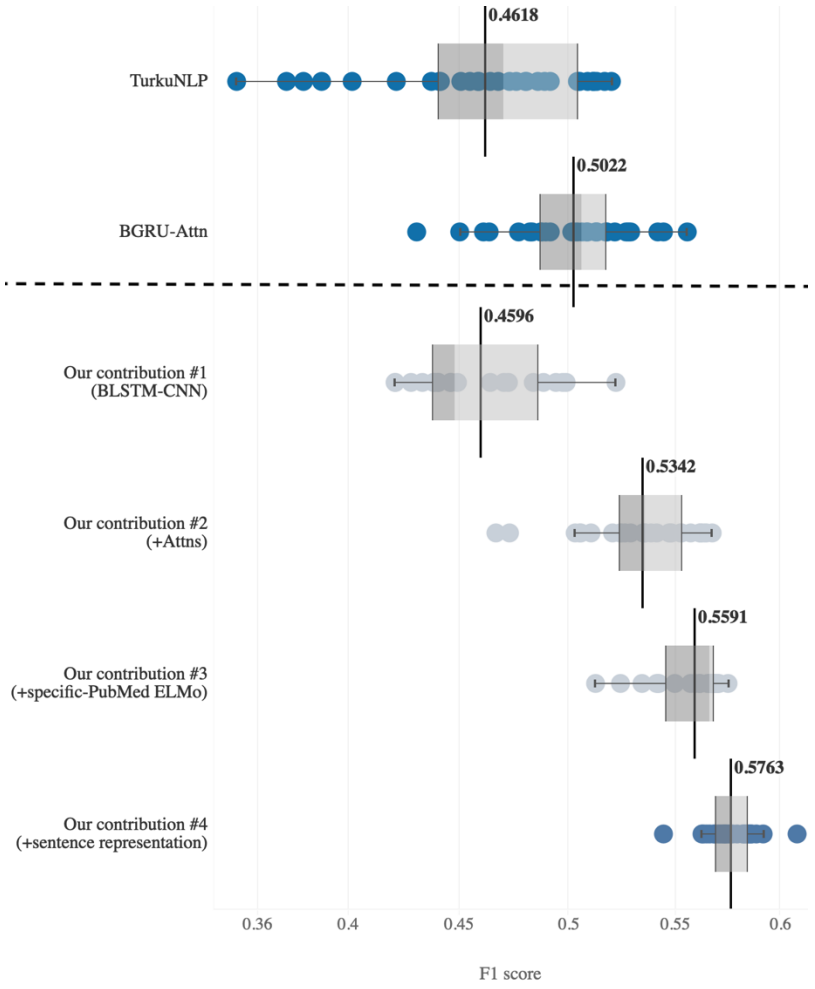

Figure 6. Box-and-whisker plot of mean F score distributions for the BB task. The comparison between our model and existing DL models is known in blue; the improvement of our model after adding each of the proposed contributions is shown in grey. Note: "Attns" denotes the use of all proposed attention mechanisms.

For binary classification problems, F score is a common metric for evaluating an overall model's performance because it conveys both precision and recall into one coherent metric. In some applications, however, it is more important to correctly classify instances than to obtain highly convergent results (i.e., high precision). On the other hand, some other applications place more emphasis on convergence rather than correctness (high recall). We experimented with using a frequency cut-off to explore how the probabilities output by the model function as a trade-off between precision and recall. Figure 7 shows the precision-recall curve (PRC) of our proposed model. When applied to real-world scenarios, users of the model are responsible for choosing the right cut-off value for their applications. For example, in semi-automated text-mining applications for knowledge management researchers never want to miss any bacteria-biotope

relations. As a result, models with a high recall will be chosen to prescreen these relations. On the other hand, automated text-mining applications for decision support systems will require more precise relations. In Figure 7, our model with the default (0.5) cut-off value achieved an F score of 60.77% with balanced 56.85% recall and 65.28% precision. With a cut-off of 0.025, our model achieved the highest recall at 70.54% with 50.11% precision and an F score of 58.59%. With this cut-off value, our model outperformed the existing highest-recall model (BGRU-Attn) by both 0.72% recall and 1.35% precision. Similarly, the line plot shown in Figure 7 shows that our model with a 0.975 cut-off achieved the highest precision (72.60%), recall (46.90%) and F score (56.99%). This model also outperformed the existing highest-precision model (TurkuNLP) by 10.30% in precision and 2.10% in recall.

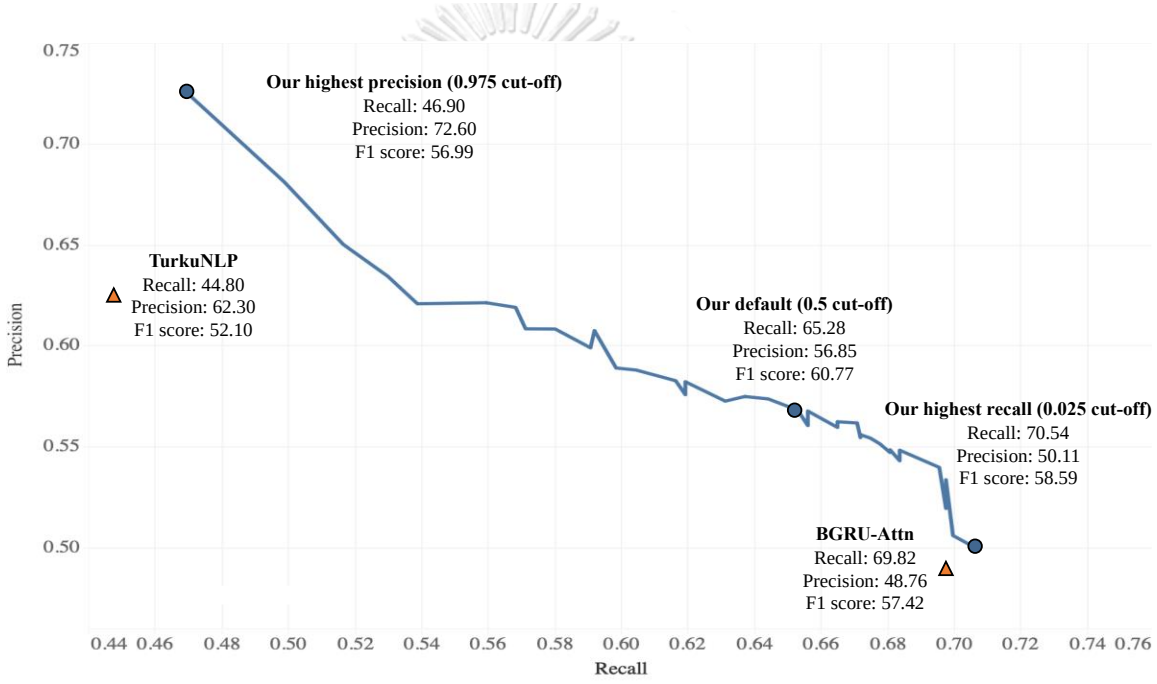

Figure 7. The precision-recall curve for our proposed model showing the trade-off between the true positive rate and the positive value for our model using different probability thresholds (cut-off values).

5.2. Multi-class relation extraction

We used **drug-drug interaction (DDI)** task [2] in BioNLP-ST'13 as our multi-class relation extraction benchmark, which concerns the extraction of drug-drug interactions that appear in biomedical literature. The automatic extraction of relevant information on DDI provides an interesting way of reducing the time spent by health care professionals on reviewing the literature.

5.2.1. Experimental setups

5.2.1.1. Training and testing datasets

For DDI task, because only training and testing sets are provided, we split 20% of training set to be validation set used for hyper-parameter selection and model evaluation. After that, we excluded inter-sentence and no-SDP samples from the data resulting in the total removal of 112 (18 inter-sentences and 94 no-SDP), 28 (5 inter-sentences and 23 no-SDP), and 30 no-SDP annotated samples from training, validation, and test data, respectively. The statistic of DDI dataset in BioNLP-ST'13 are listed in Tables 12.

Table 12. The statistics of DDI dataset.

Instance	DDI type	Training	Validation	Test
Positive relations	Mechanism	1,054	263	301
	Effect	1,349	336	359
	Advice	660	164	220
	Int	139	37	95
Negative relations	False	19,015	4,753	4,363
Total relations		22,227	5,553	5,338

Here, as shown in Table 12, "False" classes contain 19,015 and 4,753 instances, while "Effect" classes contain only 1,349 and 336 instances in training and validation sets, respectively. Thus, because DDI data is extremely imbalanced, we handled the imbalanced problem by randomly under-sampling ($2.6.1) the majority classes (False) of training and validation sets by the factor of 10. Table 13 lists the statistic of DDI dataset after random under-sampling strategy.

Table 13. The statistics of DDI dataset after random under-sampling strategy.

Instance	DDI type	Training	Validation	Test
Positive relations	Mechanism	1,054	263	301
	Effect	1,349	336	359
	Advice	660	164	220
	Int	139	37	95
Negative relations	False	1,900	475	4,363
Total relations		5,112	1,275	5,338

5.2.1.2. The pre-training corpus for contextual word representation

Because we have shown in the earlier experiments ($5.1.3.3) that the proposed domain-specific word embeddings (specific-PubMed ELMo) can perform better than other ELMo models. In the following experiments, we only focus on a comparison between specific-PubMed ELMo and PubMed word2vec. For specific-PubMed ELMo, we pre-trained ELMo on the drug-specific abstracts downloaded from the PubMed database before November 2018. These specific abstracts contain roughly 236 million words that use the word "drug" as keyword. Similar to ($5.1.1.2.), we only used the words in the training, validation, and test sets to construct the vocabularies to reduce the memory requirement of the pre-training models.

5.2.1.3. Hyper parameters

In these experiments, we empirically tuned the hyper-parameters using 5-fold cross-validation on the training and validation data. After tuning, the dimensions of the contextual, word-embedding (ELMo), context-free word embedding, POS embedding, distance embedding, and sentence embedding (BERT) were 400, 200, 100, 300, and 768, respectively. The dimension of PE was set to either 200 or 400 for the context-free or contextual word embeddings, respectively. The hidden unit numbers of Bi-LSTM and the filter numbers of CNN were 300. The convolutional window sizes were 3 and 5. For the Multi-Head attention mechanism, we used four stacks of Multi-Head attentions with respect to the residual connections; the number of heads for each stack was 1. The dropout rate of 0.4 was applied to the concatenation of full-sentence, SDP, and sentence-embedding features before the output layer. The mini-batch was set to 128, and a rectified linear unit (ReLU) was used as our activation functions. We set the learning rate to 0.001 and the weight decay parameter to 10^{-7} for RMSprop optimization.

5.2.2. Multi-class Relation Extraction Results

In this section, we explore the overall performance of our proposed model on multi-class relation extraction. We compared our model with other existing models in terms of maximum and mean F scores (micro). After that, we also analyzed the effectiveness of each main contribution, including handling imbalanced data techniques, and error generated by our model.

5.2.2.1. Maximum F score comparison (Leaderboard)

Table 14 lists the maximum F score (micro) of our proposed model compared with other previous models for the DDI task. Similar to other biomedical relation tasks, most of the existing systems were based either on SVM or DL models. The SVM-based baseline for the DDI task was Uturku [9] which used SVM with information from domain resources and dependency features to achieve F score of 42.3%. FBK-irst [7] combined different three kernels of SVMs to rank first in BioNLP-ST'13 with F score of 65.1% and RAIHANI [6] utilized SVM with multiple rules and features to get 71.1% F score and be the best-performing model among the SVM-based models for the DDI task. As shown in Table 14, DL-based models can automatically learn high-quality features and slightly outperform SVM-based models. For example, Liu-CNN [11] used CNN model with syntactic features to get 69.8 F score. TEES-CNN [38] utilized multiple-filter-widths CNN model with different embeddings to achieve F score of 73.5% and the highest precision of 80.5%. Recursive NN [50] used recursive neural network models with position features, subtree containment features, and ensemble methods to improve the models' performance and achieved 73.5% F score. Attn-BLSTM [15] utilized BLSTM with Entity-Oriented attention to achieve the overall F score of 77.3% with the highest F score of 85.1% and 57.7% on Advice and Int types, respectively. In Table 14, the results reveal that our proposed model can achieve the highest overall F score of 80.3% and the highest recall of 83.0%. Our proposed model can also achieve the highest F score on both Effect and Mechanism types which are 84.4% and 81.0%, respectively. These results show that our model can combine the advantages of every contribution to achieve the highest recall and F score resulting in its superior performance for the DDI task.

Table 14. Performance comparison (maximum F score) with existing models for DDI task. The highest scores are highlighted in bold.

Model		F score (micro) on each DDI type				Overall performance		
		Effect	Mechanism	Advice	Int	Precision	Recall	F score (micro)
SVM-based	Uturku [9]	60.0	58.2	63.0	50.7	73.2	49.9	59.4
	FBK-irst [7]	62.8	67.9	69.2	54.7	64.6	65.6	65.1
	RAIHANI [6]	69.6	73.6	77.4	52.4	73.7	68.7	71.1
DL-based	Liu-CNN [11]	69.3	70.2	77.8	48.4	75.7	64.7	69.8
	MCCNN [12]	68.2	72.2	78.2	51.0	76.0	65.3	70.2
	TEES-CNN [38]	-	-	-	-	**80.5**	67.6	73.5
	Joint AB-LSTM [16]	65.5	76.3	80.3	44.1	74.5	65.0	71.5
	Char-RNNs [18]	-	-	-	-	80.0	65.9	72.1
	Hierarchy RNN [17]	71.8	74.0	80.3	54.3	74.1	71.8	72.9
	Recursive NN [50]	-	-	-	-	77.8	69.6	73.5
	Attn-BLSTM [15]	76.6	77.5	**85.1**	**57.7**	78.4	76.8	77.3
	Our model	**84.4**	**81.0**	82.5	57.1	77.6	**83.0**	**80.3**

5.2.2.2. Mean F score comparison

Due to that face that we have failed to reimplement the state-of-the-art model for DDI task which is Attn-BLSTM [15], we cannot report its performance in terms of mean and standard deviation F scores, only maximum F score provided. It also is impossible for us to study the robustness to randomness in its performance. We instead compared our model's average performance with the state-of-the-art model's maximum performance. In Table 15, the results show that our proposed model can achieve mean F score of 77.7%, which is still more than the Attn-BLSTM's maximum performance of 77.3%.

Table 15. Performance comparison (mean F score) with existing models for DDI task.

Model	F score				p-value
	Min	Max	Mean	Sd	
Attn-BLSTM [15]	-	77.3	-	-	-
Our model	**73.4**	**80.2**	**77.7**	**1.5**	**-**

5.2.3. Contribution analysis

Here, we evaluate the effectiveness of each model contribution: combined full-sentence and SDP models, attention mechanisms, contextual word representation, and contextual sentence representation (Tables 16—20) for DDI task. Unlike BB task, the test set is now provided in the DDI task. Thus, we can combine training, validation, and test sets to compute **5-fold cross-validation** as our model selection and evaluation.

5.2.3.1. Influence of combined full-sentence and SDP features

Table 16 lists the 5-fold cross-validation of F score obtained from the experiment to analyze the effectiveness of the use of full-sentence and SDP models. The results show that the use of combined both features can get 67.7% average F score of the five folds which outperform both the use of only full-sentence and the use of only SDP feature. The results also suggest that our combined full-sentence and SDP models can effectively extract high-quality syntactic features from two different sequences with two different neural networks.

Table 16. The effectiveness of the application of full-sentence and SDP features for the DDI task according to 5-fold cross-validation. All of the highest scores are highlighted in bold.

Our model		F score					
Full sentence	**SDP**	**#1**	**#2**	**#3**	**#4**	**#5**	**Avg.**
BLSTM	-	28.7	27.0	23.1	23.4	24.0	25.2
-	CNN	62.1	60.4	66.6	67.4	69.5	65.2
BLSTM	**CNN**	**64.5**	**64.5**	**67.0**	**70.1**	**72.2**	**67.7**

5.2.3.2. Influence of attention mechanisms

In this experiment analysis, we explored the effects of each proposed attention mechanisms: Additive, Entity-Oriented, and Multi-Head attention networks. These attention mechanisms were used to focus on the most relevant input features to the task's outputs. Table 17 lists the effectiveness of each attention network adapted into our full-sentence and SDP models using 5-fold cross-validation. The experimental results show that the full-sentence model with Additive attention can improve average cross-validation results from 67.7% F score to 68.5% F score, which is considered to outperform the full-sentence model with Entity-Oriented attention by 0.5% F score. Because Additive and Entity-Oriented attention networks are supposed to extract different input features, the results show that the use of both these attention mechanisms together can get 70.2% F score, which can perform better than the use of only one of them. In addition, the results from Table 17 also show that the SDP model using Multi-Head attention can increase the model's performance to 74.5% F score, which is more than using CNN by 4.3% F score.

Table 17. The effectiveness of the integrated attention mechanisms for the DDI task according to 5-fold cross-validation. All of the highest scores are highlighted in bold. The first-row results derive from the best results of previous experiments (i.e., the last row in Table 16). Note: "PE" denotes positional encoding, "Attn" denotes Additive attention, "EAttn" denotes Entity-Oriented attention, and "MAttn" denotes Multi-Head attention.

Our model		PE	F score					
Fulls	**SDPs**		**#1**	**#2**	**#3**	**#4**	**#5**	**Avg.**
BLSTM	CNN	✗	64.5	64.5	67.0	70.1	72.2	67.7
BLSTM + Attn	CNN	✓	59.8	63.8	70.8	73.0	75.1	68.5
BLSTM + EAttn	CNN	✓	63.8	63.8	69.8	70.2	72.5	68.0
BLSTM + Attn + EAttn	CNN	✓	67.7	65.8	72.0	71.9	73.7	70.2
BLSTM + Attn + EAttn	**MAttn**	✓	**68.6**	**74.1**	**76.2**	**76.1**	**77.5**	**74.5**

5.2.3.3. Influence of domain-specific contextual word representation

Although the context-free word model was pre-trained on entire PubMed corpus, the results in Table 18 show that our proposed domain-specific contextual word model can outperform the context-free word model with the cross-validation result of 80.0% F score. The contextual model, which was proposed to solve word sense ambiguation problem, can encode the higher-quality word embeddings and better semantic representations than the context-free model.

Table 18. The effectiveness of domain-specific contextual word representation for the DDI task according to 5-fold cross-validation. All of the highest scores are highlighted in bold. The first-row results derive from the best results of previous experiments (i.e., the last row in Table 17). Note: "PubMed word2vec" denotes the context-free word model and "specific-PubMed ELMo" denotes the domain-specific contextual word model based on 236 million drug-relevant tokens from PubMed.

Pre-trained word model	F score					
	#1	**#2**	**#3**	**#4**	**#5**	**Avg.**
PubMed word2vec	68.6	74.1	76.2	76.1	77.5	74.5
Specific-PubMed ELMo	**72.9**	**78.8**	**81.0**	**82.4**	**85.1**	**80.0**

5.2.3.4. Influence of contextual sentence representation

Similar to the earlier experiments on binary relation extraction task (§5.1.3.4), we sum up the last four 768-dimensional hidden layers from BERT into a 768-dimension sentence embedding. This sentence embedding can be considered as fixed features. In Table 19, the results suggest that our proposed model with the sentence representation from the fine-tuned BERT can achieve average F score of 90.9%, which outperforms our model without sentence representation by 10.9% F score. The results also suggest that our sentence representation can effectively support our full-sentence model to better encode long and complicated sentences into very high-quality features.

Table 19. The effectiveness of the contextual sentence representation for the DDI task according to 5-fold cross-validation. All of the highest scores are highlighted in bold. The first-row results derive from the best results of previous experiments (i.e., the last row in Table 18).

Sentence representation	F score					
	#1	#2	#3	#4	#5	Avg.
without	72.9	78.8	81.0	82.4	85.1	80.0
with	**86.9**	**93.3**	**92.8**	**92.0**	**89.3**	**90.9**

5.2.3.5. Influence of focal loss

Table 20 lists the effectiveness of focal loss to our model with all proposed contributions in this study. The results reveal that, due to our highly imbalanced data, the focal loss can improve the model's performance by 0.5% F score compared with the cross-entropy loss. A possible reason for this improvement is that the focal loss can solve the problem of class imbalance by making the loss implicitly focus in those problematic classes.

Table 20. The effectiveness of the focal loss for the DDI task according to 5-fold cross-validation. All of the highest scores are highlighted in bold. The first-row results derive from the best results of previous experiments (i.e., the last row in Table 19).

Focal loss	F score					
	#1	#2	#3	#4	#5	Avg.
without	86.9	93.3	92.8	**92.0**	89.3	90.9
with	**87.2**	**94.4**	**94.2**	91.7	**89.4**	**91.4**

5.2.4. Error analysis

In this section, we manually analyzed the correct and incorrect predictions from a development set to determine the factors that adversely affected the performance of our proposed model to multi-class relation extraction task. From the experiment analysis, we have found that our proposed model tends to predict negative relations correctly, true negatives (TNs), because their SDP lengths are shorter than the average. In addition, similar to the previous section (\$5.1.4), we have found that false positives (FPs) are mainly caused by the complex and coordinate structures

of full sentences. Examples of these effects can be seen in the following sentence which the SDPs are highlighted in bold:

- “a multiple dose drug-drug interaction study demonstrated that prot0 approximately doubled prot1 auc0 - . since prot2 is partially metabolized by cyp3a and prot3 le is known to be a strong inhibitor of cytochrome p450 zqwnum an enzyme , care should be taken while dosing prot4 with prot5 and other strong p450 zqwnum inhibitors including prot6 , **entity_1** , prot8 , prot9 , prot10 , prot11 , **entity_2** , prot13 , prot14 or prot15 .”

5.2.5. Discussion

The model architecture with all of our proposed contributions can extract very high-quality syntactic and semantic features to define biomedical relationships. Our proposed model outperforms other existing models with the highest F score (80.3%) and the highest recall (83.0%). Unfortunately, we cannot reimplement the state-of-the-art model (Attn-BLSTM) for the DDI task to get the same results as provided in its paper. To provide the comparison with low-bias evaluation metric, we instead compared mean F score of our model with maximum F score of the state-of-the-art model. As a result, our proposed model with mean F score of 77.5% outperforms the state-of-the-art model with maximum F score of 77.3%. From the contribution analysis, we found that the main mechanisms that largely support our reliable performance (5-fold cross-validation) are contextual sentence representation and domain-specific contextual word representation. These mechanisms are responsible for F score increases of 10.9% and 5.5%, respectively. Similar to the previous experimental results, domain-specific contextual word representation might good at capturing the contextual and relevant embeddings from drug-oriented corpus. In addition, the contextual sentence representation might extract the high-quality sentence features from long and complex full sentences from the fine-tuned language model.

6. CONCLUSIONS

We have presented a DL extraction model for the biomedical relation extraction tasks based on a combination of full-sentence and SDP models that integrate various attention mechanisms. Furthermore, we introduced a domain-specific contextual word-embedding model based on the large bacteria-relevant corpus and fine-tuned contextual sentence representation. These embeddings encouraged the model to effectively learn high-quality feature representations from the pre-trained language modeling. We compared our proposed model with the existing models based on maximum and mean F scores for both BB and DDI tasks. We also explored and analyzed the effectiveness of each proposed contribution to our proposed model. The experimental results demonstrated that our model effectively integrated all proposed contributions. The results showed that we could improve the performance of relation extraction to achieve the highest maximum F scores and significantly outperform other existing models for both BB and DDI tasks (60.77% and 80.3%, respectively). Additionally, compared with low-bias performance, our model is more robust to real-world applications than the previous models.

Despite our model exhibiting the best performance on both relation extraction tasks, some challenges remain. One of the most important limitations of our model is that it cannot extract inter-sentence relations between the entities. Hence, all true inter-sentence relations become false negatives. Inter-sentence relation extraction is much more challenging because it requires a more nuanced understanding of language to classify relations between entities in different sentences and clauses characterized by complex syntax. Because the size of common biomedical dataset in the challenge is quite small, it is very difficult for DL models to learn sufficient high-quality features for the target tasks. However, this challenging task is left for future work. Furthermore, there is a large repertoire of biomedical literature and domain resources that are freely accessible and can be used as unlabeled data for semi-supervised learning and transfer learning methods [59-62].

REFERENCES

1. Cohen, K.B. and L. Hunter, *Getting started in text mining*, in *PLoS Computational Biology*. 2008.
2. Segura-Bedmar, I., et al., *SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013)*, in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. 2013.
3. Deléger, L., et al., *Overview of the Bacteria Biotope Task at BioNLP Shared Task 2016*, in *Proceedings of the 4th BioNLP Shared Task Workshop*. 2016.
4. Bossy, R., et al., *Bionlp shared task 2011: bacteria biotope.* Proceedings of the BioNLP Workshop at ACL Conference, 2011.
5. Bossy, R., et al., *BioNLP shared Task 2013--An Overview of the Bacteria Biotope Task.* Proceedings of the BioNLP Workshop at ACL Conference, 2013.
6. Raihani, A. and N. Laachfoubi, *Extracting drug-drug interactions from biomedical text using a feature-based kernel approach.* Journal of Theoretical and Applied Information Technology, 2016.
7. Chowdhury, F.M. and A. Lavelli, *FBK-irst : A Multi-Phase Kernel Based Approach for Drug-Drug Interaction Detection and Classification that Exploits Linguistic Information*, in *Seventh International Workshop on Semantic Evaluation*. 2013.
8. Björne, J. and T. Salakoski, *TEES 2.1: Automated annotation scheme learning in the BioNLP 2013 Shared Task.* Proceedings of the BioNLP Shared Task 2013 ..., 2013.
9. Björne, J., S. Kaewphan, and T. Salakoski, *UTurku : Drug Named Entity Recognition and Drug-Drug Interaction Extraction Using SVM Classification and Domain Knowledge.* Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), 2013. **2**: p. 651-659.
10. Lever, J. and S.J.M. Jones, *VERSE : Event and Relation Extraction in the BioNLP*

2016 Shared Task. Proceedings of the 4th BioNLP Shared Task Workshop, 2016: p. 42-49.

11. Liu, S., et al., *Drug-Drug Interaction Extraction via Convolutional Neural Networks.* Computational and Mathematical Methods in Medicine, 2016.
12. C., Q., et al., *Multichannel convolutional neural network for biological relation extraction.* BioMed Research International, 2016. **2016**.
13. Zeng, D., et al., *Relation Classification via Convolutional Deep Neural Network*, in *Proceedings of the 25th International Conference on Computational Linguistics (COLING'14)*. 2014.
14. Zhao, Z., et al., *A protein-protein interaction extraction approach based on deep neural network.* International Journal of Data Mining and Bioinformatics, 2016.
15. Zheng, W., et al., *An attention-based effective neural model for drug-drug interactions extraction.* BMC Bioinformatics, 2017.
16. Sahu, S.K. and A. Anand, *Drug-drug interaction extraction from biomedical texts using long short-term memory network.* Journal of Biomedical Informatics, 2018.
17. Zhang, Y., et al., *Drug-drug interaction extraction via hierarchical RNNs on sequence and shortest dependency paths.* Bioinformatics, 2018.
18. Kavuluru, R., A. Rios, and T. Tran, *Extracting Drug-Drug Interactions with Word and Character-Level Recurrent Neural Networks*, in *Proceedings - 2017 IEEE International Conference on Healthcare Informatics, ICHI 2017*. 2017.
19. Dey, R. and F.M. Salemt, *Gate-variants of Gated Recurrent Unit (GRU) neural networks*, in *Midwest Symposium on Circuits and Systems*. 2017.
20. Greff, K., et al., *LSTM: A Search Space Odyssey.* IEEE Transactions on Neural Networks and Learning Systems, 2017.
21. Luong, M.-T., H. Pham, and C.D. Manning, *Effective Approaches to Attention-based Neural Machine Translation.* 2015.
22. Bahdanau, D., K. Cho, and Y. Bengio, *NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE.* arXiv:1409.0473 [cs, stat], 2014.
23. Li, L., et al., *Biomedical event extraction based on GRU integrating attention mechanism.* BMC Bioinformatics, 2018.

24. Vaswani, A., et al., *Attention Is All You Need.* 2017.
25. Mikolov, T., et al., *Distributed Respresentation of Words and Pharses and their Compositionality.* CrossRef Listing of Deleted DOIs, 2000.
26. Pennington, J., R. Socher, and C. Manning, *Glove: Global Vectors for Word Representation*, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014.
27. Devlin, J., et al., *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* 2018.
28. Peters, M.E., et al., *Deep contextualized word representations.* 2018.
29. Wang, Y., et al., *A comparison of word embeddings for the biomedical natural language processing.* Journal of Biomedical Informatics, 2018.
30. Huang, C.C. and Z. Lu, *Community challenges in biomedical text mining over 10 years: Success, failure and the future.* Briefings in Bioinformatics, 2016.
31. Schuster, M. and K.K. Paliwal, *Bidirectional recurrent neural networks.* IEEE Transactions on Signal Processing, 1997.
32. Graves, A., S. Fernández, and J. Schmidhuber, *Bidirectional LSTM networks for improved phoneme classification and recognition.* International Conference on Artificial Neural Networks, 2005.
33. Kim, Y., *Convolutional Neural Networks for Sentence Classification.* 2014.
34. Wang, Q., et al., *Overview of the Interactive Task in BioCreative V*. 2015.
35. Shahab, E., *A Short Survey of Biomedical Relation Extraction Techniques.* CoRR, 2017. **abs/1707.0**.
36. Chen, E.S., et al., *Automated acquisition of disease drug knowledge from biomedical and clinical documents: an initial study.* Journal of the American Medical Informatics Association : JAMIA, 2008. **15**: p. 87-98.
37. Saric, J., et al., *Extraction of regulatory gene/protein networks from Medline.* Bioinformatics (Oxford, England), 2006. **22**: p. 645-650.
38. Bj, J., *Biomedical Event Extraction Using Convolutional Neural Networks and Dependency Parsing.* 2018: p. 1-11.
39. Li, L., et al., *Biomedical event extraction via Long Short Term Memory networks along dynamic extended tree*, in *Proceedings - 2016 IEEE International*

Conference on Bioinformatics and Biomedicine, BIBM 2016. 2017.

40. Hazim, B., et al., *Deep Learning with Minimal Training Data: TurkuNLP Entry in the BioNLP Shared Task 2016.* Frontiers in Plant Science, 2016.
41. Li, H., et al., *DUTIR in BioNLP-ST 2016 : Utilizing Convolutional Network and Distributed Representation to Extract Complicate Relations.* 2016: p. 93-100.
42. Verleysen, M. and D. François, *The Curse of Dimensionality in Data Mining.* Lecture Notes in Computer Science, 2005.
43. Joulin, A.G., Edouard; Bojanowski, Piotr; Mikolov, Tomas, *Bag of Tricks for Efficient Text Classification – Facebook Research.* Facebook Research, 2017.
44. Bojanowski, P., et al., *Enriching Word Vectors with Subword Information.* 2017.
45. Pyysalo, S., et al., *Distributional Semantics Resources for Biomedical Text Processing*, in *Languages in Biology and Medicine*. 2013.
46. Kubat, M. and S. Matwin, *Addressing the Curse of Imbalanced Training Sets: One-Sided Selection.* Proceedings of the Fourteenth International Conference on Machine Learning, 1997: p. 179-186.
47. Lin, T.Y., et al., *Focal Loss for Dense Object Detection.* Proceedings of the IEEE International Conference on Computer Vision, 2017. **2017-Octob**: p. 2999-3007.
48. Tran, G.S., et al., *Improving Accuracy of Lung Nodule Classification Using Deep Learning with Focal Loss.* Journal of Healthcare Engineering, 2019. **2019**.
49. De Boer, P.T., et al., *A tutorial on the cross-entropy method.* Annals of Operations Research, 2005. **134**: p. 19-67.
50. Lim, S., K. Lee, and J. Kang, *Drug drug interaction extraction from the literature using a recursive neural network.* PLoS ONE, 2018. **13**: p. 1-17.
51. Charniak, E. and M. Johnson, *Coarse-to-fine n-best parsing and MaxEnt discriminative reranking*, in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*. 2005.
52. de Marneffe, M.-C., B. Maccartney, and C.D. Manning, *Generating typed dependency parses from phrase structure parses*, in *Proc of 5th Int Conf Lang Resour Eval (LREC 2006)*. 2006.
53. Stevenson, M. and Y. Guo, *Disambiguation in the biomedical domain: The role of ambiguity type.* Journal of Biomedical Informatics, 2010.

54. Jiang, Z., et al., *Training word embeddings for deep learning in biomedical text mining tasks*, in *Proceedings - 2015 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2015*. 2015.

55. Zamani, H. and W.B. Croft, *Relevance-based Word Embedding.* 2017: p. 505-514.

56. Bridle, J.S., *Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition*, in *Neurocomputing*, F.F. Soulié and J. Hérault, Editors. 1990, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 227-236.

57. Colas, C., O. Sigaud, and P.-Y. Oudeyer, *How Many Random Seeds? Statistical Power Analysis in Deep Reinforcement Learning Experiments.* 2018: p. 1-20.

58. Adam Paszke, S.G., Soumith Chintala, Gregory Chanan, Edward Yang, *Automatic differentiation in PyTorch.* Conference on Neural Information Processing Systems, 2017.

59. Thang Luong, C., Quoc Le, *Combine Semi-Supervised Learning with Transfer Learning.* Emnlp, 2018. **77**: p. 76-82.

60. Peters, M.E., et al., *Dissecting Contextual Word Embeddings: Architecture and Representation.* 2018: p. 1499-1509.

61. Joshi, M., et al., *pair2vec: Compositional Word-Pair Embeddings for Cross-Sentence Inference.* 2018.

62. Clark, K., et al., *Semi-Supervised Sequence Modeling with Cross-View Training*, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, Association for Computational Linguistics: Brussels, Belgium. p. 1914-1925.

VITA

NAME	Amarin Jettakul
DATE OF BIRTH	9 December 1995
PLACE OF BIRTH	Nakhonratchasima
INSTITUTIONS ATTENDED	Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University
HOME ADDRESS	99/9 M.4, Bangkhuntaek, Muang Samutsongkhram, Samutsongkhram, 75000
PUBLICATION	Jettakul, A., Thamjarat, C., Liaowongphuthorn, K., Udomcharoenchaikit, C., Vateekul, P., & Boonkwan, P. (2018). A Comparative Study on Various Deep Learning Techniques for Thai NLP Lexical and Syntactic Tasks on Noisy Data. 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), 1-6.