NEURAL NAMED ENTITY RECOGNITION AND TEMPORAL RELATION EXTRACTION

A thesis submitted to The University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Science and Engineering

2020

Meizhi Ju

Department of Computer Science

Contents

Word Count: 30,446

List of Tables

List of Figures

Abstract

Neural Named Entity Recognition and Temporal Relation Extraction

Meizhi Ju

A thesis submitted to The University of Manchester
for the degree of Doctor of Philosophy, 2020

Automatically identifying information of interest from texts is one of the most difficult challenges. One crucial step towards information extraction is named entity recognition, where many entities are embedded in other entities (i.e., nested entities). Nested entities contain rich fine-grained information, which is essential in understanding texts. However, most work ignored nested entity recognition though they are common in many domains. In addition to the semantic information expressed in named entities, temporal information conveyed by named entities is another important dimension in understanding texts. Temporally classifying the relations (e.g., before) between entities is known as temporal relation extraction, which is required in many tasks such as text summarisation.

The thesis is the first comprehensive research focusing on nested entity recognition for information extraction using neural network methods. In this research, we describe our work on (1) neural nested entity recognition; (2) evaluation on different domains of corpora; (3) task-specific evaluation including (a) neuroscience entity extraction; (b) screening reference documents; (c) extraction of medication and adverse drug information; (d) and extraction of chronic obstructive pulmonary disease phenotypes. In addition to nested entity recognition, we further investigate neural temporal relation extraction, which focuses on the extraction of both intra-sentence and inter-sentence temporal relations.

Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright

Acknowledgements

Undertaking this three-year PhD has been a truly life-challenging experience for me and it would not have been possible to do without the support and guidance that I received from many people. I would like to thank six groups of people: my supervisor, thesis committee, research collaborators, funding agencies, friends and my family.

My Supervisors

I would like to first say a very big thank you to my supervisor Prof. Sophia Ananiadou for the continuous support of my three-year PhD study and related research, for her patience, motivation, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor for my PhD journey.

I am also very grateful to my co-suprevisor Dr Riza Theresa Batista-Navarro, who has given me support and deep understanding like family especially at the beginning of my PhD study, making me brave and confident towards difficulties in my research and life.

Research Collaborators

My sincere thanks also go to Prof. Junichi Tsujii, who provided me an opportunity to join Artificial Intelligence Research Center as intern, and gave me access to the center and research facilities. Without his precious support it would not be possible to conduct this research.

I gratefully appreciate the discussions and technical help received throughout the collaborative work undertaken with Prof. Makoto Miwa, during the first phrase of my field work – thank you to for making those neural networks all the more magical and interesting. You provided me with the tools, reading materials

and even coding instructions for the right way and successfully completed my work.

Thesis Committee

Many thanks to my thesis committee: Prof. Goran Nenadic and Prof. Pierre Zweigenbaum, for not only their invaluable comments and suggestions, but also for the hard question which incented me to widen my research form various perspectives.

Funding Agencies

I gratefully acknowledge the funding Doctoral President's Scholar Award received from the University of Manchester for the financial support that I otherwise would not have been able to develop my scientific discoveries.

Friends

My deep appreciation goes to the all the people from NaCTeM for their daily accompany for lunch and funny discussions on various interesting topics. I thank Piotr and Austin, who were always patient to instruct me how to use gym equipment and helped me develop interest in sports. I also thank Thy, Maolin and Fenia for their sweet gifts towards important moments such as birthdays and oral defense. A very special thank you for Matt, Nhung and Paul, whom I had very happy collaborations with, for the detailed writing suggestion and oral English correction. I also thank Chryssa for her very detailed suggestion in dealing with thesis and other tough situation.

Thanks are also due to my friends, who were of great support in deliberating over our problems and findings in different areas, as well as providing happy distraction to rest my mind outside of my research. Many thanks to Fang, Hiris, and other friends for their thoughtful accompany for delicious food, interesting movies and the surprising gifts for holidays.

My Family

I am indebted to my parents for their wise counsel and sympathetic ear. You are always there for me. I would not where I am today without their love and encouragement.

Abbreviations

ACE Automatic Content Extraction

ADE adverse drug event

ADEs adverse drug events

bc broadcast conversation

BERT bidirectional encoder representations from transformers

bn broadcast news

BOW bag-of-words

C comparator

CBOW continuous bag-of-words

CCG chunking and combinatory categorial grammar

CE category error

cf cluster features

CNN convolutional neural network

COPD chronic obstructive pulmonary disease

CRF conditional random field

CRFs conditional random fields

DCT document creation time

df dictionary features

EHR electronic health record

EHRs electronic health records

FAC facility

FN false negatives

FOFE fixed-size ordinally-forgetting encoding

FP false positives

GCL global context layer

GEO geo-political entity

GPT generative pretrained transformer

GRU gated recurrent unit network

GRUs gated recurrent unit networks

HMM hidden markov model

I intervention

i2b2 Informatics for Integrating Biology and the Bedside

ID-CNNs iterated dilated convolutional neural networks

IE information extraction

IREX information retrieval and information extraction

LBCWLE layered-BiLSTM-CRF w/o layered out-of-entities

LBCWLL layered-BiLSTM-CRF w/o layered LSTM

LOC location

LSTM long short-term memory network

LSTMs long short-term memory networks

MeSH Medical Subject Headings

MIMIC III Medical Information Mart for Intensive Care III

ML machine learning

MNLP maximum normalized log-probability

MTL multi-task learning

MUC-6 Sixth Message Understanding Conference

NE named entity

NER named entity recognition

NLP natural language processing

NN neural network

NNs neural networks

nw newswire

O outcomes

ORG organisation

P patient/population

PER person

POS part-of-speech

RNN recurrent neural network

SDPs shortest dependency paths

SE span error

SemEval Semantic Evaluation

SVM support vector machine

TBD TimeBank-Dense

TEE timex extraction

TempEval temporal evaluation

TERQAS Time and Event Recognition for Question Answering Systems

THYME Temporal Histories of Your Medical Event

TIE temporal information extraction

TimeML time mark-up language

timexes time expressions

TP true positives

TRE temporal relation extraction

UMLS Unified Medical Language System

UNK unknown

VAERS Vaccine Adverse Event Reporting System

VEH vehicle

WEA weapon

wl weblog

ws word shape

Chapter 1

Introduction

1.1 Motivation

Texts written in human (natural) languages such as scientific articles are a rich source for information extraction (IE), which targets the automatic extraction of structured information from unstructured texts. Computational techniques such as natural language processing (NLP), machine learning (ML) have been widely used to automatically process and analyze massive natural language data, saving time and human effort. With these techniques, one of the most important steps towards IE is to extract entities of interest (e.g., organisations, places, people) from texts. Such a task is known as named entity recognition (NER), which serves as the first step in many NLP downstream tasks such as entity linking (Gupta et al., 2017), relation extraction (Miwa and Bansal, 2016; Christopoulou et al., 2018), event extraction (Feng et al., 2016) and co-reference resolution (Fragkou, 2017; Stone and Arora, 2017; Trieu et al., 2018). As a result, NER has been receiving constant attention from the community (Grishman and Sundheim, 1996; Scheffer et al., 2001; Mohit and Hwa, 2005; Sasaki et al., 2008; Tsuruoka et al., 2008; Lample et al., 2016; Shardlow et al., 2018; Ju et al., 2019b).

Due to the properties of natural language, many entities are embedded in other entities, which are referred to as nested entities. For example, Figure 1.1 contains three inner entities, which are nested within the outermost entity *"Reporter Carl Dinnon of Britain's ITN"*.

Nested entities are quite common in many domains. The GENIA corpus (Kim et al., 2003), one of the most popular biomedical corpora, contains 16.7% of nested entities (Gu, 2006). In the general domain, around 24% of entities in

$$\underbrace{\underbrace{\text{Reporter}}_{\textbf{PER}}\ \text{Carl Dinnon of}\ \underbrace{\underbrace{\text{Britain}}_{\textbf{GPE}}\text{'s ITN}}_{\textbf{ORG}}}_{\textbf{PER}}\ \text{fielded this report.}$$

Figure 1.1: A sentence from ACE2005 (Walker et al., 2006) containing nested entities. "PER", "GPE" and "ORG" represent person, geo-political entity and organization, respectively.

ACE2005 corpus (Walker et al., 2006) are nested (Ju et al., 2018). However, most work on NER copes with only non-nested entities (i.e., flat entities) and neglects nested ones. This leads to information loss since nested entities contain fine-grained information, which is crucial in understanding texts. There were only a few efforts in addressing nested NER. Approaches focused on nested NER either require hand-crafted features (Gu, 2006; Finkel and Manning, 2009; Lu and Roth, 2015; Muis and Lu, 2017) or ignore the dependencies among nested entities (Xu et al., 2017b; Li et al., 2017a). Dependencies refer to the occurrence of outer entities depend on the occurrence of inner entities.

In addition to the entity semantics, temporality between entities is another important dimension in understanding natural language. Many NLP tasks such as question answering (Llorens et al., 2015; Meng et al., 2017), text summarization (Ng et al., 2014; Wang et al., 2017) and causality (Mirza and Tonelli, 2014; Mirza, 2014; Ning et al., 2018) require the extraction of temporal information. For example, to understand the disease progression, we need the temporal information such as the starting time points of symptoms, drug frequencies and other disease history. Meanwhile, when summarizing the storyline from news reports, it is necessary to know the development of events over time, requiring timestamping or temporally ordering them. The process of identifying such information in time dimension is defined as temporal information extraction (TIE), which in general includes the extraction of time expressions (timexes), events, and relations (e.g., before, after) between any of them.

According to TimeML (Boguraev et al., 2005), events are textual spans that describe things are happening (e.g., *"reading"*) and timexes are spans that represent explicit temporal expressions, such as times, dates (e.g., *"3rd of July"*). These events in combination with timexes are called temporal entities, which can

be nested within each other. The process of identifying the temporal relation between any pair of temporal entities is known as temporal relation extraction (TRE), which has been remaining a challenging task because of the following two factors. The first one is that temporal information is often explicitly expressed in texts, requiring inference from additional information such as linguistic tenses and context-independent prior knowledge. For example, the event *"jump"* is punctual while the event *"know"* can be long-lasting. Such knowledge reflects the lasting time of events, which is informative in timestamping events. Moreover, the prior knowledge includes natural temporal orders such as people are injured first and then they will die. Context-independent knowledge, as described in these examples, is tough to be automatically modelled by computers. The other factor is the global dependencies, requiring models to consider both intra- and inter-sentence information.

With the advances in neural networks (NNs), many NLP tasks have achieved state-of-the-art (Miwa and Bansal, 2016; Zhang et al., 2018b) by adopting neural models such as convolutional neural networks (LeCun et al., 1999), long short-term memory networks (Hochreiter and Schmidhuber, 1997) and attention mechanisms (Vaswani et al., 2017b), which remove the dependence on both hand-crafted feature engineering and external knowledge bases (Lample et al., 2016; Song et al., 2018). Under this circumstance, we take advantage of NNs in our research to address the tasks: nested named entity recognition and temporal relation extraction. The outcome of our research will help the investigation of downstream tasks, which depend on the extraction of nested entities and their relations in time dimension. In addition, the proposed approaches can represent a useful means to populate resources such as knowledge bases and data annotation.

1.2 Research Questions and Hypotheses

To address the issues mentioned above, we have raised the following research questions:

RQ1 What are the state-of-the-art methods in extracting both flat and nested entities?

RQ2 How nested NER can be improved using NNs?

RQ3 How to measure the model generalisation ?

RQ4 How to temporally relate named entities?

In particular, we decompose the last question into the following sub questions:

RQ4-1 What are the state-of-the-art methods for TRE?

RQ4-2 How TRE can be improved using NNs?

RQ4-3 How to measure the ability of the TRE model in extracting both sentence- and document-level temporal relations?

We undertake this research by making a comprehensive investigation of work dealing with NER including both flat and nested NER. Then, we propose a novel neural model for nested NER without depending on external knowledge resources and hand-crafted linguistic features. To prove its effectiveness in coping with nested entities, we experiment with different domains of corpora. Moreover, we apply the model to different tasks, which evaluate the model by plugging it into other components designed for the corresponding NLP task. To temporally relate the entities of interest, we walk through the history of temporal information extraction and propose a neural approach to extract both intra- and inter-sentence temporal relations. We formulate the tasks in the following two research hypotheses in combination with their research objectives:

H1 Utilisation of inner entities can improve the detection of outer entities using NNs.

RO1 To conduct a comprehensive literature review including methods, resources and tools for NER.

RO2 To design neural NER models without feature-engineering and external knowledge bases.

RO3 To conduct evaluations in the settings of different domains and task-specific applications.

H2 Incorporation of latent information (i.e., event arguments) between temporal entities can improve TRE.

RO4 To conduct a comprehensive literature review focusing on temporal entity and their relation extraction.

RO5 To design a novel model to extract temporal relations including both intra- and inter-sentence relations.

RO6 To conduct evaluations in the settings of both intra- and inter-sentence temporal relations.

We detail in the succeeding chapters how each objective was achieved. Chapters 2 and 3 focus on RO1 and RO2, respectively. Chapter 4 details how RO3 is accomplished. Chapter 5 describes how RO4 and RO5 are fulfilled.

1.3 Contributions

The contributions of this research are summarised as follows:

- Propose a novel neural approach for nested NER without relying on hand-crafted features or external knowledge resources. Chapter 3 presents the model details.

- Employ the proposed model in the neuroscience domain under flat NER setting to help curation of neuroscience entities. We present this work in Section 4.2.1 of Chapter 4.

- Develop non-neural methods in combination with our model to identify pertinent and complex information about chronic obstructive pulmonary disease phenotypes from clinical textual data. This work is detailed in Section 4.2.2 of Chapter 4.

- Introduce subword units to the proposed model to extract adverse drug event and medication information from clinical records. With subword units, the model improves the extraction of sparse entities without depending on any external knowledge resources and hand-crafted features. We discuss this work in Section 4.2.3 of Chapter 4.

- Develop a PICO model for recognizing elements of patient/population (P), intervention (I), comparator (C), and outcomes (O). The PICO model was further used in the task of scientific abstract screening in combination of biomedical and health domains. Section 4.2.4 of Chapter 4 gives details of this work.

Publication	Venue	Chapter
A Text Mining Pipeline Using Active and Deep Learning Aimed at Curating Information in Computational Neuroscience (Shardlow et al., 2018)	Neuroinformatics	Chapter 4
A Neural Layered Model for Nested Named Entity Recognition (Ju et al., 2018)	Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies	Chapter 3, 4
An Ensemble of Neural Models for Nested Adverse Drug Events and Medication Extraction with Subwords (Ju et al., 2019a)	Journal of the American Medical Informatics Association	Chapter 4
Annotating and Detecting Phenotypic Information for Chronic Obstructive Pulmonary Disease (Ju et al., 2019b)	Journal of the American Medical Informatics Association Open	Chapter 4
Improving Reference Prioritisation with PICO Recognition (Brockmeier et al., 2019)	BMC Medical Informatics and Decision Making	Chapter 4

Table 1.1: Publications published during the course of the PhD.

- Propose a novel neural model to extract both intra- and inter-sentence temporal relations using latent information between temporal entities. We present this work in Chapter 5.

A significant proportion of the thesis is already published. Table 1.1 presents a list of the publications, as well as their correspondence to the thesis chapters. The contents of most publications can be replicated with our released codes.

Chapter 2

Background

In this chapter, we present the following regarding named entity recognition:

- Explain definitions of named entities, named entity recognition and their related concepts
- Summarise related work towards named entity recognition
- Summarise the existing available data sources and tools for named entity recognition
- Summarise the evaluation of named entity recognition

2.1 Introduction

The term named entity (NE) was first coined in 1995 in the Sixth Message Understanding Conference (MUC-6) (Grishman and Sundheim, 1996), which aimed to extract structural information of company and military activities from unstructured text, such as news articles. When identifying such information, researchers noted that it is crucial to recognise information units such as names of people, location and organisation, temporal expressions (e.g., time and dates) and other numeric expressions including money and percentage expressions. The process of identifying such information units (i.e., named entities) was defined as named entity recognition (NER). As a follow-up of MUC-6, MUC-7 (Marsh and Perzanowski, 1998) focused on the topics of airplane crashes and aircraft launches. After MUC, there have been shared tasks to deal with language-independent

NER, such as Conference on Natural Language Learning 2002 (CoNLL-2002) (Tjong Kim Sang, 2002) and information retrieval and information extraction (IREX) (Sekine and Isahara, 2000), CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003). The Automatic Content Extraction (ACE) program aimed to develop information extraction technology to support automatic processing of source language data including newswire, broadcast news, telephone conversations (Consortium et al., 2004). The 2004 edition of ACE included seven entity categories: person (PER), organisation (ORG), location (LOC), geo-political entity (GEO), facility (FAC), vehicle (VEH) and weapon (WEA), which are widely used for developing NER tools and annotating corpora in the general domain. Those entity categories are kept in the more targeted data including weblogs, broadcast news, newsgroups, broadcast conversation in the 2005 edition of ACE program (Walker et al., 2006). An example of annotated named entities from ACE2005 is shown in Figure 2.1 where the boundaries of entity spans are marked with square brackets and the categories of entities are in subscript.

Besides general domain tasks, NER has also been tackled in other domains such as biomedicine, pharmacy, chemistry, etc. Common entity categories in biomedicine include but are not limited to genes, gene products, chemicals, drugs, numeric expressions (e.g., drug dosage, frequencies, temporal expressions) and diseases. One of the most common corpora for biomedical NER is the GENIA corpus (Kim et al., 2003) which contains 1,999 abstracts from PubMed database. A biomedical example of annotated named entities from GENIA is shown in Figure 2.1 where the boundaries of entity spans are marked with square brackets and the categories of entities are in red.

Due to the properties of natural language, many entities are nested within other entities: embedded names which are included in other entities (i.e., *nested entities*), as illustrated in Figure 2.2. In comparison with nested entities, we refer to non-nested named entities as *flat entities*. GENIA (Kim et al., 2003) was the first corpus to include nested entity annotations, which account for 16.7% among all entities (Gu, 2006). The process of dealing with the extraction of both nested and flat entities is referred to as nested NER.

In the next section, we present an overview of the state-of-the-art NER. Our literature review focuses on methods, resources, tools and evaluation metrics for both flat and nested NER. We individually describe each topic in the following sections.

ACE2005 [The federal government]$_{\text{GPE}}$ will not appeal the [court]$_{\text{ORG}}$ ruling that cleared the way for [same-sex]$_{\text{PER}}$ unions.

GENIA [IL-2 gene]$_{\text{DNA}}$ expression and [NF-kappa B]$_{\text{protein}}$ activation through [CD28]$_{\text{protein}}$ requires reactive oxygen production by [5-lipoxygenase]$_{\text{protein}}$.

Figure 2.1: Sentences with named entity annotations.

$$\underbrace{\text{The premier of the } \underbrace{\underbrace{\text{western Canadian}}_{\textbf{LOC}} \text{ province of } \underbrace{\text{British Columbia}}_{\textbf{GPE}}}_{\textbf{GPE}} \ldots}_{\textbf{PER}}$$

Figure 2.2: A sentence from ACE2005 (Walker et al., 2006) containing the nested four entities.

2.2 Overview of Flat NER

In the 1990s, work on flat NER mainly depended on domain terminologies and hand-crafted linguistic features to extract entities. These feature sets depended on domain experts with expertise in specific domains. One way to reduce such dependence is to utilise machine learning (ML) based methods (i.e., learning-based methods). ML is an area that studies algorithms to automatically make task-specific decisions/predictions with patterns and inference (Bishop, 2006). Learning-based methods most often use a rich set of hand-crafted features (properties/attributes of texts), which are subsequently fed into the ML algorithms. Feature selection is an empirical process, which mainly relies on linguistic intuition and task-specific error analysis, making it time consuming and expensive. To avoid hand-crafted feature engineering, neural networks (NNs) have been widely used to enable automatic extraction of high-level and abstract features. We denote methods that employ NNs and non-neural ML methods to conduct NER as neural-based and conventional learning-based methods, respectively. Therefore, learning-based methods are split into neural-based methods and conventional learning-based methods. In general, we roughly divide methods of flat NER into three categories: rule-based, conventional learning-based and neural-based methods. We present methods in each category in the following sections.

Feature	**Example (Manchester)**
Whether the word contains Greek letters	0
Whether the word contains digits	0
Whether the word contains symbols	0
Whether the word in dictionary(e.g. gazetteer)/terminology	1
Part-of-speech tag	Noun
The word itself	Manchester
The lowercase of the word	manchester
The character 3-grams that compose the word	Man, anc, nch, che, hes,est, ste, ter
The character 4-grams that compose the word	Manc, anch, nche, ches, hest, este, ster
Word's capitalization pattern	Xxxxxxxxxx
Word length	10
Weather the word contains a dash	0
Whether the word is inside double quote marks	0
Whether the word is part of any entity	1

Table 2.1: Common feature sets used in earlier NER work. 0 and 1 represent "False" and "True", respectively.

2.2.1 Rule-based Methods

Earlier work on flat NER was mostly based on gazetteers and linguistic hand-engineered rules (Nadeau and Sekine, 2007). Common rules are based on orthographical, lexical and syntactic patterns, which reflect the properties of linguistic knowledge. Table 2.1 shows examples of common features. The development of rules requires researchers to have domain-specific expertise and linguistic knowledge. Moreover, the performance of rule-based systems heavily relies on the coverage of rules, making it difficult to learn new entity mentions. In addition, rules in general are domain-specific, thus can not be easily adapted to other domains. These limitations greatly restrict the generalization and adaptation abilities of rule-based systems. To alleviate this issue, researchers investigated ML methods that enable automatic learning to make decisions.

Sequence	The federal government will not appeal the court ruling that cleared the way for same-sex unions.
BIO sequence	B-GPE, I-GPE, I-GPE, O, O, O, O, B-ORG, O, O , O, O , O, O, B-PER, O, O
BIOLU sequence	B-GPE, I-GPE, L-GPE, O, O, O, O, U-ORG, O, O , O, O , O, O, U-PER, O, O

Figure 2.3: A sentence with different labelling schemes.

2.2.2 Conventional learning-based Methods

Conventional learning-based methods for flat NER can be roughly divided into four categories: fully-supervised, semi-supervised, active learning and unsupervised methods. Our research investigates NER methods with supervision, we therefore focus on the first three categories and present the related work for each category in the following sections.

2.2.2.1 Fully-supervised Methods

Fully-supervised methods aim to learn from a labelled data set to produce an inferred function that maps new input to a predefined label (Russell and Norvig, 2010). Such labelled data sets are called **training data**, which is a prerequisite condition to apply supervised ML algorithms. Given unlabelled texts, supervised ML systems generally start from text preprocessing (e.g., sentence splitting, word segmentation), and then is followed by hand-crafted feature engineering, automatic pattern learning. Those learnt patterns are subsequently used for inferring new entities. In particular, sentence splitting refers to the process of splitting the textual data into a list of sentences based on a set of punctuations that mark the linguistic boundary of sentences. Word segmentation is the process of segmenting a sentence into tokens.

Fully-supervised ML methods generally formalise flat NER as a sequence labelling problem that is to assign one of the B (Beginning of the entity), I (Inside of the entity), and O (Outside of the entity) labels to each word which corresponds to the ACE2005 sentence in Figure 2.1.

In addition to BIO, label L (last word of the entity) and U (entity contains only one single word) are introduced to indicate the span boundary of entities, constituting the BIOLU tagging scheme (Ratinov and Roth, 2009). Figure 2.3

shows the label sequence annotated with BIOLU tagging scheme. A sequence is composed of a set of words of arbitrary length. Therefore, one sentence or the whole paragraph can be considered as one sequence. In earlier stages of NER, one of the common approaches was to employ a list of entity mentions derived either from a corpus or existing gazetteers (Mikheev et al., 1999; Stevenson and Gaizauskas, 2000; Toral and Munoz, 2006; Kazama and Torisawa, 2008), where entities were recognised in a dictionary lookup manner, which significantly limited the ability in identifying new entities. To better identify new entities, researchers started to investigate ML algorithms coupled with hand-crafted features and domain knowledge.

For example, Borthwick and Grishman (1999) and Bender et al. (2003) employed maximum entropy (Berger et al., 1996) driven by features including lexical, dictionary and word-surface features (e.g. number of digits contained in the word) to identify entities. Lin et al. (2004), Finkel et al. (2004) and Ahmed and Sathyaraj (2015) adopted this method coupled with specific features, improving the performances of biomedical flat NER (Tsuruoka and Tsujii, 2004; Sasaki et al., 2008). In addition, McCallum and Li (2003), Settles (2004), Klinger and Friedrich (2009), Özkaya and Diri (2011) and Nguyen et al. (2019) utilised conditional random fields (CRFs) (Lafferty et al., 2001) to identify entities, showing the effectiveness of CRFs in different domains. Furthermore, hidden markov model (HMM) (Malouf, 2002; GuoDong, 2004; Liu et al., 2005) were also commonly utilised coupled with hand-crafted linguistic features and various knowledge resources to identify entities. Support vector machine (SVM) (Cortes and Vapnik, 1995), another widely used algorithm (Kazama et al., 2002; Takeuchi and Collier, 2002, 2005; Singh et al., 2009) were also investigated to enable automatic entity extraction.

2.2.2.2 Semi-supervised Methods

Fully-supervised methods depend on annotated corpora (i.e., a collection of textual data) to design ML models. However, such corpora are not always available as they are labor-intensive, time-consuming and even might not be available for some languages. Therefore, it is desirable for ML algorithms to work with semi-supervision, which exploits the unlabelled data besides limited labelled data to improve learning performance. One of the most common approaches in semi-supervision is bootstrapping (Abney, 2002), which generally starts training with

only a small set of labelled data serving as a small degree of supervision to develop models. Specifically, a NER model firstly searches the sentences containing the names that are included in the given seed set. The model then identifies contextual clues where these names appear and attempts to search other instances of these names that share similar contexts. Such a process is reapplied to new instances to discover new related contexts. The model repeats this process, yielding a large number of both names and contexts. The first semi-supervised work dealing with NER was from Collins and Singer (1999), who developed the following seven seed rules, which provided the only supervision in their study:

- full string = New York –>Location
- full-string = California –>Location
- full-string = US. –>Location
- contains (Mr.) –>Person
- contains (Incorporated) –>Organization
- full-string = Microsoft –>Organization
- full-string = I.B.M. –>Organization

Riloff et al. (1999) presented a multi-level bootstrapping approach, which simultaneously generated a semantic lexicon and extracted patterns based on the requirements of dictionaries. A mutual bootstrapping technique was developed to alternately select the best extraction pattern for the category and bootstrap its extraction into the semantic lexicon, which is used for selecting the next extraction pattern. Furthermore, more attempts were made to improve bootstrapping methods (Cucchiarelli and Velardi, 2001; Etzioni et al., 2005; Elsner et al., 2009), which had been widely adopted to reduce the dependencies on labelled data for NER (Mohit and Hwa, 2005; Kozareva, 2006; Liao and Veeramachaneni, 2009; Han et al., 2015; Mishra and Diesner, 2016). Note that most work in this category only dealt with entity categorisation, which assumes that entities are given.

Another popular approach in semi-supervised NER is distant supervision, which leverages knowledge bases to populate training data without human cost. In a naive distant supervision NER setting, if a string in textual data is included in a predefined dictionary of entities, the string might be an entity. Such straight

forward string matching introduces noisy as well as incomplete entities. To address the noise, Bing et al. (2015) presented a pipeline system that contained three steps: (1) identified lists of semantically-related items using lexico-syntactic patterns, (2) used distant supervision in combination with a label-propagation method to find entity mentions that can be confidently labelled and (3) used these entity mentions to train classifiers to label more entity mentions. Similarly, Tu et al. (2017) proposed an approach that leveraged the conjunction and comma writing style as the list constraint to enlarge the set of training instances. Those constraints were further incorporated into a unified discriminative learning framework for NER, showing effectiveness in extracting drugs from clinical documents. Besides, Lee et al. (2016b) presented a method that used distant supervision to generate labelled training data, whose labels were refined using a bagging-based active learning method. We detail NER with active learning in the next section.

2.2.2.3 Active Learning Methods

Fully-supervised methods significantly benefit from the size and quality of annotated data sets. In general, the larger the training set is, the better the accuracy the model can achieve. As mentioned above, annotating a large corpus is a time-consuming and expensive process. One way to obtain annotated training data with saving cost is crowdsourcing, which outsources the unlabelled data to a crowd of workers for labelling. In practice, some workers may be "adversaries" or "spammers" (e.g., robots) or lack enough expertise for the annotation task. These factors lead to inaccurate supervision returned by the crowd, thus negatively affecting the learning performance.

Another alternative way to obtain annotated training data is to annotate only the data which are helpful to improve the overall accuracy. It resulted in the active learning (Settles, 2009), which gives the learner a degree of control by allowing it to select the most informative instances to add to the training set. Compared with semi-supervised learning, active learning assumes that the ground-truth labels of unlabelled instances can be queried from a human annotator (Zhou, 2017). A typical active learner begins with a small labelled set L, selects one or more informative mention instances from a large unlabeled pool U, learns from these labelled mentions (which are then added to L), and then repeats the process. In this way, the learner aims to achieve high accuracy with minimum labelling effort, thus enabling flat NER in the domains where only limited annotated data

is available.

Active learning methods were mostly based on uncertainty sampling, which selects instances with least uncertainty (Scheffer et al., 2001; Culotta and McCallum, 2005; Kim et al., 2006) or chooses the most confident instance based on multiple learners (Dagan and Engelson, 1995). Earlier work such as Scheffer et al. (2001) selected "difficult" unlabeled instances by querying the instance with the smallest margin between the posteriors for its two most likely labels. In addition to considering only the uncertainty of the models (learners), Kim et al. (2006) further incorporated the diversity of the corpus in their uncertainty sampling strategy, to select the most informative instances, showing effectiveness in reducing human effort in the biomedical data. Similarly, Shen et al. (2004) proposed a multi-criteria technique including informativeness, representativeness and diversity, aiming to maximise the contribution of the selected instances, thus minimising the human annotation efforts during the selection of examples for labelling. They demonstrated the advantage in reducing labelling effort required in flat NER without harming its performance. To find the optimal stopping criteria in selecting informative instances, Laws and Schätze (2008) proposed a gradient-based stopping criterion, which was able to stop active learning with high reliability, but can achieve comparable optimal flat NER performance which only needed around 20% training data compared to exhaustive labelling.

The common drawback of the above-mentioned methods is that their work assumed that the annotation cost for each sentence was the same, which is not the case. For example, informative sentences might differ from each other, such as in terms of sentence lengths, thus requiring different annotation efforts. This assumption could inevitably lead to the underestimation of manual effort, especially for tasks that require a massive corpus. To address the assumption, online learning was designed for massive data training when computing resources are limited. Compared to batch learning that aims to induce an optimal model by training on all the available labelled data, online learning generates a model based on every single fresh random sample in the massive data, but could obtain better performances (Bottou and Cun, 2004).

Active learning assumes that labels from human annotators are always correct, which is invalid. In addition, every annotator is paid equally regardless of their different levels of expertise. To relax such unrealistic assumptions, proactive learning (Donmez and Carbonell, 2008, 2010), a generalisation of active learning,

was proposed to allow the existence of fallible annotators who are accordingly paid based on their expertise. In a proactive learning NER setting, the annotation of unlabelled data is similar to active learning except that proactive learning considers the reliability of each annotator in annotating each selected instance. The first attempt at proactive learning methods for NER was Li et al. (2017b), who proposed a batch sampling method that assigned the tough instances to the reliable experts while the remaining instances were presented to the fallible annotators.

2.2.3 Neural-based Methods

Neural networks (NNs) are a family of powerful ML algorithms, introduced in 1943 when Warren McCulloch and Walter Pitts used electrical circuits to model how neurons work in brain functions, which was used to simulate intelligent behaviour (McCulloch and Pitts, 1943). Compared with traditional ML algorithms (e.g., SVM), neural network (NN) systems consist of an input layer, an output layer and hidden layers, as shown in Figure 2.4. As shown in Figure 2.5, each layer is composed of a set of neurons, each of which conducts a weighted sum of input from the previous layer and then passes the result into the next layer through non-linear operations. The computation process that the input flows from the lower to upper layers is referred to as the forward pass computation. To allow the networks to be aware of the feedback, the neural model subsequently takes the forward pass computation results as input and computes their gradients following the derivative chain from top to bottom of the architecture, as shown in Figure 2.6. Such a computation process is called backward pass computation (i.e., backward propagation). Through forward and backward pass computations, NN models are able to learn high-level and abstract features in the form of vectors/representations, which encode semantic and syntactic properties from the raw texts. We present the common NN models in NLP in the next section.

2.2.3.1 Common Neural Networks

Convolutional neural network (CNN) (LeCun et al., 1999) is one type of NNs, which represents a feature function that is used to learn representative

[1]https://medium.com/datadriveninvestor/letting-neural-networks-be-weird-6792ea587d67
[2]https://i.stack.imgur.com/gzrsx.png
[3]http://2017.igem.org/Team:Heidelberg/Software/DeeProtein

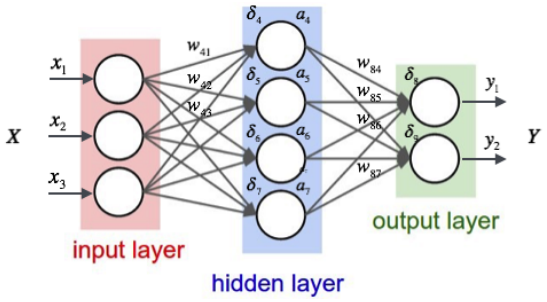

Figure 2.4: A typical neural network architecture.[1]

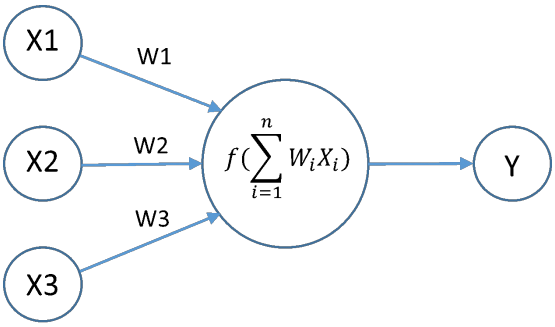

Figure 2.5: Operation in one neuron.[2]

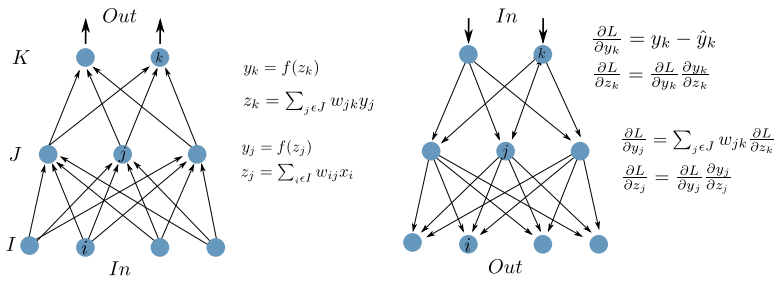

Figure 2.6: A forward and backward pass through a neural network.[3]

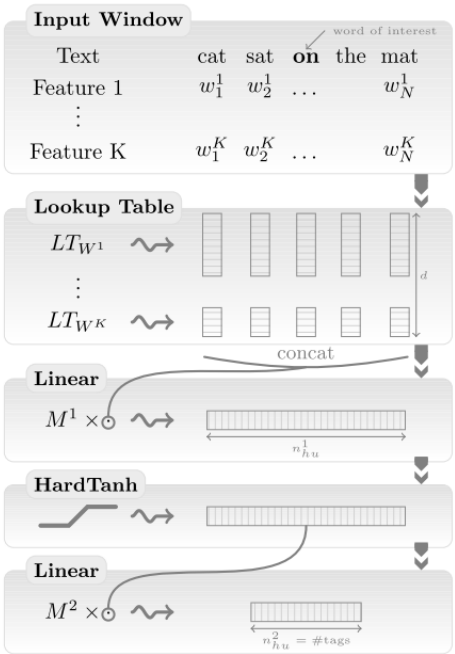

Figure 2.7: An example of CNN-based model for flat NER (Collobert et al., 2011a).

features from a word-level or character-level sequence. These abstract features have been widely used for various natural language processing (NLP) tasks such as NER (Ma and Hovy, 2016; Lample et al., 2016; Gridach, 2017), entity linking (Gupta et al., 2017), and co-reference resolution (Fragkou, 2017; Stone and Arora, 2017). Figure 2.7 shows an example of feature extraction using a CNN network. In CNN-based methods, the first step towards sentence modelling is to tokenize sentences into words, which are further transformed into a word embedding matrix (i.e., embedding layer). Then, convolutional filters are applied on the word embeddings to extract features, which are subsequently fed into a max-pooling layer to obtain the most representative feature from each filter. Concretely, the function of a max-pooling layer is to get the maximum value from each filter to produce a fix-sized output, which is the final representation for the given sequence. CNN-based methods are capable of learning latent semantic patterns of the data but fail to preserve sequential order and model long-distance contextual information. Such information, however, can be taken into account by recurrent neural networks, which are described in the next paragraph.

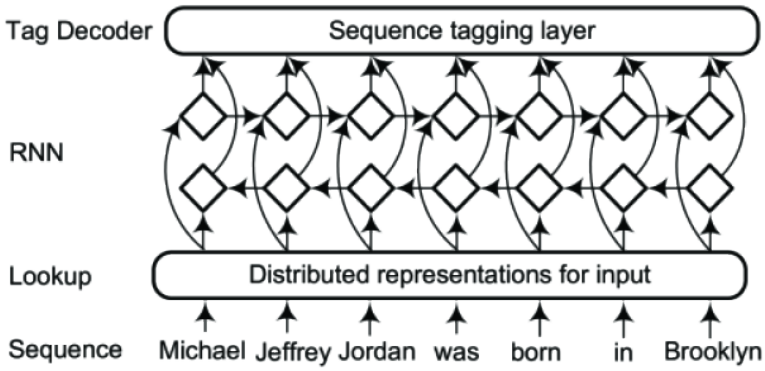

Figure 2.8: An example of RNN-based method for flat NER (Li et al., 2018b).

Recurrent neural network (RNN) (Goller and Kuchler, 1996), is a special type of NN architecture that connects each neuron in a temporal order. In a RNN model, each neuron corresponds to an instance (e.g. word, character) of the input sequence, as shown in Figure 2.8. The input sequence of RNN is typically represented by a matrix where each row corresponds to the vector of instance, which is fed sequentially (one by one) to a chain of recurrent units that are temporally connected. Given an input sequence, a RNN recursively computes the output of each neuron and uses that result to compute the output of next neuron, thus enabling memorizing sequential information and long-distance contextual dependencies of the input sequence. Vanilla RNNs suffer from the gradient exploding/vanishing problem arising in backward pass computation, where gradients from the upper layers have to go through continuous matrix multiplications to propagate the lower layers. During backward pass computation, if the gradients are less than one, they shrink exponentially until they vanish. This phenomenon is called gradient vanishing. Similarly, if the gradient values are larger than one, they get larger and eventually blow up and crash the model, this is the gradient exploding. Gradient vanishing/exploding problem makes the training and tuning processes extremely difficult. Variants of RNN such as long short-term memory network (LSTM) and gated recurrent unit network (GRU) (Cho et al., 2014) were introduced to address the gradient exploding/vanishing problem.

Long short-term memory networks (LSTMs) (Hochreiter and Schmidhuber, 1997) use three gates (input, forget, and output gates) to calculate the output of each neuron through a combination of these three gates. Given an input sequence $X = x_1, x_2, \ldots, x_n$ where x_i represents a word/character, the output at

time t is computed using the following equations:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2.1}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2.2}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{2.3}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \sigma(W_c x_t + U_c h_{t-1} + b_c) \tag{2.4}$$

$$h_t = o_t \cdot \sigma(c_t) \tag{2.5}$$

where f_t, i_t, o_t, c_t, h_t represent the values of forget gate, input gate, output gate, cell state and hidden state at time t. c_{t-1}, h_{t-1} represent the values of cell state and hidden state at time $t-1$. W, U, b are parameters and $\cdot$ represents element-wise multiplication.

Similarly to LSTMs, gated recurrent unit networks (GRUs) also use the gate mechanism that consists of one input gate and forget gate, thus is more computationally efficient as it removes the output gate. The selection between GRUs and LSTMs is decided by the availability of computational resources. For flat NER, LSTM-based models have been widely used in many work (Ma and Hovy, 2016; Lample et al., 2016; Lin et al., 2017a; Muis and Lu, 2017; Ju et al., 2018).

Attention mechanisms were originally used in the context of neural machine translation. Machine translation is the task of translating text or speech from one language to another, formalised as a sequence to sequence problem. Models designed for machine translation are called seq2seq models, which are generally composed of one encoder stacked with one decoder. Figure 2.9 shows a typical seq2seq model. The encoder, generally RNN-based frameworks, copes with processing the input sequence and encodes the last hidden state of the calculated output into a fix-sized context vector. This representation is expected to be a good summary of the entire input sequence. The decoder, generally RNN-based, uses the context vector as the initial state to generate the translated output sequence. When producing the context vector, traditional seq2seq models used only the last hidden state of the encoder but ignored the remaining hidden rests,

resulting in information loss when summarising longer sequences. To address the information loss issue, attention mechanisms are introduced to allow the decoder to calculate its weighted sum from all hidden states of the encoder, thus enabling better summary of context vector. Note that the weights are varied for each unit (e.g., word) composed in the transformed output sequence. Such a mechanism significantly benefits tasks that require an alignment between input and output.

Attention mechanisms have achieved great success in many tasks such as machine translation (Luong et al., 2015; Liu et al., 2016; Malaviya et al., 2018), text summarisation (Rush et al., 2015; Kryściński et al., 2018; Cohan et al., 2018), dialogue generation (Mei et al., 2017; Pasunuru and Bansal, 2019; Zhou et al., 2018; Huang et al., 2018), and aspect-based sentiment analysis (He et al., 2017; Saeidi et al., 2016; Ruder et al., 2016; Hazarika et al., 2018).

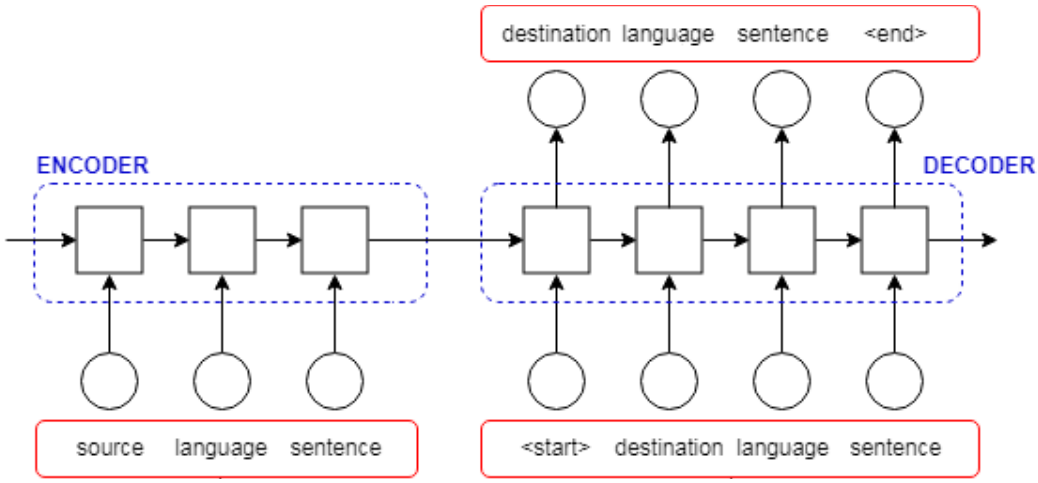

Figure 2.9: A typical architecture of seq2seq model.

Transformers (Vaswani et al., 2017a) are based on the attention mechanism to look at an input sequence and decide at each step which other parts of the sequence are important. Figure 2.10 shows the architecture of the transformer, which is composed of an encoder-decoder. The encoder and decoder are composed of a stack of blocks, which are described by N_x in Figure 2.10. Unlike the encoder, each block in the decoder additionally incorporates one more layer besides the two layers. The core of the transformer is multi-head self-attention. Each head is trained to encode the context information from different aspects such as semantics, syntax (Voita et al., 2019). Self-attention is the scaled dot-product attention whose output is a weighted sum of the word representations, where the weight assigned to each word representation is determined by the dot-product of

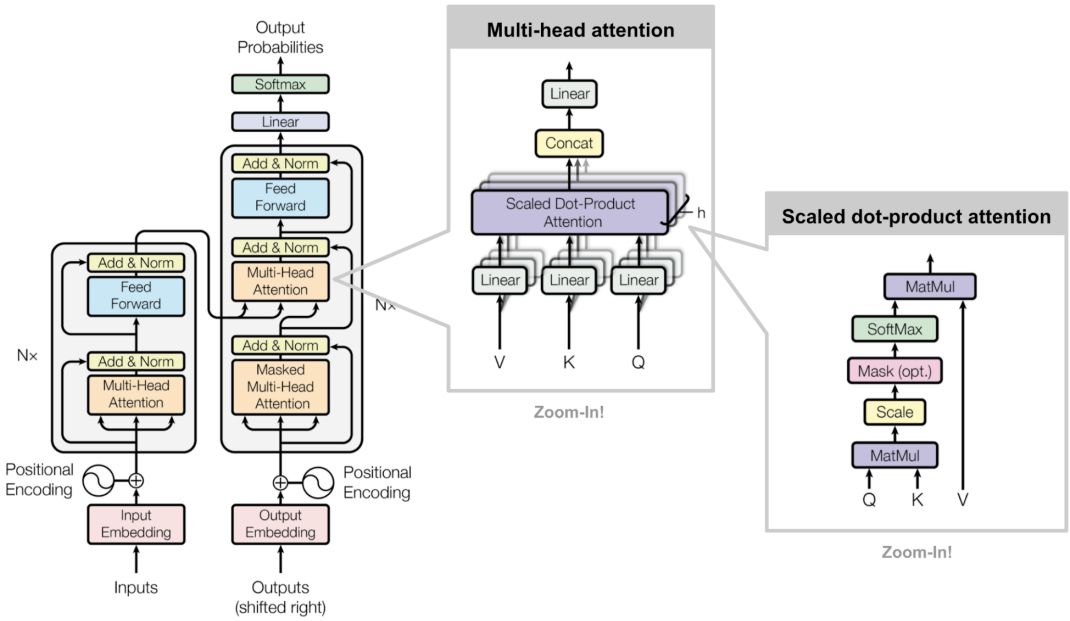

Figure 2.10: The architecture of transformer (Vaswani et al., 2017a). Q, K and V represent the query, key and its corresponding value. N_x is the number of identical layers, each of which has a multi-head self-attention layer and a fully connected feed-forward network.

the word with the sequence. Given the input sentence, the transformer firstly calculates the position information of each word to obtain its position embeddings, which carry the sequential order information. To get contextualised word representations, the encoder uses the multi-head self-attention mechanism to calculate a weighted sum of the input sentence to obtain the representation for the current word.

Based on the initial transformer idea, Radford et al. (2018) proposed the single directional generative pretrained transformer (GPT) for language understanding tasks. Moreover, Devlin et al. (2019) proposed bidirectional encoder representations from transformers (BERT) by incorporating both left and right context in all layers. Figure 2.11 describes the difference between GPT and BERT models. Transformer models use the attention mechanism as an alternative to the RNN family.

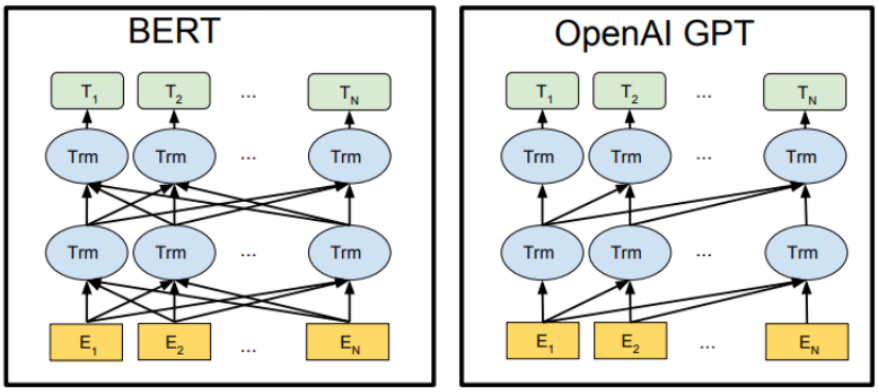

Figure 2.11: Differences between BERT, OpenAI GPT (Devlin et al., 2019). In both models, E, Trm and T represent input, transformers and output.

2.2.3.2 Components of Neural NER Models

In flat NER tasks, neural-based models generally include three components: word encoder, context encoder and label decoder. The word encoder deals with the preparation of input while the context encoder is used to capture contextual features by taking the output from the word encoder. The label decoder is responsible for decoding the output from context encoder into a label sequence. We present each component in the following sections.

Word Encoders

The function of a word encoder is to transform a sequence of words into vectors and then optionally concatenate other types of representations (e.g., character-level representation) to augment the word representation. In a flat NER task, there are four types of representations: word-level representation, character-level representation, subword-level representation and their combination.

Word-level Representations

One straightforward way to represent a word is to use a one-hot vector, which assigns an orthogonal vector to each unique word. However, one-hot vector representation fails to represent the semantic meaning between words such as "king" and "queen" which share semantic meaning. One effective solution is to use low dimensional vectors for word representations, which are called *word embeddings*. There are mainly two approaches to obtain word embeddings, namely continuous bag-of-words (CBOW) and continuous skip-gram algorithms, as shown in Figure 2.12. In the CBOW architecture, the model predicts the word based on the given

surrounding context words. However, the skip-gram model targets on predicting the neighbouring words for the given word. Pre-trained word embeddings are obtained from an extensive collection of text data (e.g., Wikipedia) using either of these models. Pre-trained word embeddings have been proved effective in many tasks (Lample et al., 2016; Ma and Hovy, 2016; Miwa and Bansal, 2016). In the general domain, common pre-trained word embeddings include Google Word2Vec[4], Stanford GloVe (Pennington et al., 2014), Facebook fastText (Mikolov et al., 2018) and SENNA[5]. In biomedicine, Moen and Ananiadou (2013) trained word embeddings on the PubMed[6] and PMC texts and their combination using the Google Word2Vec[7]. Besides, Chen et al. (2018) trained large word embeddings using the abstracts from PubMed and clinical notes in MIMIC III Clinical database (Johnson et al., 2016).

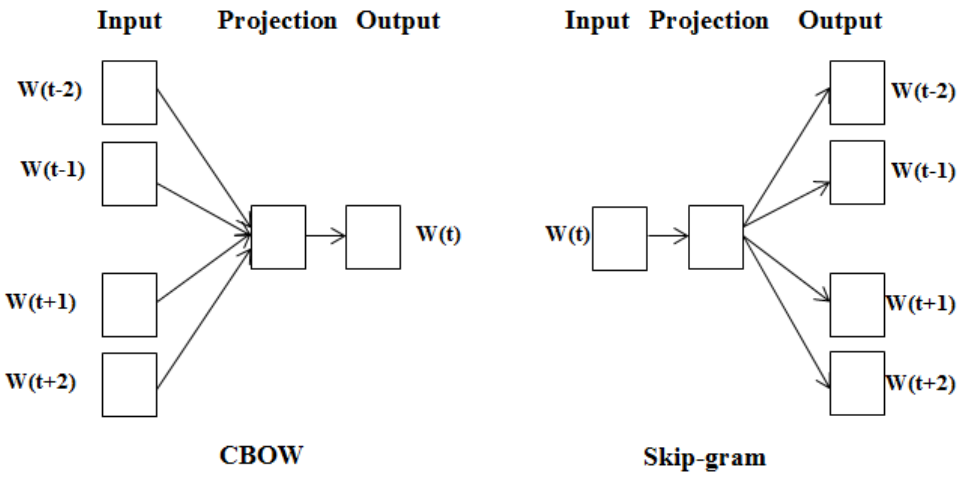

Figure 2.12: CBOW and continuous skip-gram models (Suleiman et al., 2017).

Character-level Representations

In addition to word-level representations, another effective way to augment word-level representations is the character-level representation, which encodes the morphological features of a word such as prefix and suffix. There are two popular approaches to obtain character-level representations, namely: CNN and RNN models, as shown in Figure 2.13. In both models, each word is considered as a sequence of characters. Given one word, these models first assign a vector to

[4]https://code.google.com/archive/p/word2vec/
[5]https://ronan.collobert.com/senn
[6]https://www.ncbi.nlm.nih.gov/pubm
[7]https://code.google.com/archive/p/word2vec/

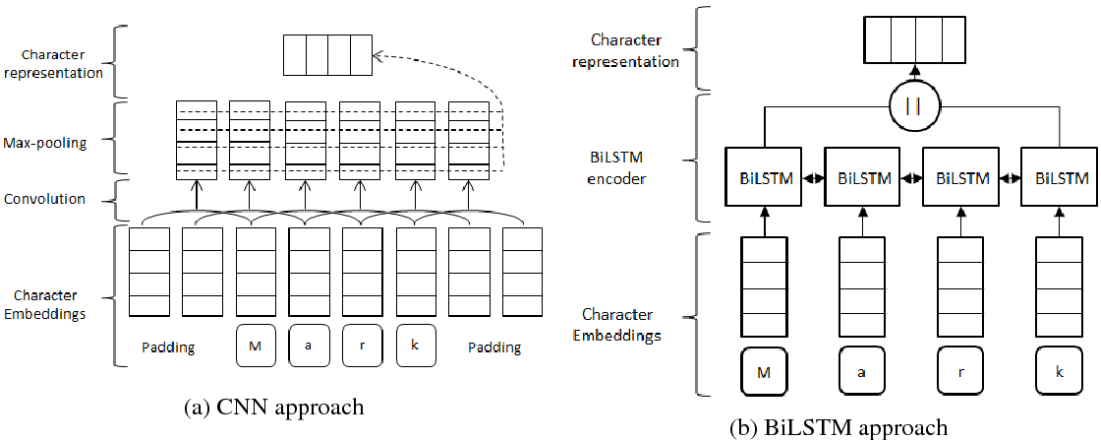

Figure 2.13: CNN-based and RNN-based (LSTM) architectures for character representations (Reimers and Gurevych, 2017).

each character through a dictionary lookup. These vectors are generally randomly initialized. In the CNN model, one CNN is applied to the matrix where each row is the corresponding vector of the character. Convolutional operations are applied to the matrix, and then the output goes through a max pooling function, whose outputs are further concatenated to produce the character-level representation. Earlier work (Ma and Hovy, 2016; Chiu and Nichols, 2016) mainly used one CNN to generate character-representations. More recently, Peters et al. (2018) used a weighted sum of the output generated from two-layer bidirectional language models with character convolutions.

Unlike CNN models, LSTM is the most popular model for RNN models. Figure 2.13 shows an example of the RNN model to obtain character-level representations. Lample et al. (2016) applied one LSTM to compute the forward and backward representation of the given character sequence and then concatenated with last hidden state from each direction as the character-level representation. This approach has been widely used in other work (Rei et al., 2016; Gridach, 2017; Ju et al., 2018), demonstrating its effectiveness in capturing morphological information. Instead of focusing on the characters on word level, Kuru et al. (2016) treat the sentence as a sequence of characters and applied LSTM for flat NER. Furthermore, Akbik et al. (2018) used the pre-trained character embeddings obtained from character-level language modelling for flat NER. These character-level pre-trained embeddings encode syntactic-semantic word features contextualized by their surrounding text, meaning that the embeddings of the same word differ depending on its context.

Subword-level Representations

The idea of subwords is used to represent unseen and rare words using byte pair encoding (Sennrich et al., 2016), which represents words by iteratively merging the most frequent adjacent/consecutive characters into longer character sequences (i.e., subwords). Similarly to character-level representations, subword-level representations can be obtained from the usage of CNN and RNN models. Sheng and Natarajan (2018) and Ju et al. (2019a) used subword-level representations as input and assign the predicted label to each subword instead of a word. When merging the subword labels into their corresponding word labels, the first subword label is kept as their word label. Benefiting from contextualized representations (Kuru et al., 2016; Peters et al., 2018; Radford et al., 2018), Devlin et al. (2019) proposed a new model that takes the subword sequences as input and learns bidirectional contextualized representations of the sequence. In flat NER, an additional layer is stacked on top of BERT (Devlin et al., 2019) to fine-tune the representations.

Hybrid Representations

To improve task-specific performance, other representations such as part-of-speech (POS) tags are additionally concatenated with the above mentioned representations, serving as input for neural models. Huang et al. (2015) gained improvement of flat NER by incorporating spelling and gazetteer features in addition to word-level representations. Likewise, Chiu and Nichols (2016) included features such as word capitalization to improve the performance. In addition, Wei et al. (2016) appended POS tags, chunking and word-shape features to the word embeddings to augment word representations, yielding better performances. Features including word shape, syntactic features (e.g., POS tags, dependency roles, morphological features) have been shown helpful in flat NER (Strubell et al., 2017; Lin et al., 2017a), showing their effectiveness in augmenting information.

Context Encoders

The context encoder takes in the output from word encoder and outputs representations that encode context information. Popular context encoders in flat NER include CNN, LSTM, RNN, transformer-based models (e.g., BERT (Devlin et al., 2019)). We briefly introduce each in the following sections.

CNN

The typical CNN-based context encoder is shown in Figure 2.7. Given the

output from word encoder, the CNN-based context encoder applies convolutional operations on the output to generate local features around the word, forming a vector with maximum/average operation. The number of words taken into consideration depends on the filter window size. The results are concatenated and then used as the input for the label decoder to compute all possible labels for each word. Such context encoder has been applied in different domains (Collobert et al., 2011a; Yao et al., 2015). Based on CNN, Strubell et al. (2017) proposed iterated dilated convolutional neural networks (ID-CNNs) to enable faster computation without sacrificing the performance.

RNN Family

Compared with CNN, there are more studies which adopt RNN family networks (RNN, GRU and LSTM) especially their bidirectional versions to capture both left and right context information. The typical bidirectional RNN-based context encoder used in flat NER is shown in Figure 2.8, which takes in the representation from the word encoder and iteratively calculates each hidden state conditioned on the previous states. The concatenation of hidden state at each time step (i.e., word position) constitutes the context representation, which is subsequently fed into a label decoder for label prediction.

Transformers

A transformer-based context encoder takes in the representation from one word encoder and produces an output by using multi-head self-attention. Another approach to use transformer context encoders is to stack an additional layer on top of BERT (Devlin et al., 2019) to fine-tune the intermediates from earlier layers.

Label Decoders

The label decoder takes in the output from a context encoder and decodes the representation into a label sequence. We discuss the commonly used label decoders: softmax and CRFs in the following parts.

Softmax

The softmax formalises label decoding as a multi-class task. Given the context representation from context encoder, the softmax layer calculates the distribution scores for each label class of each word. The label with the maximum probability is selected as the final label for the word. Softmax has been widely used in many studies (Collobert et al., 2011a; Strubell et al., 2017; Xu et al., 2017c; Wang and

Lu, 2018), showing its effectiveness in decoding labels in flat NER.

CRFs

In flat NER, the most popular label decoder is CRF, which has achieved state-of-the-art performance on flat NER coupled with LSTM (Ma and Hovy, 2016; Lample et al., 2016) on CoNLL2003. Moreover, Zhuo et al. (2016) proposed a gated recursive semi-Markov CRF for segment-level sequence tagging tasks without adopting a tagging scheme, thus enabling label decoding on the segment level. Furthermore, Ye and Ling (2018) proposed a novel hybrid semi-Markov CRF model to enable label prediction on the segment level by employing word-level and segment-level information simultaneously.

With these neural NER components, we split neural-based flat NER work into four categories: transfer learning, active learning, multi-task learning and attentive methods.

2.2.3.3 Neural Transfer Learning

The idea of transfer learning is to take advantage of the knowledge acquired from one task and transfer it to another task where training data is limited. Data limitation could be due to the data being rare, the data being expensive to collect and label, or the data being non-available (Weiss et al., 2016). The typical way to transfer knowledge is to use the trained model from one task as the starting point for training the model in another task, thus avoiding learning from scratch, especially when the data is insufficient. In NLP, transfer learning is also called *domain adaptation*. In learning-based methods for flat NER, bootstrapping is the most popular method for flat NER (Kozareva, 2006; Jiang and Zhai, 2007; Wu et al., 2009). However, in neural-based methods, related work also considers sharing the parameters of neural architectures in addition to transferring knowledge. Pan et al. (2013) proposed a transfer joint embedding method that embeds both labels (outputs) and features (inputs) from different domains (i.e., a source domain and a target domain) for cross-domain flat NER. Their method is able to fully exploit the relationships between classes (labels), and reduce domain difference in data distributions for domain adaptation. Besides, Lee et al. (2017a) transferred the model trained on a large labelled corpus to alleviate the label sparsity issue in de-identifying clinical notes. As observed by Qu et al. (2016), related named entity types from different domains often share lexical and context features, which are helpful in transfer learning NER. Based on this observation, Qu

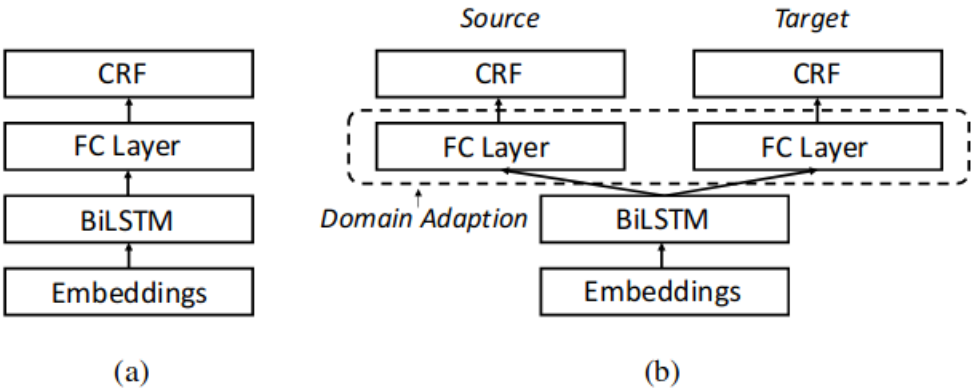

Figure 2.14: The model architectures for mention detection and classification. a) Single-task model. b) Multi-task model with domain adaptions. (Zhao et al., 2018).

et al. (2016) presented a method where, given training data in a related domain with similar (but not identical) entity types and a small amount of in-domain training data, they used transfer learning to learn a domain-specific NER model. In addition, Yang et al. (2017) utilised annotations (e.g., POS) from one task to improve flat NER where annotations are fewer, showing the effectiveness in transfer knowledge across domains. Based on Yang et al. (2017), von Däniken and Cieliebak (2017) incorporated additional labelled data, which are different from the entity labels in the target task and then improved the performance on Twitter data (WNUT2017 Named Entity Recognition challenge) by considering sentence-level features. Recently, Zhao et al. (2018) proposed a multi-task model which was based on the LSTM-CRF architecture for transfer learning. As shown in Figure 2.14, two fully connected layers are stacked on top of a LSTM context encoder, and they are jointly trained to address data heterogeneity between target and source data sets. The output from each fully connected layer is further fed into its cascading CRF label decoder, which additionally incorporates an external knowledge base that contains entity aliases built from Wikipedia articles (Radford et al., 2015; Dalton et al., 2014) to guide the decoding process at the document level.

2.2.3.4 Neural Active Learning

Classic active learning algorithms are well studied and achieved encouraging performances (Settles and Craven, 2008; Dasgupta et al., 2005; Laws and Schätze,

2008; Kim et al., 2006), which generally employ a range of heuristic procedures to select examples with least uncertainty. However, these heuristic rules are not well generalized to neural network-based models which can achieve state of the art in flat NER. Furthermore, classic active learning requires retraining the model during each iteration as it augments new examples with least uncertainty, making neural models highly computationally expensive when combining active learning algorithms. To address this, Shen et al. (2017) proposed the maximum normalised log-probability (MNLP) method, which calculates the maximum log-probability normalised on sentence lengths. MNLP ranks unlabeled sentences based on the uncertainty in their predictions. To speed up the iterative retraining, Shen et al. (2017) used CNN-based word encoder and label decoder in their model, which was updated only on the new incremental data in a batch rather than retraining from scratch. Taking advantage of MNLP, their use of active learning combined with NNs achieved state of the art with much less training data. As an extension, Lowell et al. (2018) further proved the effectiveness of MNLP on random sampling from the native inference model. However, they also observed that MNLP non-native models are not suitable for active learning.

2.2.3.5 Neural Multi-task Learning

Multi-task learning (MTL) is defined as: "Given m learning tasks $\mathcal{T}_{i=1}^{m}$ where all the tasks or a subset of them are related, multi-task learning aims to help improve the learning of a model for task $\mathcal{T}_i$ by using the knowledge contained in the m tasks. (Zhang and Yang, 2017). Instead of modelling each task individually, MTL deals with all tasks at once through exploiting commonalities and differences across all tasks and then leverage them to each task. MTL has been widely used in NLP (Miwa and Bansal, 2016; Dong et al., 2015; Søgaard and Goldberg, 2016; Stratos, 2017; Liu et al., 2018; Przybya et al., 2019). As the pioneering work in neural flat NER, Collobert et al. (2011a) proposed a window-based neural method that used the last layer which was task-specific to jointly model POS tagging, chunking, NER, and semantic role labelling on the sentence level. To enable joint training, the losses from all tasks were averaged, which collected the information of each task, and then the information was further propagated across tasks. Instead of using the last layer for task-specific purposes, Søgaard and Goldberg (2016) presented a multi-task learning architecture which uses different layers for tasks with different levels. That is, they used lower layers to model

lower-level tasks (POS tagging) and the upper layer to model higher-level tasks (chunking and combinatory categorial grammar (CCG) super tagging). This design was consistently proven helpful in their experimental results, providing a new way to make use of the shared representation of the lower-level tasks in higher-level tasks. In another empirical study, Changpinyo et al. (2018) showed that jointly learning all eleven tasks improves upon either independent or pairwise learning of the tasks. Furthermore, they observed that representations produced by their MTL approaches could reveal the natural clustering of semantic and syntactic tasks.

In addition to modeling flat NER with other sequence labelling tasks (e.g., POS tagging), joint model flat NER and its downstream task such as relation extraction, co-reference resolution are helpful for all tasks (Cai and Strube, 2010; Miwa and Bansal, 2016; Lee et al., 2017b; Kolitsas et al., 2018; Le and Titov, 2018). More recently, Luan et al. (2019) proposed a general framework to jointly extract entities, their relations and co-references, achieving state-of-the-art performance.

2.2.3.6 Neural Attentive Methods

An attention mechanism allows the model to consider each instance (e.g., word) representation composed in the input with different weights when making a decision, thus enabling the model to focus on the parts that contribute most to decision making. There are many studies that introduce attention mechanisms for flat NER (Rei et al., 2016; Zhang et al., 2018a; Cao et al., 2018; Zukov-Gregoric et al., 2017). Attention can be applied to produce better input representations or enrich context representation. To produce better input representation, Rei et al. (2016) employed attention to dynamically decide how much information to use from a word- or character-level component. Zhang et al. (2018a) employed attention to incorporate information on the document level besides the sentence-level local information to augment input representations which are further fed to the context encoder. To better encode context representation, Luo et al. (2017) stacked one attention layer on top of the LSTM-based context encoder, to incorporate document-level global information thus achieving tagging consistency across multiple instances of the same token in a document. To incorporate document-level information, Xu et al. (2018b) proposed a novel architecture which is composed of three components: one embedding layer as the word encoder, a stacked LSTM

as the context encoder which introduced a language model to obtain document-level information, and the CRF label decoder. Specifically, after using the first LSTM layer which takes the output from word encoder to obtain the local context representation, another language model is employed to produce the representation for each sentence in the document. Subsequently, an attention mechanism is used to calculate the global context representation, which is a weighted sum of all the sentence representations. The global context in the document can supply extra useful information to each word. Then, the global context representation is concatenated with the local context representation for each word. As a result, each word representation encodes both sentence- and document-level information, which is further fed to the second LSTM layer to produce the output. Finally, the typical CRF label decoder is used to predict the label for each word. Beyond text information, Zhang et al. (2018c) proposed a novel multi-modal model for tweets flat NER, which considers the image posted by users through an adaptive co-attention network to decide whether to attend to the image and to which regions of the image. With the advent of BERT (Devlin et al., 2019)which achieves the state of art in many NLP tasks, many studies use it by stacking one more tack-specific layer on top of BERT or downstream tasks including NER, obtaining state-of-the-art performance.

2.3 Overview of Nested NER

Nested entities are quite common in many domains (Alex et al., 2007; Byrne, 2007; Wang, 2009; Màrquez et al., 2007). Although flat NER was proposed in 1995 (Grishman and Sundheim, 1996), nested NER was first addressed in Shen et al. (2003). Note that nested NER is different from nested automatic term recognition which in general requires only location of terms without categorisation (Marciniak and Mykowiecka, 2015). In particular, nested terms are lexical units of smaller units within a larger lexical unit (Vintar, 2004). They are selected based on their frequencies in the specific corpus. Entities, however, can be defined as customised and their frequencies are unnecessarily top in the corpus. Early work focused on nested NER mainly used conventional learning-based methods coupled with hand-designed rules to extract nested entities. Such methods are referred to as hybrid methods. Rules designed for nested NER are generally expensive and time-consuming to obtain. Some studies adopted ML algorithms coupled with

feature sets to extract those cascaded entities. In addition, related shared tasks (Benikova et al., 2014; Ji et al., 2015, 2016) were held to advance the state of the art in nested NER. Later on, the success of neural-based methods has boosted the performance of flat NER (Lample et al., 2016; Ma and Hovy, 2016; Gridach, 2017; Strubell et al., 2017) without any hand-crafted features and external knowledge resources. This greatly encouraged the NLP community to adopt neural-based methods for nested NER. Contrary to flat NER, less emphasis has been placed on nested NER. Existing work towards nested NER can be roughly divided into three categories, namely: hybrid, conventional learning-based and neural-based methods. We present each category in the following sections.

2.3.1 Hybrid Methods

Hybrid methods towards nested NER generally employed ML algorithms to extract inner entities and then used rule-based methods to obtain outer entities. Shen et al. (2003) firstly dealt with nested entities by adopting the HMM model to biomedicine based on the GENIA corpus (Kim et al., 2003) to extract flat entities. Then 102 manually designed rules were developed to classify cascaded entities. Similarly, Zhou et al. (2004) applied an HMM-based method to recognize inner/embedded and flat entities and then employed a pattern-based post-processing step to automatically extract rules from the training data to deal with the cascaded entities that contain the flat entities as substrings. In the GENIA corpus, the patterns designed by Zhou et al. (2004) were listed as follows:

- <ENTITY>:= <ENTITY>+ head noun, e.g. <PROTEIN>binding motif → <DNA>
- <ENTITY>:= modifier + <ENTITY>, e.g. anti <PROTEIN>→ <PROTEIN>
- <ENTITY>:= <ENTITY>+ <ENTITY>, e.g. <LIPID><PROTEIN>→ <PROTEIN>
- <ENTITY>:= <ENTITY>+ word + <ENTITY>, e.g. <VIRUS>infected + <MULTICELL>→ <MULTICELL>
- <ENTITY>:= modifier + <ENTITY>+ head noun
- <ENTITY>:= <ENTITY>+ <ENTITY>+ head noun

Similarly to Zhou et al. (2004), Zhang et al. (2004) also applied an HMM-based model to extract innermost and flat entities and designed a similar set of patterns which help recognize the outermost entities based on the embedded ones. They used the top four patterns shown above as a basis and combined them iteratively to automatically construct the rules from the training corpus. In addition to the rule-based method, they also proposed an HMM-based model to extract outermost entities by reconstructing the corpus where inner entities are normalised with the corresponding entity types. Experimental results in Zhang et al. (2004) showed that the rule-based method outperformed the HMM method in extracting the outermost entities since there were not enough outermost/cascaded entities to train the HMM-based method. However, the HMM method is more general and its performance can be improved when more data is available while the rule-based method is neither flexible nor easily adaptable to new domains.

2.3.2 Learning-based Methods

Unlike hybrid methods, learning-based methods rely on ML algorithms to extract cascaded entities. McDonald et al. (2005) formulated the cascaded entity extraction as a structured multi-label classification problem, where each word composed of entities can be assigned with multiple labels. One drawback of this method was that the label assigned to each word was decided for the instance in hand, which leads to exponential label decisions when a word is part of a nested entity. To address this problem, Gu (2006) instead modelled it as a binary classification task with SVM, using a one-vs-rest scheme. Gu (2006) trained two separate SVM classifiers, one for outermost entity extraction and the other one for inner entity extraction. All the mentioned methods failed to consider the interactions between inner entities and their outer ones. Interactions mean that occurrences of inner entities are informative indicators for the occurrence of outer entities. To consider such interactions, Alex et al. (2007) designed the cascading approach, which first grouped one or more entity types and then cascaded separate CRFs to train each group by using the output from previous CRFs as features for the current CRFs, yielding the best performance. The main drawback of their approach was that it failed to handle nested entities sharing the same entity type, which are quite common in natural language. Another drawback was that the grouping of entity types requires extensive experimentation to decide the best combination/group. Apart from the cascading approach, they additionally built an inside-out and

outside-in layer CRFs, where each level of nesting was modelled as a separate BIO problem without incorporating the interactions between entities on different nesting levels. In the layer approach, entities on higher nesting levels were more difficult to model due to the sparsity issue.

Finkel and Manning (2009) proposed a discriminative constituency tree to represent each sentence where the root node was used for connecting the entire sentence. All entities were treated as phrases and represented as subtrees following the whole tree structure. To encode those features, Finkel and Manning (2009) used a CRF-based approach driven by entity-level features to detect nested entities.

More recently, Lu and Roth (2015) proposed novel mention hypergraphs to compactly represent mention candidates, which enable the joint learning of boundaries, entity types and head information of candidates. A mention hypergraph is a type of conventional graph, whose edge (i.e., hyperedge) consists of nodes that represent semantic types and boundaries. Every sentence is represented as a complete hypergraph, where each mention candidate constitutes a subset of the hypergraph i.e., sub-hypergraph. Figure 2.15 provides an example of such a sub-hypergraph, which represents two nested entity mentions. Driven by a set of features including POS tags, word n-grams, bags of words, word patterns, a CRF-like log-linear method was applied to extract nested and flat entities. One problem with their approach was the spurious structures of hypergraphs as they enumerate combinations of nodes, types and boundaries to represent entities. Moreover, they failed to encode the interactions among embedded entities using hyper-graphs. To overcome spurious structures, Muis and Lu (2017) further incorporated mention separators along with features to yield better performance on nested entities. Both Lu and Roth (2015) and Muis and Lu (2017) relied on hand-crafted features to extract nested entities without incorporating hidden dependencies in nested entities.

2.3.3 Neural-based Methods

Utilizing the ability of neural models in extracting features, Xu et al. (2017b) pioneered neural-based nested NER with the fixed-size ordinally-forgetting encoding (FOFE) (Zhang et al., 2015) for text span representations, as shown in Figure 2.16. Given a sequence of words $S = w_1, w_2, \cdots, w_T$, FOFE represents each word w_t using a 1-of-K representation e_t, which accumulates information from the first

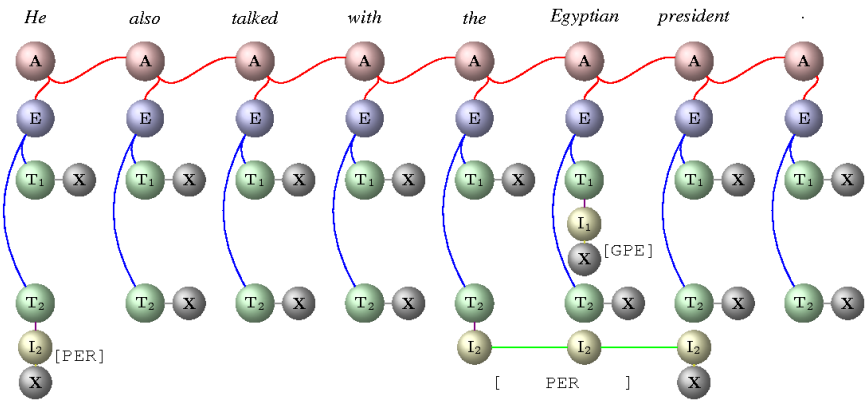

Figure 2.15: An example sub-hypergraph structure for jointly representating all the three entities contained in the sentence. "He also talked with the Egyptian president. For simplicity and the ease of illustration, we assume there are only two possible mention types: PER and GPE (Lu and Roth, 2015).

word ($t = 1$) to the last word ($t = T$) of the sentence, based on the following recursive formula (with $z_0 = 0$):

$$z_t = \alpha \cdot z_{t-1} + e_t (1 \leq t \leq T) \tag{2.6}$$

where z_t denotes the FOFE code for the partial sequence up to w_t, and $\alpha (0 < \alpha < 1)$ is a constant forgetting factor to control the influence of the history on the current position. All the possible entity candidates (length is up to n words) along with their contexts were represented using this novel tagging scheme. Unlike the extensively used LSTM-RNNs in sequence labelling task, a feed-forward neural network was used to predict labels on entity level for each fragment in any of the given sequences.

Additionally, Li et al. (2017a) used the model proposed in Lample et al. (2016) to extract both flat entities and components composed in nested and discontinuous entities. Then, another LSTM was applied to combine the components to get nested and discontinuous entities. However, these methods failed to capture and utilize the inner entity representation to facilitate outer entity detection.

More recently, Katiyar and Cardie (2018) represented each tag sequence in the single hypergraph structure of Figure 2.17 and then designed an LSTM-based method that produces the correct nested entity hypergraph for a given input sentence. Wang and Lu (2018) proposed to use neural networks to produce segmental hypergraph representations to model overlapping entity mentions. Wang

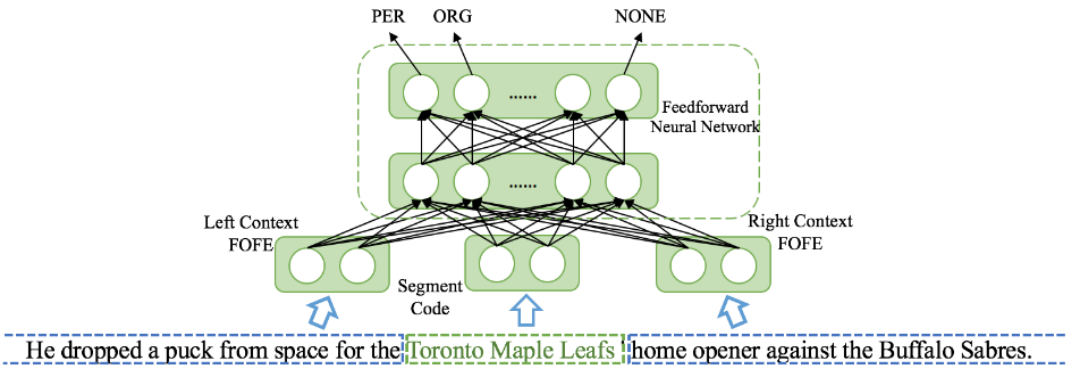

Figure 2.16: The model architecture (Xu et al., 2017b). The window currently examines the fragment of "Toronto Maple Leafs". The window will scan and scrutinize all fragments up to K words.

et al. (2018a) introduced a scalable transition-based method to model the nested structure of mentions. With this method, a sentence with nested mentions is firstly mapped to a designated forest where each mention corresponds to a constituent of the forest, shown in Figure 2.18. Based on the shift-reduce parser for constituency parsing (Watanabe and Sumita, 2015), a shift-reduce based system was then used to learn to construct the forest structure in a bottom-up manner through an action sequence whose maximal length is guaranteed to be three times of the sentence length. Based on Stack-LSTM, which is employed to efficiently and effectively represent the states of the system in a continuous space, our system is further incorporated with a character-based component to capture letter-level patterns. Sohrab and Miwa (2018) enumerated all possible regions or spans as potential entity mentions and classified them with deep NNs. To reduce the computational costs and capture the information of the contexts around the regions, the model represents the regions using the outputs of shared underlying the LSTM. Marinho et al. (2019) designed a hierarchical model based on a transition-based parser that is able to recognize hierarchical and nested mentions with undefined levels of complexity.

2.4 Resources and Tools

There are many resources including data, knowledge bases and systems that are built for neural NER. We summarised the common data sets in Table 2.2. Table 2.3 lists the tools for flat and nested NER.

Corpus	Text Source	URL
MUC-6	Wall Street Journal texts	https://catalog.ldc.upenn.edu/LDC2003T13
MUC-7	New York Times news	https://catalog.ldc.upenn.edu/LDC96T10
CoNLL2003	Reuters news	https://www.clips.uantwerpen.be/conll2003/ner/
GENIA	MEDLINE abstracts	http://www.geniaproject.org/home
GENETAG	MEDLINE abstracts	http://www.geniaproject.org/home
ACE2004	Transcripts, news	https://catalog.ldc.upenn.edu/LDC2005T09
ACE2005	Transcripts, news	https://catalog.ldc.upenn.edu/LDC2006T06
OntoNotes	Magazine, news, conversation, web	https://catalog.ldc.upenn.edu/LDC2013T19
JNLPBA	MEDLINE abstracts	http://www.nactem.ac.uk/tsujii/GENIA/ERtask/report.html
NYT	New York Times texts	https://catalog.ldc.upenn.edu/LDC2008T19
FSU-PRGE	PubMed and MEDLINE	https://julielab.de/Resources/FSU_PRGE.html
NCBI-Disease	PubMed	https://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/
BC5CDR	PubMed	http://bioc.sourceforge.net/
COPD	PubMed full-text articles	http://www.nactem.ac.uk/COPD/index.php
NeuroScience	PubMed abstracts	https://github.com/nactem/TM4NS
Quaero Broadcast News	news and broadcast conversation	http://catalog.elra.info/en-us/repository/browse/ELRA-S0349/
Quaero Old Press	newspaper issues, extracted pages in text format	http://catalog.elra.info/en-us/repository/browse/ELRA-W0073/

Table 2.2: Available corpora in NER. Corpora that contain nested entities are in bold.

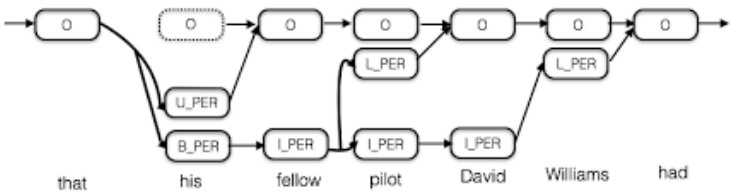

Figure 2.17: A directed hypergraph. Curved edges represent hyperarcs and straight edges are normal edges (Katiyar and Cardie, 2018).

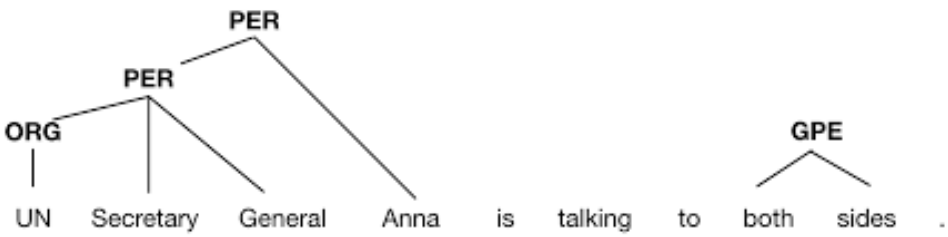

Figure 2.18: An example sentence of nested mentions represented in the structure of forest. PER:Person,ORG:Organization, GPE:Geo-Political Entity (Wang et al., 2018a).

Tool	URL
StanfordCoreNLP	https://stanfordnlp.github.io/CoreNLP/
OSU Twitter NLP	https://github.com/aritter/twitter_nlp
Illinois NLP	http://cogcomp.org/page/software/
NeuroNER	http://neuroner.com/
NERSuite	http://nersuite.nlplab.org/
Polyglot	https://polyglot.readthedocs.io/
Gimli	http://bioinformatics.ua.pt/gimli
spaCy	https://spacy.io/
NLTK	https://www.nltk.org/
OpenNLP	https://opennlp.apache.org/
LingPipe	http://alias-i.com/lingpipe-3.9.3/
AllenNLP	https://allennlp.org/models
IBM Watson	https://www.ibm.com/watson/

Table 2.3: Available tools in NER.

2.5 Evaluation Metrics

To examine the performances of models/systems in both flat and nested NER, several sets of evaluation metrics are employed for model measurement.

2.5.1 Precision, Recall and F-score

Precision, recall and F-score are calculated based on the numbers of true positives (TP), false positives (FP) and false negatives (FN), as shown in the following equations. TP and FP are the numbers of instances in the data set which are correctly and incorrectly identified by models/systems, respectively. FN is the number of instances in the data set which are incorrectly rejected by models/systems. Precision reflects how many predictions of the model/system are correct while recall measures how many gold entities are corrected predicted by the model/system. F1-score is the harmonic mean of precision and recall.

$$\text{Precision (P)} = \frac{TP}{TP + FP} \tag{2.7}$$

$$\text{Recall (R)} = \frac{TP}{TP + FN} \tag{2.8}$$

$$\text{F1-score (F1)} = 2 \times \frac{P \times R}{P + R} \tag{2.9}$$

2.5.2 Strict and Lenient Matching

One entity is considered as correct only if its text span and boundary are matched against the corresponding gold standard. This matching manner is referred to as strict matching. CoNLL2003 (Tjong Kim Sang and De Meulder, 2003) and JNLPBA (Kim et al., 2004a) are examples of strict matching. To better measure how well the model can make decisions of entity boundaries regardless of semantic types, the MUC-6 (Grishman and Sundheim, 1996) defined the lenient matching, which additionally considers predictions that are partially overlapping with the gold standard as correct.

2.5.3 Macro and Micro Metrics

Micro-averaged metrics are calculated by summing up all TP, FP, and FN of the model for different sets of predictions (e.g. different sets of entity types) and are further used to compute precision, recall and F-score, namely Micro-averaged precision, Micro-averaged recall, and Micro-averaged F-score, respectively. Different from Micro metrics, the calculation of Macro-averaged is quite straightforward,

which is the corresponding average of a set of metrics. For example, the Macro-average precision is the average of the set of precisions.

2.5.4 Cross Validation

To evaluate the generalization ability of the model to an unseen data set, one approach is to split the data into K (K is an integer) chunks, and then keep one chunk for model validation while the remaining is for training. This process is repeated for each chunk, known as K-fold cross-validation. The common value of K is 10, which has been widely used in many studies (Ekbal and Bandyopadhyay, 2007; Wang and Patrick, 2009; Ekbal and Bandyopadhyay, 2010; Xu et al., 2015).

2.5.5 Extended Metrics

To investigate the effectiveness of layered models on different nested levels of entities, Ju et al. (2018) proposed the layer-wise evaluation. Under this setting, entities which are predicted in previous layers during evaluation are removed. Predictions and gold entities from the corresponding layer are collected to calculate precision, recall and F-score. Since predicted entities on a specific layer might be from other layers, Ju et al. (2018) defined extended precision (EP), extended recall (ER) and extended F-score (EF1) to measure the performance. In detail, EP is calculated by comparing the predicted entities in a specific layer with all the gold entities, and ER is calculated by comparing the gold entities in a specific layer with all the predicted entities. EF1 is calculated in the same way as F-score. The calculation of extended evaluation metrics are shown as follows:

$$\text{Extended Precision (EP)} = \frac{TP_e}{TP_e + FP_e} \tag{2.10}$$

$$\text{Extended Recall (ER)} = \frac{TP_e}{TP_e + FN_e} \tag{2.11}$$

$$\text{Extended F1-score (EF1)} = 2 \times \frac{EP \times EF}{EP + EF} \tag{2.12}$$

where TP_e, FP_e and FN_e are defined as: TP_e: the number of instances from the corresponding layer which are correctly predicted by models/systems; FP_e: the number of instances from the corresponding layer which are incorrectly predicted by models/systems; FN_e: the number of instances in the corresponding

layer which are incorrectly rejected by models/systems. Extended metrics are lenient metrics that reflect the model to layer-wise ability in predicting different nesting levels of entities.

2.6 Summary

In this chapter, we have provided an overview of related work for both flat and nested NER including resources, tools, methodologies in terms of timeline. The detailed evaluation metrics were also discussed. One of the drawbacks in related work is that most NER work only deal with flat entities but ignore nested ones. This leads to information loss as nested entities contain rich and fine-grained information, thus negatively impacting downstream NLP tasks. There are some efforts to deal with nested entities. However, they ignored the interactions between nested entities, which convey informative clues. To consider such clues, we describe our methodology in the next chapter.

Chapter 3

Methodology

In response to our second research question (*RQ2*) and the corresponding hypothesis (*H1*):

RQ2: How nested NER can be improved using NNs?
H1: Utilisation of inner entities can improve the detection of outer entities using NNs.

we describe our methodology of nested named entity recognition (NER). The method described in this chapter has been published at 2018 NAACL-HLT (Ju et al., 2018).

3.1 Introduction

Work on nested NER, which was studied before our research, ignored the dependencies between nested entities, which are informative clues for detecting nested entities. Dependencies refer to the occurrences of inner entities are informative indicators to the occurrences of outer entities. In other words, when inner entities appear in the texts, outer entities only appear in the places that cover inner entities as their substrings. For example, in Figure 3.1, the innermost entities *"western Canadian"* and *"British Columbia"* are indicators of the outer entity *"the western Canadian province of British Columbia"*. Similarly, the outer entity *"the western Canadian province of British Columbia"* is the indicator of outermost entity *"The primier of the western Canadian province of British Columbia"*. Based on this observation, we hypothesize that interactions between nested entities are helpful in detecting nested entities. Neural approaches have enabled

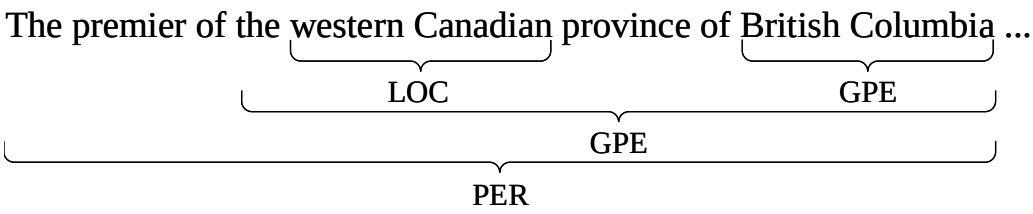

Figure 3.1: A sentence from ACE2005 (Walker et al., 2006) containing the nested 4 entities nested 3 levels deep.

NER without depending on hand-crafted engineering and external knowledge resources. Therefore, we take advantage of neural networks (NNs) to design our model. We cast nested NER as a sequence labelling problem and detail the model in the next section.

Our nested NER model is designed based on a sequential stack of flat NER layers that detects nested entities in an end-to-end manner. Figure 3.2 provides the overview of our model. Our flat NER layers are inspired by the state-of-the-art model proposed in Lample et al. (2016). The layer utilises one single BiLSTM layer to represent word sequences and predict flat entities by putting one single CRF layer on top of the LSTM layer. Therefore, we refer to our model as layered-BiLSTM-CRF model. If any entities are predicted, a new flat NER layer is introduced and the word sequence representation of each detected entity by the current flat NER layer is merged to compose a representation for the entity, which is then passed on to the new flat NER layer as its input. Otherwise, the model terminates stacking and hence finishes entity detection. Our model enables sequentially stacking flat NER layers from bottom to top and identifying entities in an end-to-end manner. The number of stacked layers depends on the level of entity nesting and dynamically adjusts to the input sequences as the nested level varies from different sequences.

Next, we provide the description of the model architecture: the flat NER layers and their stacking, the word encoder (i.e., embedding layer) and their training.

3.2 Flat NER Layer

A flat NER layer contains two components: an LSTM-based context encoder and a CRF label decoder. The LSTM-based context encoder captures the bidirectional context representation of sequences and subsequently feeds it to the label

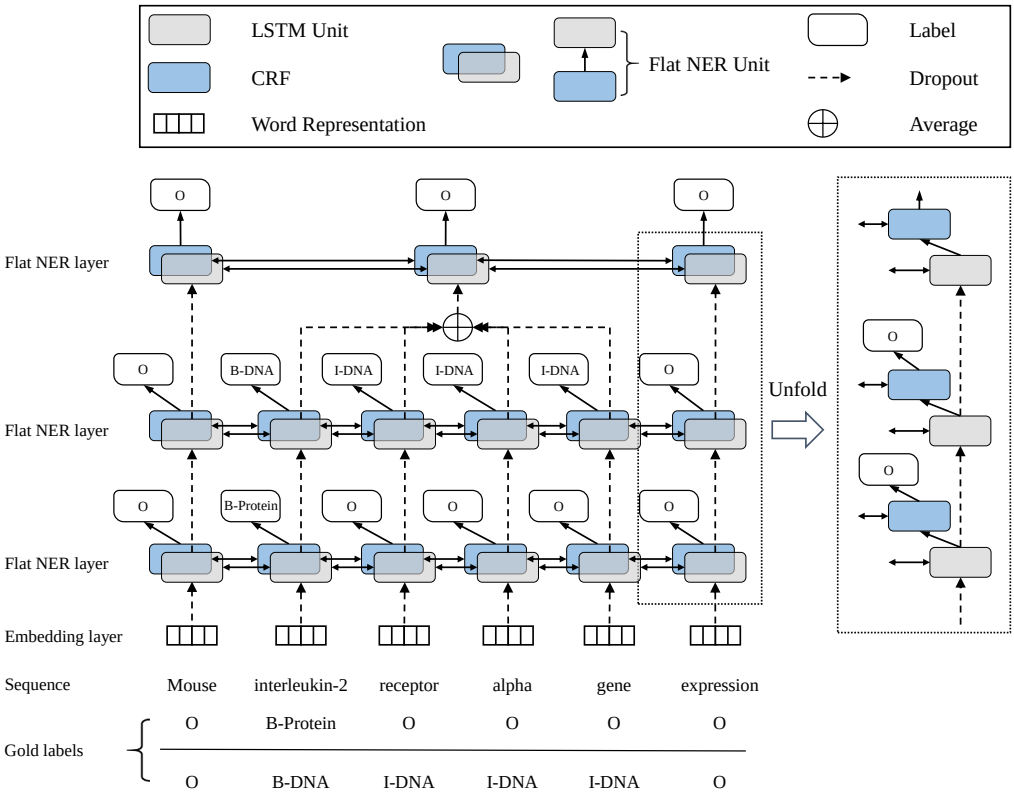

Figure 3.2: Overview of our layered model architecture. "interleukin-2" and "interleukin-2 receptor alpha gene" are nested entities.

decoder to globally decode label sequences. The details of LSTM and context encoder are described in 2.2. CRFs are used to globally predict labels for any given sequences. Given an input sequence $X = (x_1, x_2, \ldots, x_n)$ which is the output from the LSTM context encoder, we maximise the log-probability during training. In decoding, we set transition costs between illegal transitions, e.g., transition from O to I-PER, as infinite to restrict illegal labels. The expected label sequence $y = (y_1, y_2, \ldots, y_n)$ is predicted based on maximum scores in decoding. Please refer to Section 2.2 for the detail of label decoder.

3.3 Stacking Flat NER Layers

We stack a flat NER layer on the top of the current flat NER layer, aiming to extract outer entities. Specifically, we average the current context representation

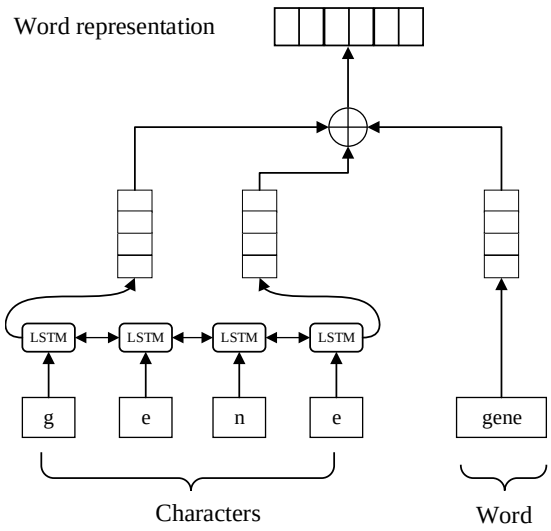

Figure 3.3: Word representation of a word 'gene'. We concatenate the outputs of character embedding from LSTM and word embedding to obtain its final word representation. The embedding layer is our word encoder.

of the regions composed of the detected entities, as described in the following equation:

$$m_i = \frac{1}{end - start + 1} \sum_{i=start}^{end} z_i, \tag{3.1}$$

where z_i denotes the representation of the i-th word from the flat NER layer, and m_i is the merged representation for an entity. The region starts from a position *start* and ends at a position *end* of the sequence. This merged representation of detected entities allows us to treat each detected entity as a single token, and hence we are able to make the most of inner entity information to encourage outer entity recognition. If the region is detected as a non-entity, we keep the representation without any processing. The processed context representation of the flat NER layer is used as the input for the next flat NER layer.

3.4 Word Encoder

The input for the first NER layer is different from the remaining flat NER layers since the first layer has no previous layers. We thus represent each word by concatenating character sequence embeddings and word embeddings for the first flat NER layer. Figure 3.3 describes the architecture of the embedding layer to

produce the word representation.

Following the successes of Ma and Hovy (2016) and Lample et al. (2016) in utilising character embeddings on the flat NER task, we also represent each word with its character sequence to capture the orthographic and morphological features of the word. Each character is mapped to a randomly initialized vector through a character lookup table. We feed the character vectors comprising a word to a bidirectional LSTM layer and concatenate the forward and backward representation to obtain the word-level embedding.

Unlike the character sequence embeddings, the pretrained word embeddings are used to initialise word embeddings. When evaluating or applying the model, words that are outside of the pretrained embeddings and training data set are mapped to an unknown (UNK) embedding, which is randomly initialised during training. To train the UNK embedding, we replace words whose frequency is 1 in the training data set with the UNK embedding with a probability 0.5.

3.5 Model Variants

When preparing the input for each flat NER layer, we also designed another two different ways, resulting in two different models. The first model variant, depicted in Figure 3.4 is called *layered-BiLSTM-CRF w/o layered out-of-entities (LBCWLE)* which uses the input of the current flat NER layer for out-of-entity words. We name the second model as *layered-BiLSTM-CRF w/o layered LSTM (LBCWLL)* as it skips all intermediate LSTM layers and only uses the output of the embedding layer to build the input for the next flat NER layer. In the LBCWLL model, depicted in Figure 3.5, we merge and average representations following Equation 3.1. For the predicted non-entity words, however, we skip the LSTM layer and directly use their corresponding representation from the input rather than the output context representation.

3.6 Training

We prepare the gold labels based on the conventional BIO (Beginning, Inside, Outside of entities) tagging scheme to represent a label attached to each word. Please refer to Section 2.2 of Chapter 2 for the details of BIO scheme. As our model detects entities from inside to outside, we keep the same order in preparing

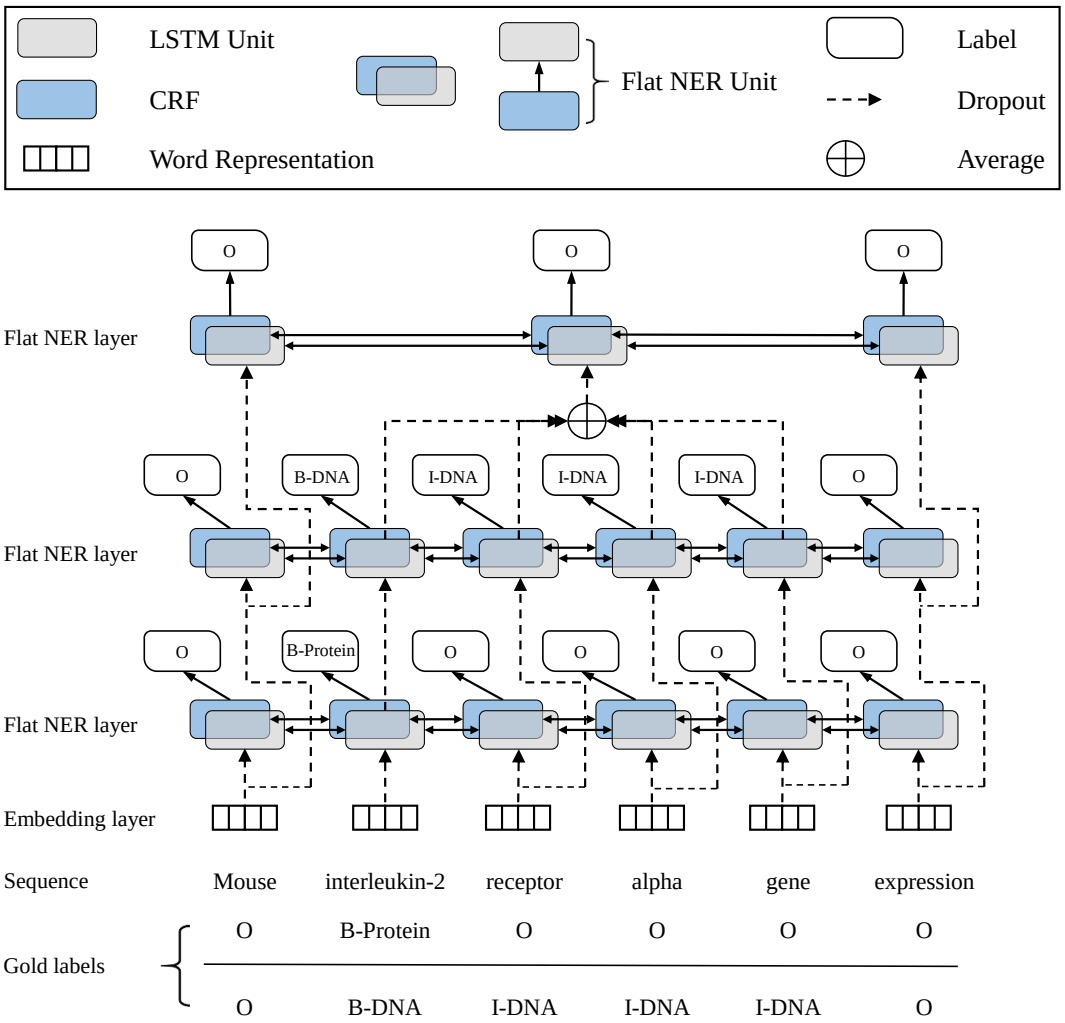

Figure 3.4: Overview of the layered-BiLSTM-CRF w/o layered out-of-entities (LBCWLE) architecture with skipping representation for non-entity words. "interleukin-2" and "interleukin-2 receptor alpha gene" are nested entities.

the gold labels for each word sequence. We call it the *detection order rule*. Meanwhile, we define that each entity region in the sequence can only be tagged once with the same entity type, referred to as the *non-duplicate rule*. For instance, in Figure 3.2, *"interleukin-2"* is tagged first while *"interleukin-2 receptor alpha gene"* is subsequently tagged following the above two rules. When assigning the label *O* to non-entity regions, we only follow the detection order rule. As a result, two gold label sequences {O, B-Protein, O, O, O, O} and {O, B-DNA, I-DNA, I-DNA, I-DNA, O} are assigned to the given word sequence *"Mouse interleukin-2 receptor alpha gene expression"* as shown in Figure 3.2. With these rules, the

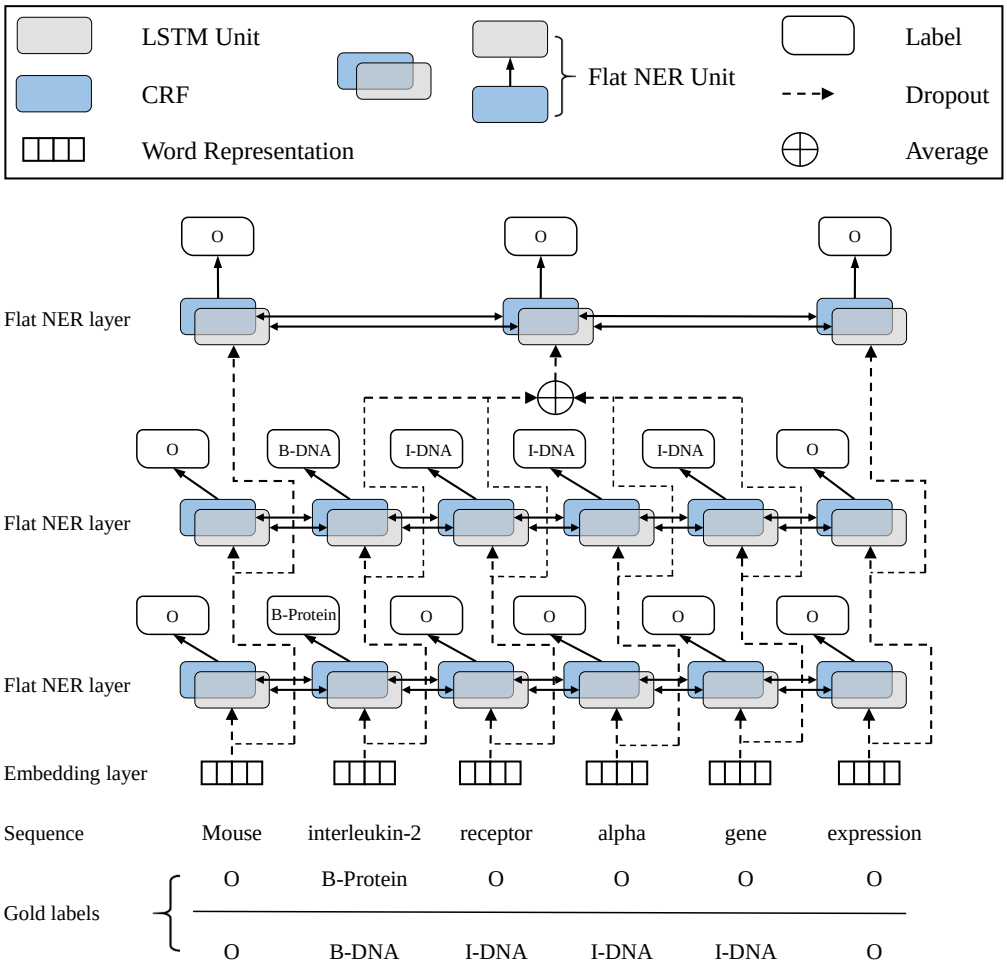

Figure 3.5: Overview of the layered-BiLSTM-CRF w/o layered LSTM (LBCWLL) architecture with skipping representation for the whole sequence. "interleukin-2" and "interleukin-2 receptor alpha gene" are nested entities.

number of labels for each word equals the nested level of entities in the given word sequence.

We employ mini-batch for training. For each sentence in one mini-batch, we accordingly pad label sequences (i.e., a sequence of label 'O') based on the maximum nesting level in the current mini-batch. In other words, each sentence in the mini-batch has the same number of label sequences. The maximum number of stacking flat NER layers equals one plus the maximum nesting level in the training set. To avoid spurious flat NER layers, we set the upper limit of stacking flat NER layers that the model can stack during training.[1]

[1] Although we set the upper limit of stacking flat NER layers, our model always generates

We update the model parameters using back-propagation through time (Werbos, 1990) with Adam (Kingma and Adam, 2015). The model parameters include weights, bias, transition costs, and embeddings of characters. We disable updating the word embeddings.[2] During the training stage, early stopping, L2-regularization and dropout (Hinton et al., 2012) are used to prevent over-fitting. Dropout is employed to the input of each flat NER layer. Hyper-parameters including batch size, number of hidden units in LSTM, character dimensions, dropout rate, Adam learning rate, gradient clipping and weight decay (L2) are all tuned with Bayesian optimization (Snoek et al., 2012).

3.7 Comparison

In this chapter, we describe our novel nested NER model which extracts nested entities in an end-to-end manner, without depending on external knowledge bases. In comparison with related work[3], our model identifies nested entities through automatic high-level and abstract feature learning while related work requires feature-engineering (Byrne, 2007; Wang, 2009; Finkel and Manning, 2009). In addition, our layered model extracts outer entities by using inner entities from previous layers while other layered approaches (Alex et al., 2007) developed separate models for each nesting level of entities without considering the interactions between nested entities. To incorporate such interactions, Alex et al. (2007) additionally cascaded the models to extract nested entities by using previous predictions as features. However, they failed to extract nested entities that share the same entity type. Such entities are considered as special nested entities in our model. Moreover, Alex et al. (2007) fixed the number of models to be trained to extract nested entities while our model dynamically decides the number of stacking layers based on the predictions from previous layers.

3.8 Summary

In this chapter, we describe our novel nested NER model which extracts nested entities in an end-to-end manner, without depending on external knowledge bases

the same or fewer layers than the maximum nesting level in the training set.

[2]We tried updating and disabling updating word embeddings. The former trial did not work.

[3]We only compared with work that was done before our work.

and hand-crafted feature-engineering. The idea of the model is to make the most of dependencies of nested entities in our model to encourage outer entity recognition by automatic learning of high-level and abstract features from sequences. To evaluate the general ability and effectiveness of the model, we evaluate our model with different different domains of datasets (i.e., data-based evaluation). In addition, we further evaluate our model with the task-specific setting to validate its applicability in NLP tasks. Those settings will be discussed in the next chapter.

Chapter 4

Evaluation

In this chapter, we evaluate our layered BiLSTM-CRF model with different settings. Specifically, we first evaluate the model under flat and nested NER settings with different domains of data sets. Then, we further evaluate our model in different tasks to demonstrate its applicability in NLP tasks.

4.1 Data-based Evaluation

In this section, we present the experimental settings, performances and discussion of our model under flat and nested NER settings. Work presented in this section has been included in Ju et al. (2018).

4.1.1 Flat NER Setting

4.1.1.1 Evaluation Setting

Precision, recall and F1-score were used as the evaluation metrics in flat NER. Please refer to Section 2.5 in Chapter 2 for the details of evaluation metrics.

4.1.1.2 Data Sets

We employed JNLPBA (Kim et al., 2004b) for evaluation. JNLPBA defines both training and testing sets. These two data sets are composed of 2,000 and 404 MEDLINE abstracts, respectively. JNLPBA is derived from the GENIA corpus (Kim et al., 2003). However, only flat and topmost entities in JNLPBA are kept while nested and discontinuous entities are removed. Following the same settings as in (Gridach, 2017), we collapsed all DNA subcategories as DNA. The same

Item	Train	Development	Test
Sentences	16,691	1,855	3,856
Split percentage	90%	10%	-
DNA	8,649	884	1,056
RNA	863	88	118
Protein	27,263	3,006	5,067
Cell Line	3,459	371	500
Cell Type	6,045	673	1,921
Overall entities	46,279	5,022	8,662

Table 4.1: Statistics of the JNLPBA corpus.

setting was applied to RNA, protein, cell line and cell type categories. As a result, only five entity types were finally preserved. We randomly chose 90% of the sentences of the original training set as our training set and the remaining as our development set. Each sentence in the JNLPBA data set has already been tokenised, so we did not apply any other preprocessing. The statistics of the flat corpus (JNLPBA) are described in Table 4.1.

4.1.1.3 Model Setting

Our model was implemented with Chainer (Tokui et al., 2015a). We initialised word embeddings in JNLPBA with pre-trained embeddings trained on MEDLINE abstracts (Chiu et al., 2016a). We trained our flat model that only kept the first flat NER layer and removed the following stacking layers. We follow Lample et al. (2016) for hyper-parameter settings in flat NER evaluation.

4.1.1.4 Result and Discussion

Compared with the state-of-the-art work on JNLPBA test set (Gridach, 2017) which achieved 75.87% in terms of F1-score, our model obtained 75.55% F1-score. Since both the model by Gridach (2017) and our flat model are based on Lample et al. (2016), it is reasonable that both models were able to get comparable performance.

4.1.2 Nested NER Setting

4.1.2.1 Evaluation Setting

Precision (P), recall (R) and F1-score (F1) were used as the evaluation metrics in nested NER. The extended set of precision (EP), recall (ER) and F1-score (EF1) described in Section 2.5 of Chapter 2 were additionally used for nested NER evaluation. We define that if the numbers of gold entities and predictions are all zeros, the evaluation metrics all equal one hundred percent.

4.1.2.2 Data Sets

We employed two data sets for evaluation: GENIA (Kim et al., 2003), ACE2005 (Walker et al., 2006). The GENIA corpus contains 36 fine-grained entity categories among total 2,000 MEDLINE abstracts. Following the same task settings as in Finkel and Manning (2009) and Lu and Roth (2015), we applied the same preparation to collapse entity types into five entity types. We used same test portion as Finkel and Manning (2009), Lu and Roth (2015) and Muis and Lu (2017) for direct comparison. The ACE2005 corpus (Walker et al., 2006) contains 7 fine-grained entity categories. We made same the modifications described in Lu and Roth (2015) and Muis and Lu (2017) by keeping files from broadcast news (bn), broadcast conversation (bc), newswire (nw) and weblog (wl) and splitting them into training, development and testing sets at random following the same ratio 8:1:1, respectively.

We used NERSuite (Cho et al., 2010) for GENIA to perform tokenisation while Stanford CoreNLP (Manning et al., 2014) was used for ACE2005. Statistics of nested corpora (GENIA, ACE2005) are described in Table 4.2.

For GENIA, we had to manually resolve two issues, in addition to the above preprocessing. One of the issues is the removal of discontinuous entities during parsing. In the GENIA XML file, each flat entity is annotated with "lex" (lexical) and "sem" (semantics) attributes while discontinuous and nested entities may have none, one or two attributes when these entities embed with each other, making it difficult to extract the strictly nested ones. For example, in Figure 4.1, the sentence contains two flat entities and four discontinuous entities. We extract these two entities based on the symbol "*" appeared in the "lex" attribute which is a connection indicator of the separated texts in discontinuous entities. Meanwhile, each of the separated texts has no "sem" attribute unless itself is an

GENIA	Train	Dev.	Test
Documents	1,599	189	212
Sentences	15,022	1,669	1,855
Split percentage	81%	9%	10%
DNA	7,921	1061	1,283
RNA	730	140	117
Protein	29,032	2,338	3,098
Cell Line	3,149	340	460
Cell Type	6,021	563	617
Outermost entity	42,462	4,020	4,942
Nested level	4	3	3
Entities in level 1	42,846	4,060	4,991
Entities in level 2	3,910	381	569
Entities in level 3	91	1	15
Entities in level 4	1	0	0
Entity avg. length	2.87	3.13	2.93
Multi-token entity	33951	3554	4203
Overall entities	46,853	4,442	5,575

ACE2005	Train	Dev.	Test
Documents	370	43	51
Sentences	9,849	1,221	1,478
FAC	924	83	173
GPE	4,725	486	671
LOC	763	81	69
ORG	3,702	479	559
PER	13,050	1,668	1,949
VEH	624	81	66
WEA	652	94	67
Outermost entity	18,455	2,285	2,724
Nested level	6	4	5
Entities in level 1	19,676	2,429	2,936
Entities in level 2	3,934	448	505
Entities in level 3	731	85	102
Entities in level 4	90	10	10
Entities in level 5	7	0	1
Entities in level 6	2	0	0
Entity avg. length	2.28	2.33	2.28
Multi-token entity	10,577	1,323	1,486
Overall entities	24,440	2,972	3,554

Table 4.2: Statistics in GENIA and ACE2005.

innermost entity. Unfortunately, there are some inconsistent cases such as "c-fos and c-jun transcripts" where symbol "*" should be in the "lex" attribute as the discontinuous entity "c-fos transcript" is connected by "c-fos" and "transcript" while "c-jun transcript" is connected by "c-jun" and "transcript". These two entities share the same text "transcript". However, each of them is annotated with two attributes: "lex" and "sem", following the same annotation for flat entities. Although it is possible to ignore the latter entity based on "lex" attribute and its belonging sentence, this rule fails to deal with entity "c-jun gene" in the example of "c-fos and c-jun genes" as the "lex" of "c-jun gene" is mistaken as "c-jun genes". Therefore, in this case, we ignored "c-fos transcript" and instead kept the "c-jun transcripts" as a flat entity.

Another issue is the incomplete tokenisation. We assigned the label to each word on the word-level instead of character level, but there are entities that correspond to parts of words. An example is *"NF-YA subunit"*, which contains two protein entities: *"NF-Y"* and *"A subunit"*. To cope with this problem, we treat both entities as false negative entities in training set as there are only 13 such entities in the training set.

Sentence The erythroid differentiation of HEL cells was inhibited by recombinant human nm23-H1, -H2, mouse nm23-M1, and -M2 proteins.

Flat Entities human nm23-H1 | mouse nm23-M1

Discontinuous Entities mouse nm23-M2 | recombinant human-H2 protein | recombinant human nm23-H1 protein | human nm23-H2 | recombinant mouse nm23-M1 protein | recombinant mouse nm23-M2 protein

Sentence in XML

```
<sentence>The
    <cons lex="erythroid_differentiation" sem="G#other_name">erythroid differentiation</cons> of
    <cons lex="HEL_cell" sem="G#cell_line">HEL cells</cons> was inhibited by
    <cons lex="(AND recombinant_human_nm23-H1_protein
    recombinant_human-H2_protein
    recombinant_mouse_nm23-M1_protein
    recombinant_mouse_nm23-M2_protein)"
          sem="(AND G#protein_molecule G#protein_molecule G#protein_molecule G#protein_molecule)">
        <cons lex="recombinant*">recombinant</cons>
        <cons lex="(AND *human_nm23-H1* *human_nm23-H2*)" sem="(AND G#protein_molecule G#protein_molecule)">
            <cons lex="human_nm23*">
                <cons lex="human" sem="G#multi_cell">human</cons> nm23</cons>
            <cons lex="*-H1">-H1</cons>,
            <cons lex="*-H2">-H2</cons></cons>,
        <cons lex="(AND *mouse_nm23-M1* *mouse_nm23-M2*)" sem="(AND G#protein_molecule G#protein_molecule)">
            <cons lex="mouse_nm23*">
                <cons lex="mouse" sem="G#multi_cell">mouse</cons> nm23</cons>
            <cons lex="*-M1">-M1</cons>, and
            <cons lex="*-M2">-M2</cons></cons>
        <cons lex="*protein">proteins</cons></cons>.
</sentence>
```

Figure 4.1: An annotated sentence in XML format.

4.1.2.3 Model Setting

Our model was implemented with Chainer (Tokui et al., 2015a). We initialised word embeddings in GENIA with the pre-trained embeddings trained on MEDLINE abstracts (Chiu et al., 2016a). For ACE2005, we initialised each word with the pre-trained embeddings which are trained by Miwa and Bansal (2016). Except for the word embeddings, parameters of word embeddings were initialized with a normal distribution. For the LSTM, we initialized hidden states, cell state and all the bias terms as 0 except for the forget gate bias that was set as 1. For other hyper-parameters, we chose the best hyper-parameters via Bayesian optimization. For nested NER experiments, the settings of the hyper-parameters of the models and Bayesian optimisation are listed in Table 4.3 and Table 4.4, respectively.

For ablation tests, we compared with our layered-BiLSTM-CRF model with two model variants that produce the input for next flat NER layer in different ways.

4.1.2.4 Results and Discussion

Table 4.5 presents the comparisons of our model with related work including the state-of-the-art feature-based model by Muis and Lu (2017). Our model

Hyper Parameters	Range	Best (GENIA)	Best (ACE2005)
Batch size	[16 – 256]	67	91
No. of hidden units	[200, 250, 300]	200	200
Dim. of char. emb.	[15 – 50]	35	28
Dropout rate	[0.1 – 0.5]	0.2144	0.1708
Learning rate	[0.001 – 0.02]	0.00754	0.00426
Gradient clipping	[5 – 50]	27	11
Weight decay (L2)	[10^{-8} – 10^{-3}]	4.54^{-5}	9.43^{-5}

Table 4.3: Value range and best value of tuned hyper parameters in GENIA and ACE2005, respectively.

Hyper Parameters	Initialized Value
Acquisition Function	gp_hedge
n-calls	10
n_random_state	None
n_random_starts	10
Acquisition Optimizer	lbfgs
n_restarts_optimizer	100
noise	gaussian
n_points	50000
xi	0.1
n_jobs	1

Table 4.4: Hyper parameters used of Bayesian Optimization.

outperforms the state-of-the-art models with 74.7% and 72.2% in terms of F1-score, achieving a new state-of-the-art in the nested NER tasks. For GENIA, our model gained more improvement in terms of recall, which enabled extracting more nested entities without reducing precision. On ACE2005, we improved recall with 12.2 percentage points and obtained 5.1% relative error reductions. Compared with GENIA, our model gained more improvements in ACE2005 in terms of F1-score. Two possible reasons account for it. One reason is that ACE2005 contains deeper nested entities (maximum nested level is 5) than GENIA (maximum nested level is 3) on the test set. This allows our model to capture the potentially "nested" relations among nested entities. The other reason is that ACE2005 has more nested entities (37.45%) compared with nested ones of GENIA (21.56%).

Table 4.6 shows the results of models on the development sets of GENIA and ACE2005, respectively. From this table, we can see that our model, which only utilises context representation for preparation of input for the next flat NER

Settings	GENIA			ACE2005		
	P (%)	**R (%)**	**F1 (%)**	**P (%)**	**R (%)**	**F1 (%)**
Finkel and Manning (2009)	75.4	65.9	70.3	-	-	-
Lu and Roth (2015)	72.5	65.2	68.7	66.3	59.2	62.5
Muis and Lu (2017)	75.4	66.8	70.8	69.1	58.1	63.1
Our model	**78.5**	**71.3**	**74.7**	**74.2**	**70.3**	**72.2**

Table 4.5: Comparisons of our model with the state-of-the-art models on nested NER.

Settings	GENIA			ACE2005		
	P (%)	**R (%)**	**F1 (%)**	**P (%)**	**R (%)**	**F1 (%)**
Layered-BiLSTM-CRF	78.27	75.97	77.10	75.37	69.41	72.27
Layered-BiLSTM-CRF w/o layered non-entities (LBCWLE)	76.55	77.01	76.78	72.90	65.54	69.02
Layered-BiLSTM-CRF w/o layered LSTM (LBCWLL)	75.76	74.60	75.18	69.94	61.94	65.70

Table 4.6: Performances of ablation tests on development sets.

layer, performs better than the other models. This demonstrates that introducing input of the current flat NER layer such as skipping either representation for any non-entity or words or all intermediate LSTM layers hurts performance. Compared with the layered-BiLSTM-CRF model, the drop of the performance in the LBCWLE model reflects the skip of representation for out-of-entity words leading to the decline in performance. This is because the representation of non-entity words did not incorporate the current context representation as we used the input rather than the output to represent them. By analogy, the LBCWLL model skips the representation for both entities and non-entity words, resulting in performance decrease. This is because, when skipping all intermediate LSTM layers, input of the first flat NER layer, i.e., word embeddings, is passed to the remaining flat NER layers. Since word embeddings do not contain contextual representation, we fail to incorporate the context representation when we use the word embeddings as the input for the flat NER layers.

Table 4.7 and Table 4.8 describe the performance for each entity type in GENIA and ACE2005 test datasets, respectively. In GENIA, our model performed best in recognizing entities with type *RNA*. This is because most of the entities pertaining to *RNA* mainly end up either with "mRNA" or "RNA". These two words are informative indicators of *RNA* entities. For entities in the other entity

Entity type	**P (%)**	**R (%)**	**F1 (%)**
DNA	74.43	69.68	71.98
RNA	90.29	79.48	84.54
Protein	80.48	73.20	76.67
Cell Line	77.83	65.65	71.22
Cell Type	76.36	68.07	71.97
Overall	78.59	71.33	74.79

Table 4.7: Results of all entities for each type in GENIA test set.

Entity type	**P (%)**	**R (%)**	**F1 (%)**
PER	78.82	77.37	78.09
LOC	54.54	43.47	48.38
ORG	63.25	54.20	58.38
GPE	76.92	78.98	77.94
VEH	61.53	48.48	54.23
WEA	66.66	53.73	59.50
FAC	49.19	35.26	41.07
Overall	74.27	70.34	72.25

Table 4.8: Results of all entities for each type in ACE2005 test set.

types, their performances are close to the overall performance. One possible reason is that there are many instances to model them. This also accounts for the high performances of entity types such as *PER*, *GPE* in ACE2005. The small amounts of instances of entity types like *FAC* in ACE2005 is one reason for their under overall performances.

When evaluating our model on top level which contains only outermost entities, the precision, recall and F1-score were 78.19%, 75.17% and 76.65% on GENIA test set. For ACE2005, the corresponding precision, recall and F1-score were 68.37%, 68.57% and 68.47%. Compared with the overall performance listed in Table 4.5, we obtained higher top level performance on GENIA but lower performance in ACE2005. We discuss the details of those results in the following tables.

Table 4.9 shows the performances of each flat NER layer in GENIA test dataset. Among all the stacking flat NER layers, our model resulted in the best performance regarding standard evaluation metrics on the first flat NER layer which contains the predictions for the gold innermost entities. When the model

Layer	P (%)	R (%)	F1 (%)	EP (%)	ER (%)	EF1 (%)	#Predicted Entities	#Gold Entities
Layer 1	72.86	69.82	71.31	78.46	71.06	74.57	4,783	4,991
Layer 2	56.88	27.59	37.15	81.15	73.98	77.39	276	569
Layer 3	0.00	0.00	0.00	0.00	60.00	0.00	1	15

Table 4.9: Results of layer evaluation on GENIA test set.

went to deeper flat NER layers, the performance dropped gradually as the number of gold entities decreased. However, the performance for predictions on each flat NER layer was different in terms of extended evaluation metrics. For the first two flat NER layers, the performance of extended evaluation is better than the performance of standard evaluation. It indicates that gold entities corresponding to some of the predictions on the specific flat NER layer are from other flat NER layers. This may lead to the zero performances for the last flat NER layer. In addition, performance on the second flat NER layer was higher than it was on the first flat NER layer in terms of extended F1-score. This demonstrates that our model is able to obtain higher performance on top level of entities than innermost entities.

Table 4.10 lists the results of each flat NER layer on ACE2005 test set. Similar to GENIA, the first flat NER layer achieved better performance than the rest flat NER layers. Performances decreased in a bottom-to-up manner regarding model architecture. This phenomenon was the same with the extended evaluation performances, which reflects that some of the predictions in a specific flat NER layer were detected in other flat NER layers. The extended F1-score dropped when the number of layers increased, which accounts for the fact that the F1-score on the top level was lower than that on the first flat NER layer. Unlike GENIA, our model on ACE2005 stopped stacking layers before reaching the maximum nested level of entities. It indicates that our model failed to model nested levels corresponding to the layers. This is one of the reasons that account for the zero predictions on the last flat NER layer. The sparse entities with high nesting levels could be another reason that resulted in the zero performances on the last flat NER layer.

4.1.2.5 Error Analysis

Based on the experimental results, we gained 3.9 and 9.1 percentage point improvements regarding F1-score over the state-of-the-art feature-based model on

Layer	**P** (%)	**R** (%)	**F1** (%)	**EP** (%)	**ER** (%)	**EF1** (%)	**#Predicted Entities**	**#Gold Entities**
Layer 1	74.46	73.39	73.92	75.84	73.77	74.79	2,894	2,936
Layer 2	60.28	50.49	54.95	66.19	58.41	62.05	423	505
Layer 3	51.02	24.51	33.11	51.02	37.25	43.06	49	102
Layer 4	0.00	0.00	0.00	0.00	10.00	0.00	0	10
Layer 5	0.00	0.00	0.00	0.00	0.00	0.00	0	1

Table 4.10: Results of layer evaluation on ACE2005 test set.

$$\text{Whether that is true now, } \underbrace{\underbrace{\text{we}}_{Annotation(ORG)}}_{Prediction(PER)} \text{ can not say.}$$

Figure 4.2: A sentence containing the annotations and predictions. "ORG" represents organization while "PER" means "person".

two nested entity corpora: GENIA and ACE2005, showing the effectiveness in utilising dependencies between nested entities for nested NER.

Furthermore, we conducted error analysis on test sets of GENIA and ACE2005, respectively. Specifically, we randomly selected 200 sentences from each test set and showed the error types coupled with the statistics in terms of entities and layers.

On ACE2005 test set, 28% of predictions were incorrect in the 200 sentences. Among these errors, 39% of them were because their text spans were assigned to other entity types. We call this type of errors *type error*. The main reason is that most of them are pronouns and co-refer with other entities which are absent in the sentence. Taking the sentence in Figure 4.2 as an example, "we" is annotated as *ORG* while our model labeled it as *PER*. Lack of context information such as the absence of co-referent entities leads our model to make the wrong decisions. In addition, 30% of the errors were because predictions were labeled as only parts of gold entities with correct entity types. This error type is referred to as *partial prediction error*. This might be because these gold entities tend to be considered as clauses or independent sentences, thus possibly containing many modifiers. For example, in Figure 4.3, our model extracted only parts of the annotation as the prediction.

Predictions from the 200 sentences only involve the first three flat NER layers. When analyzing errors on the first flat NER layer, we got 41% of type error and 11% of partial prediction error. Apart from this, our model recognized predictions

$$\underbrace{\underbrace{\text{A man who has been to Baghdad many times}}_{\textbf{Prediction (PER)}} \text{ and can tell ... Baghdad - Judy.}}_{\textbf{Annotation (PER)}}$$

Figure 4.3: A sentence containing the annotations and predictions. "PER" represents "person".

from other flat NER layers, leading to 5% errors. We define this error type as *layer error*. On the second flat NER layer, 26% of errors were caused by partial prediction error. 17% of the errors belong to type error. 22% errors were due to the *layer error*. As for the last flat NER layer, 40% errors were caused by *partial prediction error*. The remaining errors were different from the mentioned error types. One possible reason is that we have fewer gold entities to train the last flat NER layer compared with previous flat NER layers. Another reason might be the error propagation from its bottom layers.

On GENIA test set, we had 20% errors of predictions in the subset of 200 sentences. Among these errors, 17% and 24% of errors were separately due to *type error* and *partial prediction error*. Predictions in the subset were from the first two flat NER layers. In terms of layer errors, 24% of the predictions on the first flat NER layer were incorrect. Among them, the top error types were *layer error*, *partial prediction error* and *type error*, accounting for 21%, 18% and 13%, respectively. Errors on the second flat NER layer were mainly caused by *type error* and *partial prediction error*.

4.2 Task-specific Evaluation

In this section, we present task-specific evaluation with flat and nested NER settings, where our model was used as one of the steps in each task. In detail, we conducted NER in the neuroscience domain under flat NER setting to help curation of neuroscience entities. For nested NER setting, we evaluated the model by identifying pertinent and potentially complex information about chronic obstructive pulmonary disease phenotypes from textual data in clinical domain. In addition, we applied our model to extract adverse drug event and medication information from clinical records. We also adopted the model to extract elements of patient/population, intervention, comparator, and outcomes, which were further used in the scientific abstract screening task. We detail each task-specific

evaluation in the following sections.

4.2.1 Entity Extraction for Neuroscience

The curation of neuroscience entities is crucial to ongoing efforts in neuroinformatics and computational neuroscience, such as those being deployed in the context of continuing large-scale brain modelling projects. However, manually sifting through massive articles for new information about modelled entities is a time-consuming and tedious task. To aid the systematic extraction of relevant information from this literature, Shardlow et al. (2018) proposed a method that contains two steps. Specifically, two computational neuroscientists annotated a corpus of entities pertinent to neuroscience using active learning techniques (Settles, 2009) to enable swift, targeted annotation. Then ML models including CRF-based, neural-based and rule-based methods were developed to recognise the annotated entities. As one of the co-authors in Shardlow et al. (2018), we contributed to the second step by developing a neural-based model. To give a detailed overview, we present the following:

- Background of the task and its related work
- Results of our model under flat NER setting
- Discussion of the results

4.2.1.1 Background

Large projects such as the Swiss Blue Brain Project, the European Human Brain Project, the Allen Brain Observatory, and the American BRAIN initiative have recently emerged in neuroscience and are pushing traditional neuroscience toward the big science paradigm (Underwood, 2016). These projects make the most of big data to model in great detail the functioning of the brain, down to the level of the individual neuron types. The data-driven approach adopted for such large-scale modelling requires characterization of numerous entities, such as neuron types, synapses, and ion channels. Laboratory experiments are in general used for evaluating these entities in terms of structure and function. However, it is extremely challenging to comprehensively evaluate these entities due to the complexity and cross-scale of the modelled phenomena. Thus, experimental data must be complemented with the scientific knowledge accumulated world-wide

and recorded in the scientific literature. Such situation results in the urgent need for large and high-quality databases of literature-curated information about neuroscience entities.

To promote traceability and reusability of systematic curation for data-driven modelling of the brain, O'Reilly et al. (2017) presented a new manual framework that requires a group of curators to sift through abstracts and full texts to identify new entities and their properties. This is a time-consuming and painstaking process, which requires a curator to maintain domain knowledge alongside the informatics knowledge to be able to discover all the relevant documents for a new entity. Moreover, the meaning of terms may shift over time and new terms may be created to understand the field changes as new material is published. One effective approach to lift the burden of informatics from the curator is to employ text mining, which allows them to focus on applying their own domain knowledge to the entities in question. Müller et al. (2008) presented the Textpresso, a text mining framework for neuroscience, to help search through neuroscience research papers by providing a semantic search interface. Users can enter names of entities from predefined categories and are shown documents which contain their entity of choice. The system also handles relations, allowing a user to filter documents based on entities occurring in a number of relation types. One drawback of Textpresso is that it relies on dictionaries to match against the text of documents to identify named entities, thus failing to identify new entities or variants of entities.

To enable neuroscience NER, one of the common approaches is to employ CRF models (French et al., 2009a; French and Pavlidis, 2012) coupled with features including POS, lemma and other word surface features. French et al. (2009b) developed a large corpus of brain region mentions and built a custom CRF-based approach to identify their entities. They used context features which determined whether a given word was likely to occur before or after an annotation, as well as features encoding structural information about the words. As an enhancement, Richardet et al. (2015a) added features, which encode the presence of species and measurements, to the CRF-based model, leading to better performance. More recently, neural-based methods have greatly improved performance on a number of NLP tasks (Huang et al., 2015; Ma and Hovy, 2016; Lin et al., 2017a; Luo et al., 2017). Based on neural models, we proposed an approach that uses LSTM (Hochreiter and Schmidhuber, 1997) besides the CRF model (Shardlow

Entity Type	Statistics	Entity Type	Statistics
Brain Region	1,055	Neuron Type	767
Model Organism	299	Ion Channel	201
Ion Current	339	Ion Conductance	76
Value	594	Unit	507
All	3,838	–	–

Table 4.11: Statistics of entities in the corpus.

et al., 2018) for curating neuroscience entities. Our method helped the curator to quickly survey the literature for papers of interest through automatic entity extraction. In addition, when processing a paper to extract relevant experimental values, the curator benefited from these entities to speed-up the identification of characterisation of the context surrounding such experimental values (e.g., cell type, species, brain regions). In addition, our method covers more neuroscience entity types than other work which focused only on the identification of brain regions (French et al., 2009c, 2012), neuron types (Ambert et al., 2013), brain connectivity (Vasques et al., 2015; Richardet et al., 2015b), and entities related to spinal cord injuries (Stöckel et al., 2015).

4.2.1.2 Experimental Setting

In this task, the corpus used for NER is available at https://github.com/nactem/TM4NS. It contains eight entity types, whose statistics are shown in Table 4.11. We split the corpus into train, development and test chunks with a ratio of 75%, 15% and 15%, respectively. We trained the neural model for each entity type using the training set, tuned the hyper parameters on the development set and then tested on the test set. We used precision (P), recall (R) and F1-score (F1) as model evaluation metrics. In comparison with related work, we applied our model to the available corpus that contains entities "BrainRegion" only (French et al., 2009b) with both strict and lenient matching evaluation metrics.

We initialised word embeddings using the pre-trained embeddings from Chiu et al. (2016b). In addition, Adam (Kingma and Adam, 2015) was used for model optimisation. We fixed the dimensions of word and character embeddings with 200 and 50 respectively. In addition, the dropout rate (Hinton et al., 2012) was set to 0.5. For learning rate and weight decay, we tuned them using the grid/exhaustive searching method on the validation set. The batch size and epoch number were set as default values specified by Chainer (Tokui et al., 2015b). Table

Hyper Parameter	Value	Hyper Parameter	Value
Epoch	20	Character embedding dimension	25
Batch Size	10	Word embedding dimension	200
Dropout rate	0.5	Character-based word embedding	250
Learning Rate	0.013	Weight decay	0.0001

Table 4.12: The hyper parameters that were used during training of our model.

Method	Strict Matching			Lenient Matching		
	P (%)	R (%)	F1 (%)	P (%)	R (%)	F1 (%)
French et al. (2009b)	81.3	76.1	78.6	91.6	85.7	88.6
Richardet et al. (2015a)	84.6	78.8	81.6	88.4	81.0	84.6
Ours	82.1	81.5	81.8	93.4	92.7	93.1

Table 4.13: Our results compared to previously published NER tools for Brain Regions.

4.12 summarises the value of hyper parameters used in the training stage.

4.2.1.3 Results and Discussion

Table 4.13 shows the comparisons between our model and related work on the corpus provided by French et al. (2009b). Compared with related work, our neural model was able to achieve comparable performances in terms of strict matching and higher performances in terms of lenient matching.

Table 4.14 shows that the pattern-based method (Shardlow et al., 2018), which uses rules in combination with dictionaries, produced the lowest performances, which are mainly due to the following reasons. Firstly, dictionaries in pattern-based method did not contain sufficient cases to capture the wide scope of entities that were present in the corpus. Secondly, the wide variation of terms restricts the pattern-based method, which depends on limited dictionaries to cover the term variations. Compared with the pattern-based method, the CRF-based method performs much better, indicating that contextual information is important for neuroscience NER. Our neural model outperforms the CRF-based method in most entity types, achieving the highest F1-score for all entity types except *Experimental Value*. The CRF and neural models outperformed the pattern-based method, demonstrating that the CRF and neural models are both capable of learning features from the data that are helpful for extracting entities.

Our neural model performed better than the CRF-based method, indicating that the neural-based method was able to access more information about each

Entity Type	**Pattern-based Method (%)**			**CRF-based Method (%)**			**Ours (%)**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Brain Region	32.4	30.4	31.4	**88.0**	77.2	82.2	85.6	**83.3**	**84.4**
Neuron Type	22.5	33.6	26.9	86.4	67.3	75.7	**87.8**	**76.0**	**81.4**
Model Organism	59.9	36.5	43.5	**92.7**	77.6	84.4	86.0	**87.8**	**86.9**
Ion Channel	32.2	24.4	27.8	64.3	56.3	60.0	**73.7**	**87.5**	**80.0**
Ion Current	12.8	10.9	11.8	85.3	58.0	69.0	**87.2**	**68.0**	**76.4**
Ion Conductance	5.6	9.2	7.0	**100**	22.2	36.4	92.9	**72.2**	**81.3**
Value	26.4	32.0	28.9	**89.7**	**83.9**	**86.7**	89.5	82.8	86.0
Unit	25.6	54.4	34.8	**91.5**	94.2	92.9	90.4	**95.7**	**93.0**
Micro all	27.2	28.9	26.5	**87.2**	67.1	73.4	86.6	**81.7**	**83.7**

Table 4.14: The results of our methods to identify the entities in our corpus. The pattern-based and CRF methods are from Shardlow et al. (2018).

named entity, and use this information to make better decisions on which words corresponded to named entities. This is expected as the neural-based method uses word embeddings as its input, which encode deep contextual information about each word and are richer than the features passed to the CRF. Our model outperforms the CRF-based method in terms of recall for seven out of eight classes (all except for *Experimental Value*). However, for precision, the neural model only outperforms the CRF-based method in three out of eight cases (see Table 4.14). When these two statistics are combined to calculate the F1-score, our neural model outperforms the CRF-based method in all entity types except "Experimental Value".

To highlight the differences between neural and pattern-based methods, we have collected several examples from the data that highlight the differences between a pattern-based approach and our neural model. These sentences focus on the annotation of brain regions and are presented in Figure 4.4. In each case, the neural model was able to get the same results as our gold standard manual annotations (even though it has never seen these examples before), whereas the pattern-based method made some mistakes. In the first example, our model and the human annotator both extracted the text "rostral pole of the ventral posteromedial nucleus" as the brain region that was being mentioned. However, the pattern-based method missed the first part of the annotation, only getting "ventral posteromedial nucleus". Whilst this is usually a brain region, in this case it is not the brain region being described and would be misleading if it was accepted as correct. The pattern-based method was not able to detect this complex term,

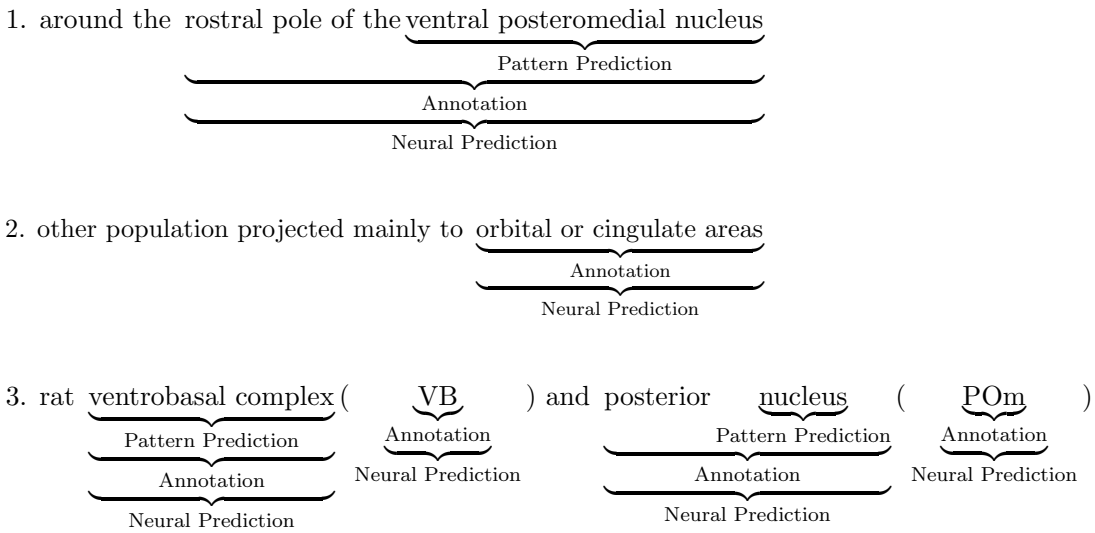

Figure 4.4: A comparison of rule-based recognition of brain regions to the neural model. The manual annotation of the texts, which we used to judge our methods performance against, is also included. The rule-based is from Shardlow et al. (2018).

instead deferring to a simpler term which was already in the dictionary. Our neural model has been able to use information about the context and structure of the sentence to correctly assign the whole part of the annotation as a brain region. In the second example, the human annotator and the neural model both picked out the phrase "orbital or cingulate areas" as a brain region of interest. The pattern-based method, however, did not find any brain region in this area. In this case, the specific phrase is not in the dictionary we used, and therefore was not picked up by the rule-based method. In the final example, the human annotator and the neural model have both picked out the following brain regions: "ventrobasal complex", "VB", "posterior nucleus" and "POm". The pattern-based method, however, has picked out the following brain regions: "ventrobasal complex" and "nucleus". Whilst the first brain region picked by the pattern-based method is correct, it has missed both the acronyms and has only found half of the second annotation (missing the word "posterior"). We were able to detect some acronyms using the pattern-based method (as shown in Example 1), however this did not have as high accuracy as for the neural model. For sentence 3, the acronyms were not part of the dictionary and could not be resolved to the text, therefore they were not annotated. The neural model has been able to learn something about the context of these acronyms that has allowed it to correctly identify them as brain regions.

4.2.2 Chronic Obstructive Pulmonary Disease Phenotype Extraction

In this section, we describe the task-specific evaluation of our neural model (i.e., layered BiLSTM-CRF) under nested NER setting in the clinical domain. The task was to identify pertinent and potentially complex information about chronic obstructive pulmonary disease (COPD) phenotypes from textual data. Specifically, it requires three steps: corpus annotation which contains COPD phenotypes, annotation normalisation and the nested NER to detect fine-grained COPD phenotypic information. Our contribution was the first step, which has been published at JAMIA Open (Ju et al., 2019b). To give a clear overview, we present the following:

- Background of the task and its related work
- Results of our neural model under nested NER setting
- Discussion of the results

4.2.2.1 Background

COPD is "a common, preventable and treatable disease that is characterized by persistent respiratory symptoms and airflow limitation that is due to airway and/or alveolar abnormalities usually caused by significant exposure to noxious particular gases" (cop). It is rapidly becoming one of the major causes of morbidity and mortality worldwide (Naghavi et al., 2017). COPD is a multifactorial and heterogeneous disease and not every patient responds to all available drugs (Miravitlles et al., 2013; Segreti et al., 2014; Cazzola et al., 2017). Due to the high prevalence and heterogeneity of COPD, improved deep phenotyping strategies are required. Such in-depth phenotyping can pave the way for personalised treatment regimens (Miravitlles et al., 2012), ensuring that the most suitable therapies are provided (Wouters et al., 2017; Heaney and McGarvey, 2017).

A phenotype can be broadly defined as "any observable characteristic of an organism" (Gkoutos et al., 2017), while Han et al. (2010) defines a COPD phenotype more specifically as "a single or combination of disease attributes that describe differences between individuals with COPD as they relate to meaningful outcomes (symptoms, exacerbations, response to therapy, rate of disease progression, or death)". Identifying such phenotypes (also described as phenotypic

traits) allows grouping of patients according to their prognostic and therapeutic characteristics (Han et al., 2010). Early classification of the COPD sub-type will facilitate superior healthcare provision and early intervention where it is most required for example, patients with rapid disease progression or frequent exacerbations. Although pinpointing relevant information in large, heterogeneous text repositories can be time-consuming, applying neural-based methods to semantically analyse these repositories (Zeng et al., 2018) can significantly reduce the time needed by clinicians and researchers for tasks such as finding relationships amongst concepts (e.g., genotype-phenotype (Van Driel et al., 2006; Singhal et al., 2016), gene-disease (Piñero et al., 2016; Thompson and Ananiadou, 2017; Bundschus et al., 2008), disease-phenotype (Kocbek and Groza, 2017; Sarntivijai et al., 2016)), diagnosis categorisation (Carroll et al., 2011) or recruiting patients for trials and studies (Wu et al., 2018; Ni et al., 2014)).

The potential complexity of COPD phenotype description, is exemplified in Figure 4.5, where the phrase "elevation of pulmonary arterial pressures" is identified as a phenotype, and is assigned the category *TestOrMeasureResult*, since it describes the outcome of a measurement. Analysing the internal structure of this phenotype reveals the specific measurement undertaken ("pulmonary arterial pressures") and anatomical entity involved ("pulmonary artery"). The annotations correspond to both complete phrases that constitute COPD phenotypes and other types of concepts frequently mentioned within them, and/or within their context. Such embedding (nesting) of shorter entity mentions within longer (outermost) phenotype descriptions is fairly common (29% of annotations in the COPD corpus are embedded). The detailed nature of the phenotype mentions in the corpus aims to facilitate the development of automated tools supporting the exploration of COPD phenotypic information in text from multiple perspectives. This will allow not only the location and categorisation of COPD phenotypes, including those identified through tests, or those constituting risk-raising individual behaviours (e.g., smoking), but will also permit detailed investigations about the nature of these phenotypes, including finding those affecting specific anatomical locations, or those concerning different results of specific tests. To demonstrate the full potential of the corpus for developing NER tools, our neural method is specifically employed to recognise nested entities. To the best of our knowledge, this is the first attempt to apply such an approach to detecting phenotypic information.

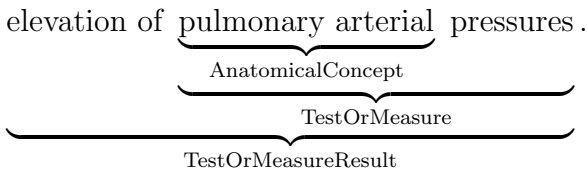

Figure 4.5: Example of a phenotype that includes other concepts nested within it.

4.2.2.2 Related Work

Previous approaches to phenotype NER depended on dictionaries (Friedman et al., 1994; Friedman and Hripcsak, 1998; Savova et al., 2017; Groza et al., 2015), possibly coupled with rules to improve accuracy and/or to handle the potentially complex structure of phenotype descriptions (Khordad et al., 2011; Afzal et al., 2018; Breitenstein et al., 2018; Mao et al., 2016; Collier et al., 2015a). Whilst some such approaches performed poorly on phenotype recognition (Oellrich et al., 2015), an optimised combination of the outputs of these methods can be beneficial (Collier et al., 2015b). However, combining or replacing rules with ML tends to achieve superior performance (Khordad et al., 2012; Collier et al., 2013; Brbić et al., 2017). Conventional ML approaches such as CRFs (Lafferty et al., 2001) have been applied to many NER tasks, including detecting CHF phenotypes (Alnazzawi et al., 2015) and recognising nested entities (Finkel and Manning, 2009; Lu and Roth, 2015; Muis and Lu, 2017). CRF-based methods generally require hand-crafted feature engineering for each new task, to determine the optimal set of textual features for predicting entities. Features include semantic information from domain-specific terminological resources or the output of linguistic processing tools, which can be time-consuming to apply to massive document collections.

Recently, however, representational methods have improved phenotype extraction performance (Gehrmann et al., 2018; Beaulieu-Jones et al., 2016; Che et al., 2015) by using word embeddings, which remove the need for hand-crafted feature engineering, linguistic processing or terminological resources (Collobert and Weston, 2008), and character embeddings, which encode word morphology information. However, they ignored nested phenotypic information, which was considered in our work (Ju et al., 2018). We describe our experimental setting in the next section.

4.2.2.3 Experimental Setting

Evaluation Setting

Our experiments evaluate performance variations of each model when entities with different levels of nesting are considered. We consider innermost entities, outermost entities and all entities in the test data set. Innermost entities are the most deeply nested entities, while outermost entities are non-nested entities. In Figure 4.5, elevation of pulmonary arterial pressures is the outermost entity, while pulmonary arterial is the innermost entity. Entities without nesting (e.g., dyspnea) are included in both the innermost and outermost sets. Precision (P), recall (R) and F1-score (F1) were used for the evaluation metrics in all models. The extended set of precision (EP), recall (ER) and F1-score (EF1) were additionally used for the evaluation of our nested model. We define that if the numbers of gold entities and predictions are all zeros, the evaluation metrics all equal one hundred percent.

Data Setting

The COPD corpus consists of 30 full-text scientific articles, which are annotated using the hierarchical scheme (Fu et al., 2015) (guidelines available at: http://www.nactem.ac.uk/COPD/download.php) to allow entities to be nested within each other. We randomly split the corpus into three different parts: four-fifths for training, one-tenth for development, and one-tenth for testing. Table 4.15 lists the statistics of the COPD corpus.

Model Setting

For the CRF and non-layered (i.e., flat) BiLSTM-CRF, we trained separate models to recognise only innermost and outermost entities. In contrast, our layered BiLSTM-CRF was trained to recognise entities at all levels of nesting.

Based on previous studies (Manda et al., 2018; Yang et al., 2018a), deciding on an optimal neural model, and whether to combine it with CRF, appears to be influenced by the task at hand. Using the layered architecture outlined above, we trained different deep learning models using different algorithms (BiRNN, BiGRU and BiLSTM), both in isolation and in combination with CRF.

We also compared our layered BiLSTM-CRF model to a CRF model and a flat BiLSTM-CRF model. We used NERSuite (Cho et al., 2010) to implement

Type	Description	Example	Statistics
Problem	An overall category for any COPD indicates of concern	COPD exacerbations, past pulmonary TB	2,556
Condition	Any disease or medical condition; includes COPD comorbidities	emphysema, pulmonary vascular disease, asthma	5,119
RiskFactor	A phrase signifying a patient's increased chances of having COPD	increased levels of the C-reactive protein, alpha1 antitrypsin deficiency	1,211
SignOrSymptom	An observable irregularity manifested by a COPD patient	chronic cough, shortness of breath	2,065
IndividualBehaviour	A patient's habits leading to susceptibility of having COPD	smoking for 25 years, exercise-limited patients	194
TestResult	Findings based on COPD-relevant examinations	decrease in rate of lung function, FEV1 45% predicted	685
Treatment	Any medication, therapy or treatment programme	inhaled corticosteroids, oxygen therapy, pulmonary rehabilitation	4,337
Test	An overall category for any COPD-relevant examinations or measures/parameters	spirometry, respiratory frequency, FEV1,	3,576
RadiologicalTest	Any radiological tests for detecting COPD	computed tomography scanning	29
MicrobiologicalTest	Examination of COPD- relevant specimen	complete blood count, bacterial isolates	11
PhysiologicalTest	A measurement of a COPD patient's capacity to exercise	6-min walking distance, incremental cardiopulmonary exercise testing	17
ConstituentConcept	An umbrella type for concepts that may form part of a phenotype description; should only be chosen if none of the subtypes below apply	bronchodilation, enhancement of skeletal muscle contractility	5
AnatomicalConcept	A mention pertaining to anatomical entities	lung, pulmonary,hepatic, respiratory airway	2,616
Drug	Any drug name	corticosteroids, short-acting bronchodilators,	2,593
Protein	Any protein name	alpha1 antitrypsin	820
Quality	Expressions that modify/qualify the concepts	chronic, obstructed,damaged, decreased rate	1,153

Table 4.15: Descriptions, examples and counts of each category in the COPD corpus.

Hyper Parameters	Layered BiLSTM-CRF	Layered BiRNN-CRF	Layered BiGRU-CRF	Layered BiLSTM	Layered BiRNN	Layered BiGRU
Batch Size	121	92	65	94	92	114
Learning Rate	0.008696	0.004173	0.004009	0.010222	0.004173	0.002182
Weight Decay	0.000293	0.00592	0.000861	0.000010	0.000592	0.000075
Dropout Rate	0.430745	0.095110	0.219963	0.330216	0.095110	0.359987
Gradient Clipping	29	28	49	16	28	29

Table 4.16: Hyperparameters used in neural models which are tuned on the development set.

Parameters	Initialisation	Parameters	Initialisation
minimizer	gp_minimizer	nCalls	10
randomstate	-1	acqFunc	gp_hedge
noise	-1	nRandomStarts	10
nRestartOpt	100	acqOpt	lbfgs
nPoints	50000	xi	0.01

Table 4.17: Parameters used in initialising Bayesian optimisation.

the model, whose features include contextual information such as n-grams (i.e., up to three words either side of the entity), parts-of-speech, syntactic chunks and word base forms (Ni et al., 2014). In contrast, the flat BiLSTM-CRF used only word and character-level embeddings instead of features, as described above. All neural-based models were tuned on the development data set using Bayesian optimization (Snoek et al., 2012). The hyper parameter values and the settings of the Bayesian optimization are shown in Table 4.16 and 4.17.

4.2.2.4 Results and Discussion

Table 4.18 shows the performance of each model. The flat BiLSTM-CRF performs best for innermost entities, demonstrating how embeddings can successfully replace the multiple linguistic features used by the CRF. At this level, however, the layered BiLSTM-CRF has lower performance than the flat BiLSTM-CRF. For the layered model, we consider only the output of its first layer, which is expected to recognise only innermost entities. However, error analysis revealed that there is actually not a one-to-one correspondence between model layers and entity nesting levels, i.e., the first layer sometimes detects entities belonging to

other (i.e., not innermost) entity levels. Conversely, higher layers of the model may detect entities that belong to the innermost nesting level. For outermost entities, the flat BiLSTM-CRF still outperforms the CRF, reinforcing the advantages of deep learning. However, in contrast to innermost entities, the layered BiLSTM-CRF outperforms the flat model in detecting outermost entities. This clearly demonstrates how the layered models use of information about lower-level entities improves recognition of higher-level entities. The higher performance of the layered BiLSTM-CRF for outermost entities also provides evidence that innermost entities are successfully recognised by lower levels of the model. This is confirmed by its superior performance to the other models in detecting all entities in the test data set. Although there is no exact correspondence between the recognition of specific levels of entities and layers of the model, the complete model is still able to exploit the output of previous layers to achieve a high level of performance in detecting both outermost and nested entities. Detailed performance statistics for the layered BiLSTM-CRF by entity type are provided in Table 4.19. In Table 4.19, the performance of *ConstituentConcept* phenotypes is 100%. This is because there are no entities with this semantic type in the test data set and our model also did not predict any entities with this semantic type. In contrast, we obtained 0% performance for *MicrobiologicalTest* and *PhysiologicalTest* semantic types due to their sparsity in the testing data set.

Level	**Model**	**P (%)**	**R (%)**	**F1 (%)**
Innermost	CRF	**77.19**	68.78	72.74*
	Flat BiLSTM-CRF	73.93	**73.38**	**73.56***
	Layered BiLSTM-CRF	69.79	70.41	70.10
Outermost	CRF	73.63	66.41	69.83*
	Flat BiLSTM-CRF	**75.61**	67.35	71.24*
	Layered BiLSTM-CRF	74.00	**74.54**	**74.27**
All	CRF	75.44	67.61	71.31*
	Flat BiLSTM-CRF	74.71	70.42	72.50*
	Layered BiLSTM-CRF	**77.02**	**75.45**	**76.23**

Table 4.18: Performance of different NER models at different levels of entity nesting. For each different level, the best precision (P), recall (R) and F1-score (F1) amongst the three models is shown in bold.

Table 4.20 illustrates the performances of neural-based models, trained on the training portion of the COPD corpus, tuned using the development set and evaluated on the test set. In each case, the models are layered, and are evaluated

Entity Type	P (%)	R (%)	F1 (%)	# Entities
Problem	70.54	69.00	69.76	229
Condition	86.75	86.57	86.66	484
RiskFactor	70.77	56.79	63.01	81
SignOrSymptom	60.38	53.63	56.81	179
IndividualBehaviour	63.64	77.78	70.00	9
TestResult	73.33	28.21	40.74	78
Treatment	81.66	81.34	81.50	509
Test	67.82	68.01	67.91	347
RadiologicalTest	50.00	25.00	33.33	4
MicrobiologicalTest	0	0	0	1
PhysiologicalTest	0	0	0	4
ConstituentConcept	100	100	100	0
AnatomicalConcept	74.09	84.33	78.88	217
Drug	85.15	87.36	86.24	348
Protein	63.89	63.01	63.45	73
Quality	76.87	76.87	76.87	134

Table 4.19: Performance of layered model on each semantic type.

Model	P (%)	R (%)	F1 (%)
Layered BiLSTM - CRF	77.02	75.45	76.23
Layered BiRNN - CRF	75.72	67.52	71.38
Layered BiGRU - CRF	76.99	72.71	74.79
Layered BiLSTM	73.69	66.37	69.84
Layered BiRNN	64.58	53.62	58.59
Layered BiGRU	72.77	66.59	69.55

Table 4.20: Performance of different layered deep learning based models applied to the test set of the COPD corpus.

on all entities (both nested and flat) in the test set of the corpus. The bi-directional versions of three neural models are evaluated, i.e., BiRNN, BiGRU and BiLSTM. In each case, two different versions of the models were evaluated (i.e., alone and in combination with CRF). In all cases, the addition of CRF helped to boost performance, and the layered BiLSTM-CRF was the highest performing model.

The results achieved by our layered BiLSTM-CRF in recognising COPD-related information are superior to those achieved by applying the same model to nested entity recognition in well-used corpora from other domains (Ju et al., 2018). This provides evidence that the COPD corpus is suitable for training

high-performance ML models, and that automatic recognition of COPD phenotypic information is a feasible task. Moreover, we have shown that detecting COPD phenotype information using neural models, which require minimal human intervention for training, can achieve superior performance to more traditional methods requiring time-consuming feature engineering, linguistic processing and terminological resources. We have furthermore demonstrated that our layered model can achieve superior performance against a "flat" model, by exploiting information about nested entities. These outcomes have important implications, in terms of improving the extraction of phenotypic information in text. In particular, our nested entity detection method not only allows efficient location of COPD phenotype descriptions hidden in large text collections, but it also detects the internal structure of these descriptions. This provides scope to explore and categorise COPD phenotypes in a fine-grained manner. Since our method can be rapidly adapted to detect different types of information, it could be readily applied to find phenotypic information relating to other diseases, given suitably annotated corpora. Error analysis of our NER results reveals that about 17% of erroneous entities have the correct text span, but the wrong semantic category. Figure 4.6 provides detailed error statistics for each semantic type, revealing that Problem is the most frequently misclassified category; these entities are mainly misclassified as either *Condition* or *SignOrSymptom*. Conversely, *MedicalCondition* entities are mostly misclassified as *Problem*. Such errors are possibly due to the fine-grained, hierarchical structure of the annotation scheme; the often subtle differences between similar categories may be difficult for the computer to distinguish. A further 23% of errors (most frequently *Treatment* and *TestOrMeasure* entities) concern cases where the model assigns the correct category, but the wrong text span (i.e., it partially overlaps with the correct span). This may be due to the heterogeneity of phenotype descriptions, which can include mentions of various concept types, and which may or may not include modifier phrases. However, it is significant that in around 40% of the erroneous cases, the model can successfully detect the presence of entities, and categorise them correctly. Thus, even if the span is not completely correct, the model can find documents mentioning relevant entities, and allow examination of the context surrounding these entities.

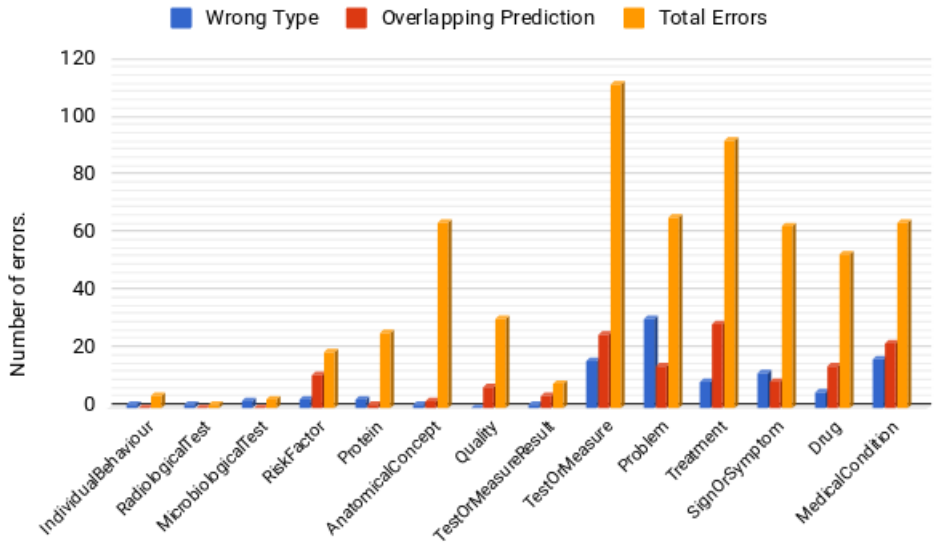

Figure 4.6: Counts of different types of errors for each semantic type.

4.2.3 Adverse Drug Event and Medication Extraction

In this section, we describe the task-specific evaluation of our neural model under nested NER setting in the clinical domain. The task is to identify drugs and their attributes (i.e., drug-related entities), which is one of the subtasks in 2018 n2c2 Shared Task Track 2. In this task, we used our neural model to extract both nested and polysemous entities (i.e., entities that have multiple semantic types) without depending on any external knowledge resources and hand-crafted features. To improve the extraction of sparse entities, we further incorporated subwords using byte pair encoding (Sennrich et al., 2016). To take advantage of feature-based models, a CRF model is further combined with our neural model enriched with subword information. Work based on this task has been published at JAMIA (Ju et al., 2019a), where my contribution was the development of neural-related models which additionally incorporates subword units. To give a clear overview, we present the following:

- Background of the task and its related work
- Results of our neural model with subword units under nested NER setting
- Discussion of the results

4.2.3.1 Background

Electronic health records (EHRs)–a digital version of a patients information and medical history– are an important source of health data that can impact on a patients care. Mining such data is crucial in understanding of treatment and diagnosis of disease (Jensen et al., 2017; Yadav et al., 2018). Among the many known application areas of electronic health record (EHR) mining (Yadav et al., 2018; Velupillai et al., 2018), adverse drug event (ADE) detection has been proven to improve and complement drug safety surveillance strategies. According to the World Health Organization, an ADE is "An injury resulting from medical intervention related to a drug" (Organization et al., 1972). Our work focuses on extracting ADE mentions and their related medications from EHRs. We base our analysis on data sets provided by the n2c2 Shared Task Track 2, consisting of discharge summaries drawn from the Medical Information Mart for Intensive Care III (MIMIC III) clinical care database (Johnson et al., 2016). This task involves identification of nine entity types, i.e.,: *ADE*, *Dosage*, *Duration*, *Drug*, *Form*, *Frequency*, *Reason*, *Route* and *Strength*.

Approaches to ADE detection in EHRs are roughly split into rule-based, ML-based and neural categories. Iqbal et al. (2017) detected adverse drug events (ADEs) based on a predefined dictionary and post-processing rules. Similarly to Iqbal et al. (2017), Yeleswarapu et al. (2014) detected drugs and ADEs from multiple data sources using dictionaries compiled from MeSH[1] and MedDRA[2], respectively. Wang et al. (2018b) proposed a framework to extract vaccine ADEs by combining formal ADE reports (Vaccine Adverse Event Reporting System (VAERS)) with ADEs in social media (Twitter) and applying multi-instance learning methods. Nikfarjam et al. (2015) also extracted adverse drug reactions from social media by utilising word embedding cluster features, while Korkontzelos et al. (2016) used sentiment analysis features.

The TAC 2017 Adverse Reaction Extraction from Drug Labels Track (Roberts et al., 2017) is a similar shared task to the n2c2 Shared Task, but it focuses instead on drug labels. One of the tasks in TAC 2017 was to recognise six ADE types: adverse reaction, drug class, severity, factor, animal and negation. The most common approach was the use of BiLSTM-CRFs (Belousov et al., 2017; Cocos and Masino, 2017; Dandala et al., 2017; Gu et al., 2017; Tiftikci et al.,

[1]https://www.nlm.nih.gov/mesh/meshhome.html
[2]https://www.meddra.org/

He was given albuterol nebs . . . for presumed $\underbrace{\underbrace{\underbrace{\text{narcotic}}_{\text{Drug}} \text{ induced respiratory distress}}_{\text{ADE}}}_{\text{Reason}}$.

Figure 4.7: An example of a sentence containing nested entity annotations.

2017; Xu et al., 2017a). These systems were implemented with pre-calculated word embeddings and dynamically learned character embeddings. The MADE1.0 NLP challenge[3] was another similar shared task, involving detection of mentions of medication names and their attributes (dosage, frequency, route, duration), as well as mentions of ADEs, indications, and other signs and symptoms in EHRs of cancer patients. Neural-based models, e.g., LSTM (Xu et al., 2018a), BiLSTM (Florez et al., 2018) and BiLSTM-CRF (Yang et al., 2018b; Wunnava et al., 2018; Li et al., 2018a), were the most popular approaches for ADE detection.

4.2.3.2 Experimental Setting

Data Setting

The 2018 n2c2 Shared Task Track 2 provided 505 annotated discharge summaries extracted from MIMIC III, of which 303 were used for training and 202 were used for testing. The statistics are shown in Table 4.21. To determine the best ensemble setting, we further divided the training set into two subsets: 80% for training and 20% for development; the latter is used to validate the models. We evaluated all models using lenient metrics in terms of precision, recall and F1-score, which were the main ones used in Track 2. As an example of nested entities, consider Figure 4.7, where the *Drug* entity is embedded (nested) inside *Reason* and *ADE*. In addition, *ADE* and *Reason* are polysemous entities, since they both cover the same text span; we treat these as a special case of nested entities. As a result, the number of flat NER layers depends on the degree of nestedness of entities contained in the input word sequences. The dynamic nature of our model enables us to extract polysemous entities by stacking flat NER layers to recognise other categories with the same text span.

Table 4.21 provides statistics regarding the training data. We observe that there are many rare and unknown words (words that are unseen in the training data) included in entities, which makes their extraction challenging. To address

[3]http://bio-nlp.org/index.php/announcements/39-nlp-challenges

Item	**Training**	**Development**
Document	242	61
Entities	41,171	9,776
Nest level 1 entity (flat entities)	41,109	9,760
Nest level 2 entity	61	16
Nest level 3 entity	1	0
Polysemous entity	47	13
Textually nested entity	15	3
ADE	785	174
Dosage	3,401	820
Drug	13,109	3,114
Duration	499	93
Form	5,340	1,311
Frequency	5,075	1,205
Reason	3,105	750
Route	4,479	996
Strength	5,378	1,313
Unknown words /Unique words	-	17.00%
Rare words /Unique words	37.19%	37.69%
EUNKs/All entities	-	2.67%
ERAREs /All entities	1.89%	3.88%

Table 4.21: Statistics of the data set. Rare words are words that occur only once in the data. Unknown words refer to words that are not seen in the training set. EUNKs and ERAREs refer to entities that contain unknown and rare words, respectively.

this problem, we used byte pair encoding (Sennrich et al., 2016) that represents words by iteratively merging the most frequent adjacent/consecutive characters into longer character sequences (i.e., subwords). We collected all the words occurring in the training data and iteratively combined the most frequent pairs of neighbouring characters or character sequences, resulting in a tokenisation model in which each line contains one subword coupled with its unique id. The tokenisation model was used to split word sequences into subword sequences that may carry patterns of informative words in entities. We then concatenated the subword embeddings with word embeddings, which were used as input to our model.

Model Setting

Besides our layered BiLSTM-CRF model, we also incorporated a feature-driven CRF model Ju et al. (2019a) with token-based features, dictionary features and cluster features. With CRF and layered-BilSTM-CRF models, we created two types of ensemble using majority voting (Boyer and Moore, 1991): (1) intra-ensemble that combines different versions of the same model with different parameter settings, and (2) inter-ensemble that combines different models or different intra-ensembles. We refer to the models with intra- and inter-ensemble settings as intra-model and inter-model, respectively.

Regarding the CRF model Ju et al. (2019a), since lexical and syntactic features are default input features of NERSuite (Cho et al., 2010), we treated them as baseline features and evaluated the combinations of the remaining features, i.e., word shape, dictionary, and cluster features.

For our layered neural model, we experimented with the following settings:
(1) **baseline model**: using word embeddings concatenated with character embeddings as the input to the neural layered model. We randomly initialised a vector for each character. Given a word, we feed its character sequence to a BiLSTM and concatenated the bidirectional last hidden states as the character embeddings.
(2) **csub model**: using subword embeddings and character embeddings as the input to the model. Similarly to character embeddings, we used a different BiLSTM to obtain subword embeddings. We used varying vocabulary sizes of [300; 1,000; 4,000; 8,000; 16,000] to train different tokenisation models. As a result, we generated 5 different versions of subword sequences for a given input word

Vocabulary size	Subword sequence
300	_v, in, c, r, ist, ine, _to, x, ic, _p, o, ly, ne, u, ro, p, at, h, y
1000	_v, in, c, r, ist, ine, _to, x, ic, _po, ly, ne, u, rop, at, hy
4000	_v, in, c, r, ist, ine, _toxic, _poly, ne, u, rop, athy
8000	_v, inc, rist, ine, _toxic, _poly, ne, uropathy
16000	_vincristine, _toxic, _polyneuropathy

Table 4.22: Five versions of subword sequences for the given ADE entity "vincristine toxic polyneuropathy" that contains a Drug entity "Vincristine" inside itself. "_" represents the whitespace.

sequence. An example is shown in Table 4.22. Each different version of subword sequences produced subword embeddings, which were individually used in the model. Instead of predicting label sequences at the word level, we predicted the label for each word at the subword level. When merging the subword labels into their corresponding word labels, we kept the first subword label as their word label. Taking the entity *"vincristine toxic polyneuropathy"* as an example, we selected one version of the tokenisation model to generate its subword sequence *" _v, in, c, r, ist, ine, _toxic, _poly, ne, u, rop, athy"* where "_" represents a whitespace character. Using the csub model, the predicted subword-level label sequence is *[B-ADE, I-ADE, I-ADE, I-ADE, I-ADE, I-ADE, I-ADE, I-ADE, I-ADE, I-ADE, I-ADE, I-ADE, I-ADE]*, while the corresponding word-level label sequence is *[B-ADE, I-ADE, I-ADE]*. When merging subword-level labels for each word, we picked up the first subword label (e.g., *"B-ADE"*) among subword labels and attached it to the word *"vincristine"* as the final word-level label.
(3) **wsub model**: using the concatenation of word embeddings and each version of the subword sequences obtained from (2) as the input to the model.
(4) **wcsub model**: using the concatenation of baseline embeddings (i.e., word and character embeddings) and each version of the subword sequences obtained from (2) as the input to the model. By combining these settings, we have 16 models in total. We produced different combinations of intra- and inter-models using the majority voting (Boyer and Moore, 1991). Specifically, we merged predictions from: (1) different feature combinations of the CRF models; (2) combinations of NN baseline and the remaining NN models, which were internally ensembled with vocabulary sizes and (3) all of the above-mentioned settings. We selected entities that have the most votes for their specific span. We summarise the model ensemble in Figure 4.8. The NN models were tuned using Bayesian optimization (Snoek et al., 2012). The best hyper parameter values are listed in Table 4.23.

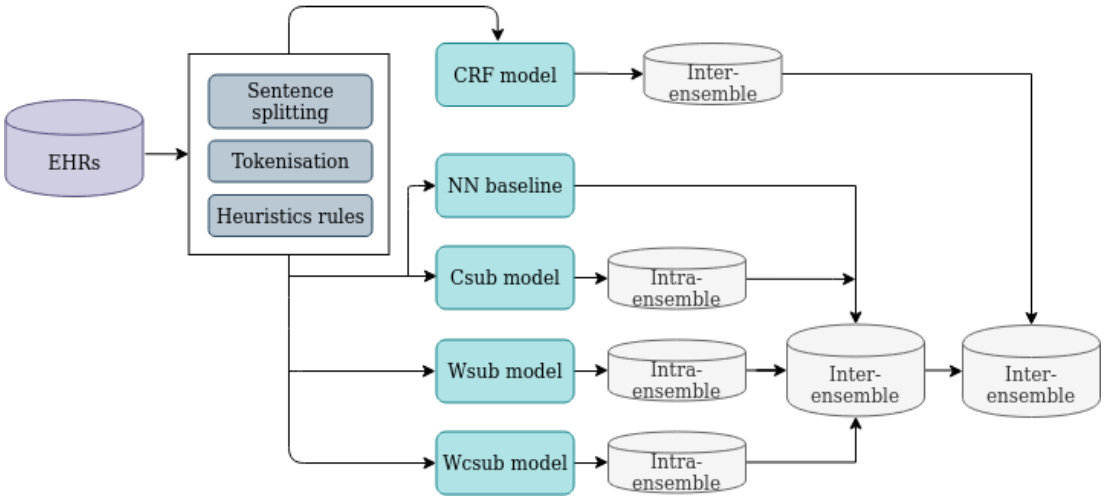

Figure 4.8: An overview of the ensemble.

Hyper Parameter	**Value Range**	**Best Value**			
		Baseline	**Csub**	**Wsub**	**Wcsub**
Batch size	[16, 256]	144	255	224	240
Learning rate	[0.001, 0.02]	0.007240	0.008162	0.008162	0.004740
Weight decay	[10^{-8}, 0.001]	2.73	1.08	8.28	1.32
Dropout	[0.1, 0.9]	0.419363	0.429264	0.475458	0.513470
Gradient clipping	[5, 50]	11	28	11	47
Vocab size	[0.3k, 1k, 4k, 8k, 16k]	-	4k	4k	0.3k

Table 4.23: Best hyper parameters of individual NN models.

Model	Vocab size	Ensemble					
		Strict			Lenient		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Baseline	-	0.8935	0.8675	0.8803	0.9484	0.9038	0.9256
Csub	300	0.8892	0.8649	0.8768	0.9472	0.9032	0.9247
	1000	0.8895	0.8769	0.8832	0.9439	0.9144	0.9289
	4000	0.8890	0.8831	0.8860	0.9454	0.9185	**0.9317**
	8000	0.8953	0.8762	0.8856	0.9493	0.9109	0.9297
	16000	0.8824	0.8673	0.8748	0.9406	0.9061	0.9230
	Ensemble	0.9218	0.8610	0.8904	0.9656	0.8981	**0.9306**
Wcub	300	0.8906	0.8710	0.8807	0.9486	0.9095	0.9286
	1000	0.8912	0.8750	0.8830	0.9463	0.9124	0.9291
	4000	0.8895	0.8776	0.8835	0.9422	0.9146	0.9282
	8000	0.8891	0.8815	0.8853	0.9458	0.9156	0.9305
	16000	0.8758	0.8782	0.8770	0.9385	0.9146	0.9264
	Ensemble	0.9198	0.8634	0.8907	0.9638	0.9013	**0.9315**
Wcsub	300	0.8909	0.8783	0.8846	0.9449	0.9153	0.9299
	1000	0.8864	0.8784	0.8824	0.9461	0.9125	0.9290
	4000	0.8897	0.8810	0.8853	0.9453	0.9183	**0.9316**
	8000	0.8889	0.8744	0.8816	0.9467	0.9115	0.9288
	16000	0.8892	0.8754	0.8823	0.9439	0.9121	0.9277
	Ensemble	0.9210	0.8637	0.8915	0.9641	0.9010	**0.9315**
Inter-NN	Ensemble	0.9105	0.8720	0.8909	0.9591	0.9084	**0.9331**
NN-CRF	Ensemble	0.8884	0.8838	0.8861	0.9423	0.9162	0.9291

Table 4.24: Performances of individual NN models and intra- and inter- ensembling models on the development set.

4.2.3.3 Results

The following experimental results were calculated using our development set and the official test set. Table 4.24 lists both strict and lenient matching detailed performances of our models on the development set. Table 4.25 summarises lenient performance of the CRF and NN models, including their combinations, on the development set.

As shown in Table 4.25, using word shape (ws) or dictionary features (df) alone reduced the CRF performance, while their combination produced the highest lenient precision. Our CRF model achieved the best lenient F1-score when using only cluster feature (cf) and the highest recall when further combined with df. Compared with the CRF models, our NN models obtained consistent improvements in terms of lenient recall and F1-score. We obtained the best performance with the wcsub model, which employs the embeddings of words, subwords and characters. The removal of word embeddings yielded the best precision without

Model	Precision	Recall	F1-score
CRF			
Baseline (Lexical and syntactic features)	0.9525	0.8825	0.9162
Baseline + word shape (ws)	0.9527	0.8815	0.9157
Baseline + dictionary features (df)*	0.9511	0.8829	0.9157
Baseline + cluster features (cf)*	0.9504	0.8902	0.9193
Baseline + ws + df	0.9523	0.8821	0.9158
Baseline + ws + cf	0.9491	0.8898	0.9185
Baseline + df + cf	0.9494	0.8903	0.9189
Baseline + ws + df + cf	0.9486	0.8900	0.9184
Neural Network			
Baseline (word + characters)	0.9476	0.8995	0.9230
Csub (subword + characters)	0.9502	0.9042	0.9266
Wsub (word + subword)	0.9496	0.9044	0.9264
Wcsub (word + subword + characters)*	0.9498	0.9066	0.9277
Ensemble			
Inter-CRF	0.9466	0.8935	0.9193
Intra-csub	**0.9656**	0.8981	0.9306
Intra-wsub	0.9638	0.9013	0.9315
Intra-wcsub	0.9641	0.9010	0.9315
Inter-NN	0.9591	0.9084	**0.9331**
NN-CRF	0.9401	**0.9209**	0.9304

Table 4.25: Performance of CRF and NN models on the development set. For each model, the best lenient metrics of precision, recall and F1-score are shown in bold. * represents significance value at $p < 0.05$ with approximate randomisation significance test (Noreen, 1989).

significantly sacrificing recall, thus achieving comparable lenient F1-score. The introduction of subwords to each individual character or word embedding produced better performance than their combination (i.e., the NN baseline model). Ensemble of models outperformed their individual ones except the inter-CRF. We obtained the best lenient F1-score when externally combining each intra-NN model, while the best recall was produced using an ensemble of inter-CRF and inter-NN models (i.e., NN-CRF).

Table 4.26 shows the performance of two ensemble settings on the test set. The first setting was our submission setting, which was an ensemble of inter-CRF, intra-csub, intra-wsub and wcsub models (initialised only with vocabulary sizes of 1,000 and 4,000). Using this setting, our ensemble model performed well in predicting entity types such as *Strength* and *Frequency*. The second setting was

Entity Type	Precision	Recall	F1-score
Submission Setting			
Strength	0.9815	0.9804	0.9810
Frequency	0.9788	0.9666	0.9727
Route	0.9662	0.9445	0.9552
Drug	0.9567	0.9533	0.9550
Form	0.9653	0.9436	0.9543
Dosage	0.9356	0.9433	0.9395
Duration	0.8875	0.7513	0.8138
Reason	0.7254	0.5470	0.6237
ADE	0.4697	0.1984	0.2790
Overall (micro)	0.9444	0.9073	0.9255
Inter-NN Setting			
Overall (micro)	0.9599	0.8979	0.9278

Table 4.26: Lenient performance on the test set with submission and inter-NN settings.

the inter-NN model setting, which produced the best lenient F1-score. However, it was not selected for submission to the shared task due to time limitations. In addition to lenient-based results, we show the strict performances in Table 4.27.

4.2.3.4 Discussion

We conducted an error analysis of predictions on the development set for the best individual and ensemble models. We divided errors into two classes: (1) category error (CE), corresponding to entities that have correct lenient spans but incorrect categories, and (2) span error (SE), corresponding to entities that have both incorrect spans and categories.

Figure 4.9 shows the statistics of CEs and SEs for our best individual and ensemble model on the development set. In general, our wcsub model made more CEs than the CRF model, indicating that the wcsub model detected more entity spans with incorrect categorisation. One reason is that the context representations enhanced with subwords from our wcsub model encode more informative than hand-crafted features used in the CRF model, thus providing more clues to locate more entity regions. Another reason is that our CRF model only handles flat entities and fails to consider nested entities, which are additionally utilised to train the wcsub model. When combining the predictions from all NN models, the number of errors (i.e., CEs and SEs) reduced, demonstrating the important

Entity Type	Precision	Recall	F1-score
Drug	0.9137	0.9342	0.9238
Strength	0.9424	0.9629	0.9525
Duration	0.7568	0.6667	0.7089
Route	0.9479	0.9331	0.9405
Form	0.9296	0.9147	0.9221
Ade	0.4491	0.1904	0.2674
Dosage	0.8984	0.9168	0.9075
Reason	0.6494	0.5014	0.5659
Frequency	0.8233	0.8445	0.8338
Overall (micro)	0.8890	0.8722	0.8805
Overall (macro)	0.8854	0.8599	0.8712

Table 4.27: The performances of our submission in terms of strict precision, recall and F1-score on the test set.

contribution of ensemble predictions.

Figure 4.10 shows the percentages of EUNKs (entities that contain unknown words) (a) and ERAREs (entities that contain rare words) (b) extracted for each category, respectively. As shown in Figure 4.10.a, our wcsub model is better able to extract EUNKs than the NN baseline model. This result demonstrates that, for most categories, the incorporation of subwords help the wcsub model to recognise such entities more accurately. Among all categories, our wcsub model achieved the highest improvement for *Strength* entities. Entities in this category commonly include words that exhibit a specific pattern within them (i.e., "digits symbol digits", e.g., *"150(2"*), indicating that subword features can help capture such internal features of words for entity recognition. For *Frequency* entities, the wcsub model misclassified three instances as *Duration*, since they are coupled with time units: *"pm"*, *"am"* and *"hs"*. Both of our models exhibit comparable performance for the *Duration* and *Form* categories, whose entities are often composed of informative words, such as *"day"*, *"month"*, *"tablet"* etc. In terms of *Reason*, the wcsub model extracted two fewer entities than the baseline, which correspond to long phrases, such as *"patchy infiltrates concerning for biliary sepsis"*. However, it was able to additionally extract a number of shorter entities (e.g., *"maculopapular rash"*), which were missed in predictions from the NN baseline model. In contrast to *Reason*, the wcsub model increases the recall of *Drug* entities, which constitute the largest proportion of all entities.

We additionally analysed how subwords can improve the extraction of entities

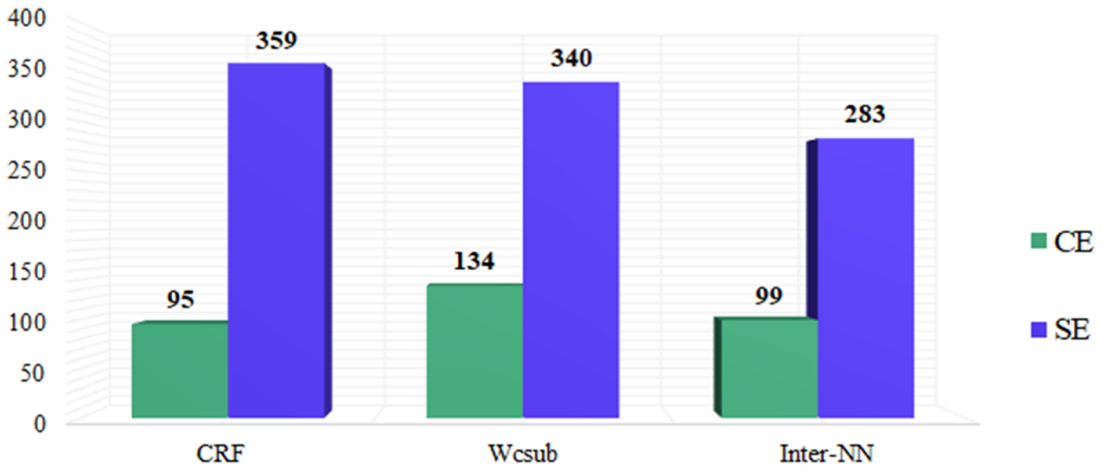

Figure 4.9: Statistics of CEs and SEs for our best individual and ensemble models on the development set.

containing rare words; the results are shown in Figure 4.10. It can be seen that the wcsub model improves the recall of sparse entities, especially for those belonging to *Strength*, *Dosage* and *Route*. This phenomenon indicates that entities with certain patterns (e.g., real values followed by units) benefit significantly from the use of subwords. In contrast, however, our wcsub model fails to capture sparse entities belonging to the categories of *Reason*, *Form* and *Frequency*. This is likely to be because they require contextual information beyond the sentence level.

4.2.4 Improving Reference Prioritisation with PICO Recognition

In this section, we evaluate our neural model under nested NER setting with a scientific abstract screening task in combination of biomedical and health domains.

Screening abstracts for systematic reviews requires users to read and evaluate abstracts to determine if the study characteristics match the inclusion criterion. A significant portion of these are described by PICO elements: patient/population (P), intervention (I), comparator (C), and outcomes (O). Therefore, words within PICO tagged segments automatically identified in abstracts are shown to be predictive features for determining inclusion. Combining PICO annotation model into the relevancy classification pipeline is promising to expedite the screening process. The task is divided into two steps: PICO recognition and relevancy

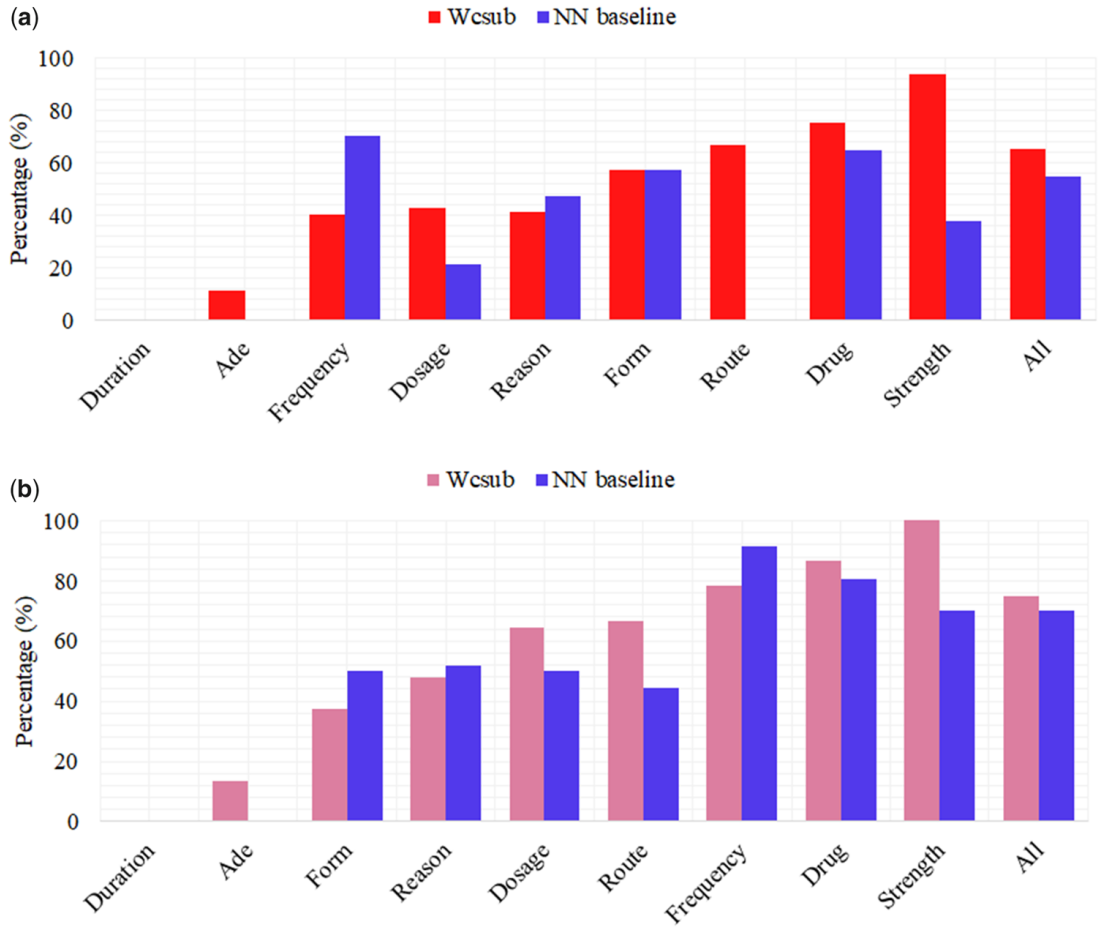

Figure 4.10: Percentage of category-wise extracted EUNKs (a) and ERAREs (b).

classification. Work based on this task has been under review at BMC Medical Informatics and Decision Making, where my contribution was the first step. To demonstrate the effectiveness of our model in the task, we briefly describe the second step: relevancy classification. Therefore, we present the following:

- Background of the task and its related work
- Explanation of relevancy classifier
- Results of our neural model under nested NER setting
- Discussion of the results

4.2.4.1 Background

Evidence-based research aims to answer a well-posed, falsifiable question using existing results together with a systematic and transparent methodology. Evidences (e.g., results of clinical trials) should be collected and evaluated without bias using consistent criteria for inclusion (Higgins and Deeks, 2011). Based on Huang et al. (2006), a research question can be decomposed into its PICO elements (Oxman et al., 1993; Richardson et al., 1995). Along with other aspects, such as study design, PICO elements are useful for formulating search queries for literature database searches (Schardt et al., 2007) and mentions of PICO elements are key to screening the search results for relevance.

A standard approach for systematic reviews (and other review types such as rapid reviews (Wagner et al., 2017) and scoping reviews (Shemilt et al., 2014)) is to perform screening initially using only the title and abstracts of a reference collection before obtaining and analysing a subset of full-text articles (Higgins and Deeks, 2011). While faster and more cost-effective than full-text screening, manually screening all reference abstracts is a protracted process for large collections (Allen and Olkin, 1999), especially those with low specificity (Lefebvre et al., 2013). Technology-assisted reviewing seeks to foreshorten this process by only screening the subset of collection most likely to be relevant (O'Mara-Eves et al., 2015; Shemilt et al., 2016; Kanoulas et al., 2017, 2018). This subset is automatically selected using information from a manual screening decisions either on another, ideally smaller, subset of the collection (Cohen et al., 2006) or through multiple rounds of iterative feedback between a ML model and the human reviewer (Wallace et al., 2010). In effect, the machine "reads" the title and

abstract and scores the relevancy of the reference based on a model trained on relevant and irrelevant examples from the human reviewer. While previous studies (Shemilt et al., 2014; Rathbone et al., 2015; Przybyła et al., 2018) have shown the potential for time-savings, the underlying models treat each word equally and do not explicitly distinguish PICO elements within an abstract. As PICO elements are crucial for a human reviewer to making inclusion decisions or design screening filters (Tsafnat et al., 2018), ML models with information on each reference's PICO will be helpful compared with models lacking this information. Therefore, we employ our neural model to automatically identify text describing PICO elements within titles and abstracts.

4.2.4.2 Related work

Previous work has shown that there are multiple avenues for automation within systematic reviews (Thomas et al., 2011; Tsafnat et al., 2014; Beller et al., 2018). Examples include retrieval of high-quality articles (Aphinyanaphongs and Aliferis, 2003; Aphinyanaphongs et al., 2005; Choi et al., 2012; Del Fiol et al., 2018), risk-of-bias assessment (Marshall et al., 2015a,b; Millard et al., 2015; Zhang et al., 2016), and identification of randomised control trials (Cohen et al., 2015; Marshall et al., 2018). As our work focuses on data extraction in abstract-level screening, we review previous work (Jonnalagadda et al., 2015) to automatically isolate PICO and other study characteristics. The two are related since inclusion and exclusion criteria can be decomposed into requirements for PICO and study characteristics to facilitate search (Sim et al., 2014).

Extracting PICO elements at the phrase level (Hara and Matsumoto, 2007; Summerscales et al., 2009, 2011) is a difficult problem due to the disagreement between human experts on the exact words constituting a PICO mention (Niu and Hirst, 2004; Demner-Fushman and Lin, 2007). Thus, many approaches (Jonnalagadda et al., 2015) firstly determine the sentences relevant to the different PICO elements, using either rules or ML models (Demner-Fushman and Lin, 2005, 2007; Hara and Matsumoto, 2007; Xu et al., 2007; Kim et al., 2011; Boudin et al., 2010a,b; Zhao et al., 2010). Fine-grained data extraction can then be applied to the identified sentences to extract the words or phrases for demographic information (age, sex, ethnicity, etc.) (Xu et al., 2007; Hara and Matsumoto, 2007; Zhao et al., 2010, 2012; Kelly and Yang, 2013), specific intervention

arms (Chung, 2009b), or the number of trial participants (Hansen et al., 2008). Instead of classifying each sentence independently, the structured form of abstracts can be exploited by identifying PICO sentences simultaneously with rhetorical types (aim, method, results, and conclusions) in the abstract (Chung and Coiera, 2007; Chung, 2009a; Dernoncourt et al., 2016; Jin and Szolovits, 2018). More broadly, PICO and other information can be extracted directly from full text articles (De Bruijn et al., 2008; Kiritchenko et al., 2010; Hsu et al., 2012; Bui et al., 2016; Wallace and Marshall, 2016).

Rather than extract specific text, Singh et al. (2017) predict which medical concepts in the Unified Medical Language System (UMLS) (Bodenreider, 2004) are described in the full-text for each PICO element. They use a neural network model that exploits embeddings of UMLS concepts in addition to word embeddings. The predicted concepts could be used as alternative features rather than just the extracted text. This would supplement manually added metadata such as Medical Subject Headings (MeSH) (Aronson et al., 2004), which are not always available or have the necessary categorisations.

Our approach differs from existing by both operating at the subsentence level (words and phrases) and using a neural model for processing text (Collobert et al., 2011b) without hand-engineered features. Our neural model jointly extracts PICO elements in theory, can provide higher recall than methods that do not allow nested PICO elements.

Recently, Tsafnat et al. (2018) have shown the screening ability of automatic PICO extraction for systematic reviews. They use manually designed filters (e.g., dictionaries and rules) (Karystianis et al., 2014, 2017) for key inclusion criterion, mentions of specific outcomes, population characteristics, and interventions (exposures) to filter collections with impressive gains. A variety of ML models have been proposed for screening references for systematic reviews (Cohen et al., 2006; Cohen, 2008; Cohen et al., 2010; Bekhuis and Demner-Fushman, 2010, 2012; Bekhuis et al., 2014; Matwin et al., 2010; Frunza et al., 2010, 2011; Wallace et al., 2010; Small et al., 2011; Wallace et al., 2012; Jonnalagadda and Petitti, 2013; Dalal et al., 2013; Miwa et al., 2014; Timsina et al., 2016; Khabsa et al., 2016; Hashimoto et al., 2016; Howard et al., 2016). However, to our knowledge none of relevancy classifiers have used as input the output of PICO recognition.

4.2.4.3 Relevancy Classification

Method of the task contains two components: PICO recognition and relevancy classification. We use our layered BilSTM-CRF model for PICO recognition. To show how the first component (i.e., PICO recognition) interacts with the second component (i.e., relevancy classification), we briefly present the second component which was conducted by other co-authors in the manuscript.

To form the relevancy classifier, we firstly adopt logistic regression classifier which will be trained on screening decisions (represented as binary variables indicating inclusion or exclusion). Then the predictions of the classifier on the unseen references are used to prioritize them, presenting those that are most likely to be relevant. We follow the RobotAnalyst (Przybyła et al., 2018), a web-based system that uses SVM to prioritise relevant references to obtain the feature set, which contains four parts: a bag-of-words (BOW) representation and topic distribution of the combination of title and abstract, BOW representation of the title and the extracted PICO elements. Specifically, we use leammata (base forms) of the occurring words that meet the following conditions:

- contain more than one character
- contain at least one letter or number
- not in the list of stop word[4]

to form BOWs. The BOW is a sparse binary vector representing whether or not a word occurred in the given context. We normalise each BOW with a Euclidean (L_2) norm of 1 for each reference, except when the bag is empty. PICO BOW representation is a combination of three BOWs, each of which corresponds to one type of the extracted course-grained P, I (C is merged into I), and O elements. Finer-grained spans that are recognised by the PICO model are mapped back to the basic PICO types. Topic distributions for the combination of title and abstract text are inferred from an LDA topic model (Blei et al., 2003) with $k = 300$ topics using MALLET (McCallum, 2002). The text is filtered to words consisting of alphabetic characters with initial or internal punctuation that are not on the stop word list.

[4]http://members.unine.ch/jacques.savoy/c

4.2.4.4 Experimental Setting

Evaluation Setting

We evaluate our neural model in terms of precision (P), recall (R) and F1-score (F1) on the token level. Each token is treated as an individual test case. True positives for each category are tokens in the category's span that matches the one assigned by the model, and false positives are tokens assigned to the category by the model but not in the original span. This solves the problem of comparing two spans that have matching category, but partially overlapping spans.

In addition, we also evaluate the model on the document level in terms of the set of included words to indicate whether the annotated PICO words would be captured when each document is represented as BOW with lemmata. In other words, the document-level matching tests how well individual documents could be retrieved by searching for words within specific PICO contexts.

To demonstrate the effectiveness of PICO information in improving the prioritisation of relevant references such that relevant references are presented as early as possible, we use both a two-fold relevancy prioritisation and a relevancy feedback setting. In addition, we follow previous work to quantify the performance in terms of work-saved over sampling at 95% recall (WSS@95%) (Cohen et al., 2006), which expresses how much manual screening effort would be saved by a reviewer that would stop the process after finding 95% of the relevant documents.

Data Setting

We used the PICO corpus (Nye et al., 2018) for our neural model. The corpus consists of 4,993 references, a subset of 4,512 are used for training and development (4,061/451). The remainder contains 191 for testing the coarse-grained spans. The remainder also contains 96 that were not used for training since they lacked at least one of the PICO elements, and 194 references which are part of a set of 200 assigned for testing fine-grained labelling. The PICO mentions in the corpus are annotated with the hierarchical categorisation shown in Table 4.28 where the top-level categories consist of population, intervention/comparator, and outcomes. The comparators are merged into interventions (Nye et al., 2018). The corpus is annotated in two passes: firstly, top-level spans are identified, and secondly, spans within these are further annotated with the fine-grained types.

Top-level	Patient-population-problem	Intervention/Comparator	Outcome
Fine-grained	Age	Control	Adverse effect
	Condition	Educational	Mental
	Sample size	Pharmacological	Mortality
	Sex	Physical	Pain
		Psychological	Physical
		Surgical	Other
		Other	

Table 4.28: The top-level and fine-grained PICO elements in the training set for the PICO recognition model.

In this manner, spans corresponding to the fine-grained types are nested within typically longer spans with top-level PICO types.

The DERP collections (Cohen, 2006; Pacific Northwest Evidence-based Practice, OHSU Center for Evidence-Based Policy) are used to test whether including the PICO features will improve the prioritisation of relevant references using simulated screening. Table 4.29 describes the collections for the different reviews.

Model Setting

For the PICO recognition model (i.e., our neural model), we initialise the word embeddings using Chiu et al. (2016a) with updating them. The dimension of the character-based word embedding is set as 56. The number of hidden units in BiLSTM is set as 256. Stochastic first-order optimisation is performed using batches of sentences, gradient clipping, and Adam (Kingma and Adam, 2015). Dropout (Srivastava et al., 2014), weight decay (L_2-regularisation), and early stopping are employed to prevent overfitting. Hyper-parameters are selected using Bayesian optimisation (Snoek et al., 2012), using the design described in Ju et al. (2018), on a development portion of the training set with the F-score of the span-level predictions as the metric.

For relevancy classification, we use the RobotAnalyst framework (Przybyła et al., 2018) as the simulation platform, where the relevancy classifiers updated at multiple stages during the screening process. Specifically, we run 100 Monte Carlo simulations. In each simulation, we begin with a random batch of 25 references. If this batch contains any relevant references, this forms the initial training set, otherwise batches of 25 are sampled randomly and appended to the training set until at least one relevant reference is found. Given the training set,

Review	Inclusion	Exclusion	Total	Prevalence (%)
ACE Inhibitors	2544	41	2503	1.61
ADHD	851	20	831	2.35
Antihistamines	310	16	294	5.16
Atypical Antipsychotics	1120	146	974	13.04
Beta Blockers	2072	42	2030	2.03
Calcium Channel Blockers	1218	100	1118	8.21
Estrogens	368	80	288	21.74
NSAIDS	393	41	352	10.43
Opioids	1915	15	1900	0.78
Oral Hypoglycemics	503	136	367	27.04
Proton Pump Inhibitors	1333	51	1282	3.83
Skeletal Muscle Relaxants	1643	9	1634	0.55
Statins	3465	85	3380	2.45
Triptans	671	24	647	3.58
Urinary Incontinence	327	40	287	12.23

Table 4.29: DERP systematic review descriptive statistics. Abbreviated columns correspond to the number of inclusions (relevant references), exclusions, total number of references, and the prevalence (percentage of inclusions compared to total).

a classifier is trained and applied to the remaining references. The references are prioritised by the classifier's score, which is proportional to the posterior probability of being relevant (using a logistic regression model). The 25 highest ranked references are then included in the training set, a classifier is retrained, and so on. This continues until all references are screened. This iterative process is readily comparable to relevance feedback methods (Salton and Buckley, 1990).

4.2.4.5 Results

Table 4.30 shows the token-wise performance for the three categories. The model achieves an F1-score of 0.70, 0.70 and 0.56 for element of P, O and I, respectively. Compared with element of P and O, the low F-score of element I is caused by the low recall which is 0.47. The performance metrics are higher for document-level matching. For element O, our neural model achieved 0.81 in terms of recall.

Table 4.31 shows the results of relevancy feedback experiment with the column labelled LR corresponding to the baseline set of features from RobotAnalyst (Przybyła et al., 2018) with logistic regression, and PICO indicating the model with the additional PICO BOW features. On average, the inclusion of PICO features increases the work-saved metric by 3.3%, with substantial gains for the

Element	Token-wise			Document-level		
	P	**R**	**F1**	**P**	**R**	**F1**
Participants	0.81	0.62	0.70	0.86	0.71	0.78
Interventions	0.69	0.47	0.56	0.83	0.52	0.64
Outcomes	0.66	0.75	0.70	0.73	0.81	0.77

Table 4.30: PICO recognition performance in terms of a token-wise evaluation and a document-level filtered bag-of-words (BOW) on the test set.

Opioids and *Triptans* collections.

We compare these results against two baselines (Ji and Yen, 2015; Ji et al., 2017) that use relevancy feedback rather than ML. The first baseline is a relevance feedback system exploiting the lexical network induced by shared word occurrence (Ji and Yen, 2015). Ji et al. (2017) follow the same experiment and for a fair comparison we report their results for the case when parameters are fixed across collections using SNOMED-CT[5] and MeSH[6] features for a semantic network. The overall performance with the PICO features is comparable to the semantic network based relevance feedback (Ji et al., 2017). This is encouraging since the latter uses a human selected seed query, versus the random initialisation for the proposed method.

To compare against other baselines from the literature, we adopt a stratified two-fold setting, where half of the inclusions and half of the exclusions are used for training. The first baseline (Matwin et al., 2010) uses a naive Bayes classifier, and the reported values are the average across five two-fold cross-validations, in each of the 10 runs the WSS value for a threshold with at least 95% recall is reported. The second baseline is a SVM-based model (Cohen, 2008, 2011) with the feature set that performed the best consisting of abstract and title text, MeSH terms, and Meta-map phrases. The final baseline (Howard et al., 2016) uses cross-validation on the training sets to select the following hyperparameters: the number of topics, the regularisation parameter, and the inclusion or exclusion of additional bigram, trigram, or MeSH term features.

The results are reported in Table 4.32. The inclusion of PICO features improves the work-saved performance metric versus the default logistic regression model, with an average improvement of 1.6%. The results are competitive against the earlier baselines, but the cross-validation selection of hyperparameters

[5]http://www.snomed.org
[6]https://www.nlm.nih.gov/mesh/meshhome.html

	Ji and Yen (2015)	Ji et al. (2017)	**LR**	**PICO**	Δ
ACE Inhibitors	74.3	*82.7	74.7	74.4	-0.3
ADHD	67.9	*82.1	67.5	68.9	1.4
Antihistamines	*24.5	17.7	-1.7	-1.9	-0.1
Atypical Antipsychotics	18.0	*33.6	18.0	20.5	2.5
Beta Blockers	65.0	*68.5	54.7	55.7	1.1
Calcium Channel Blockers	17.3	12.8	*47.6	47.1	-0.5
Estrogens	22.6	28.5	36.6	*39.1	2.4
NSAIDS	*77.4	64.1	60.9	63.1	2.2
Opioids	9.0	17.4	19.5	*34.1	14.6
Oral Hypoglycemic	13.5	*15.9	6.9	9.2	2.3
Proton Pump Inhibitors	19.7	21.0	*21.2	18.3	-2.9
Skeletal Muscle Relaxants	*58.6	29.9	25.9	32.4	6.5
Statins	27.8	*43.7	42.9	43.3	0.3
Triptans	39.6	*54.1	34.3	52.4	18.1
Urinary Incontinence	20.8	41.6	44.8	*46.4	1.6
Average	37.1	40.9	36.9	40.2	3.3

Table 4.31: Relevancy feedback performance in terms of *WSS*@95% on DERP systematic review collections. Δ indicates the change between incorporation of PICO features to the baseline logistic regression classifier (LR). Positive values reflect the amount of human effort that can be saved with PICO features. Negative values reflect the additional of human effort that requires with PICO features. * indicate best performance per review.

	Matwin et al. (2010)	Cohen (2011)	Howard et al. (2016)	**LR**	**PICO**	**Δ**
ACE Inhibitors	52.3	73.3	*80.1	78.5	77.6	-0.9
ADHD	62.2	52.6	*79.3	75.5	74.5	-0.9
Antihistamines	14.9	*23.6	13.7	4.9	5.0	0.1
Atypical Antipsychotics	20.6	17.0	*25.1	19.9	20.9	1.0
Beta Blockers	36.7	46.5	42.8	*55.5	54.1	-1.4
Calcium Channel Blockers	23.4	43.0	*44.8	38.8	39.3	0.6
Estrogens	37.5	41.4	*47.1	41.0	43.7	2.7
NSAIDS	52.8	67.2	*73.0	65.3	66.5	1.2
Opioids	55.4	36.4	*82.6	53.3	57.0	3.7
Oral Hypoglycemic	8.5	*13.6	11.7	7.1	8.9	1.8
Proton Pump Inhibitors	22.9	32.8	*37.8	32.6	31.0	-1.6
Skeletal Muscle Relaxants	26.5	37.4	*55.6	40.1	45.3	5.3
Statins	31.5	*49.1	43.6	42.2	44.3	2.1
Triptans	27.4	34.6	41.2	40.6	*51.2	10.5
Urinary Incontinence	29.6	43.2	*53.0	52.4	52.4	0.0
Average	33.5	40.8	48.8	43.2	44.8	1.6

Table 4.32: Two-fold relevancy prediction in terms of *WSS*@95%. Δ indicates the change between incorporation of PICO features to the baseline logistic regression classifier (LR). Positive values reflect the amount of human effort that can be saved with PICO features. Negative values reflect the additional of human effort that requires with PICO features. * indicate best performance per review.

(Howard et al., 2016) yields the best average performance. Searching for these hyperparameters using cross-validations is computational demanding, especially in the relevance feedback setting, where there is not a large initial training set, but rather a different training set at each stage.

4.2.4.6 Discussion

Experimental results indicate that the additional PICO tagging is useful for improving ML performance in both the two-fold and relevancy feedback scenarios. This could only be the case if the additional features carry information about the relevancy decisions and are not redundant with the existing feature sets. These questions are answered by statistical analysis, which shows that when restricted to a specific PICO context certain words are more reliable predictors. As inclusion criteria are often stated in terms of PICO (and other study characteristics) this is not a surprising result, but nonetheless, requires a well-trained PICO recognition

model to transfer the knowledge from the training set of annotations. In a way, the proposed methodology connects with previous work on generalisable classifiers that can learn from the screening decisions of other systematic reviews (Cohen et al., 2009).

Furthermore, PICO tagging is an interpretable process meant to emulate human annotation and can readily be used by reviewers themselves. For instance, highlighting the mentions of outcomes may accelerate data extraction, since identifying outcome measures and data are a critical step in many systematic reviews. However, the performance for the recognition indicates room for improvement to match human annotation. In particular, the proposed PICO recognition model operates on each sentence in the title and abstract independently. Given only a sentence it may not be clear that a mention of a drug refers to an intervention; whereas, a human annotator is reading the full abstract and is able to consistently mark the intervention throughout.

4.3 Summary

In this chapter, we conducted data-based and task-specific evaluation, respectively. In the data-based setting, experiments on the ACE2005 and GENIA corpora show the effectiveness of the model in extracting nested named entities in both general and biomedical domains. Furthermore, experimental results also indicate that the use of information from inner entities improves the extraction of entities on the higher nesting levels, which can alleviate the entity sparsity issue. Experiments on the JNLPBA corpus demonstrate that the model is comparable with state-of-the-art flat NER models in extracting flat entities. In the first task, we applied the model to help curation of neuroscience entities. In comparison with related work, the model achieved comparable performances based on both strict and lenient metrics. In combination with active learning, our model enables researchers to create competitive NER tools within a specific domain using a very small set of annotations, saving time and cost. In addition to entity curation, the extraction of nested COPD phenotypes demonstrates its application ability in different tasks without requiring additional human efforts. Apart from COPD phenotypes, the model can be used to extract other different disease phenotypes. In the task of extracting adverse drug event and medication information, the

overall results have demonstrated that the model is capable of accurately recognizing entities, including nested and polysemous entities. Additionally, the model enables recognition of sparse entities by reconsidering the clinical narratives at a finer-grained subword level, rather than at the word level. In the last task setting, the incorporation of PICO elements identified by the model demonstrates that words within PICO tagged segments in abstracts are predictive features for determining inclusion. Those predictive features are helpful for users to screen relevant abstracts for systematic reviews.

Chapter 5

Temporally Relating Named Entities

In Chapter 4, we evaluated our nested NER model with data-based settings, showing its effectiveness in extracting structured information from textual data. In addition, the task-specific setting further demonstrates the significance of entities in different NLP tasks. In addition to entity semantics, temporality between entities is another crucial aspect in understanding text. Many NLP tasks such as question answering (Llorens et al., 2015; Meng et al., 2017), text summarisation (Ng et al., 2014; Wang et al., 2017) and causality (Mirza and Tonelli, 2014; Mirza, 2014; Ning et al., 2018) require extracting information in a time dimension. For example, to understand the disease progression, we need temporal information such as the starting time points of symptoms and disease history. Meanwhile, when summarizing the storyline from news reports, it is necessary to know the development of events over time, requiring time-stamping or temporally ordering them. The process of identifying such information in the time dimension is defined as temporal information extraction (TIE), which in general includes the extraction of time expressions (timexes), events, and the relation between event-event, event-timex, timex-timex. In particular, events in combination with timexes are defined as temporal entities, which are a type of entities and can be nested within each other. To investigate the temporality between entities of interest, temporal entity recognition is required, which is then used for their temporality extraction. To investigate TIE, we consider temporal entities and their temporal relations (e.g., before, after) in a pipeline manner, where the temporal entity extraction can be addressed using our proposed layered-BiLSTM-CRF model described in

Chapter 3. In this chapter, we focus on the extraction of temporal relations and present the following:

- Concepts and definitions related to TIE
- An overview of the literature focusing on the methods, resources for TIE
- A neural method for identifying temporal relations using non-local information
- Experimental settings, results and discussions

5.1 Introduction

The earliest work in TIE focused on temporal relation extraction (TRE) including timestamping and temporally ordering verbs. An important work in TIE was by Reichenbach (1947), who proposed a three-point framework that uses the concepts of *speech*, *event*, and *reference* and the relations (anteriority and simultaneity) between them to describe the verb tenses, offering the foundations for temporally ordering events in terms of linguistics. TIE has been receiving great attention since the 1950s (Ebersole, 1952; Garey, 1957; Davidson, 1967; Bronckart and Sinclair, 1973; Erbaugh, 1978).

In the 1980s, Comrie (1985) extended Reichenbach (1947) by adopting three temporal orders (i.e., "simultaneous", "before" and "after") that are based on the relative reference point which does not necessarily correspond with the moment of utterance. Meanwhile, Allen (1984) created a formal classification of 13 temporal relations, as shown in Figure 5.1. Inspired by Allen's work, Allen and Hayes (1989) presented a concise, formal axiomatization of "interval-based" time as described by Allen (1984) and further investigated the relationship between interval-based and point-based temporal theories.

In the 1990s, TIE was extended to include the timex extraction, which was formally proposed in MUC-6 (Grishman and Sundheim, 1996) in the form of NER. MUC-6 (Grishman and Sundheim, 1996) considered only absolute timexes (e.g., date, time), which were further expanded to include relative timexes in the MUC-7 (Marsh and Perzanowski, 1998). To enhance answering temporally-based questions, the Time and Event Recognition for Question Answering Systems (TERQAS) workshop (Pustejovsky, 2002) conceptualized time mark-up

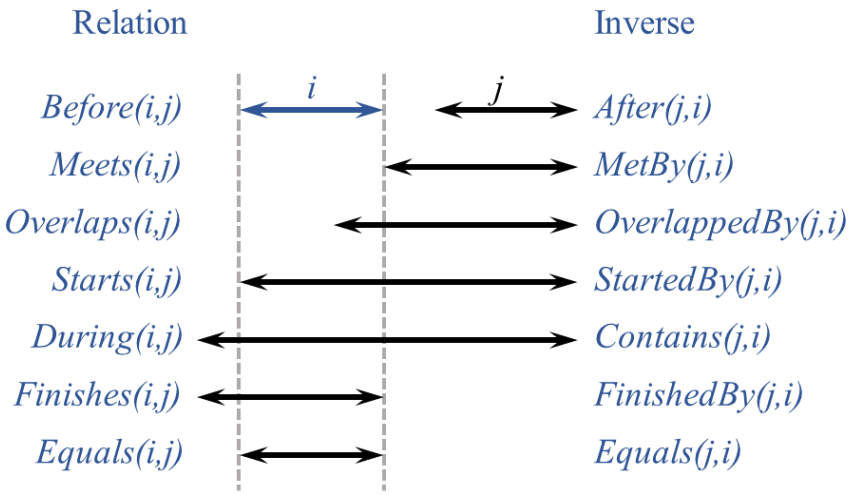

Figure 5.1: Thirteen elementary possible relations between time periods (Allen and Ferguson, 1994).

language (TimeML) (Ingria and Pustejovsky, 2002) for annotating timexes and events, which was further used for annotating the TimeBank corpus (Pustejovsky et al., 2003a). Following TERQAS, the TimeML Annotation Graphical Organizer workshop (Pustejovsky et al., 2003b) developed a graphical annotation tool for temporal annotation. An annotated example from TimeBank corpus is shown in Figure 5.2, where events and timexes are tagged with "EVENT" and "TIMEX3" labels, respectively. The "SIGNAL" tag represents a temporal signal, which are function words that specify the modality (e.g. modal auxiliaries) or suggest a particular temporal relation (e.g., after). Based on the latest TimeML (Boguraev et al., 2005), a revision of Ingria and Pustejovsky (2002), the temporal relation is denoted by the "TLINK" label coupled with attributes linking it to the temporal entities. Each event is annotated with five attributes: class, tense, aspect, polarity and part-of-speech (POS) in addition to the IDs (e.g., eid, eiid). Each timex, however, is mainly annotated with three types of attributes: type, value and functions.

TimeML (Boguraev et al., 2005) and TimeBank (Pustejovsky et al., 2003a) have been widely used for temporal evaluation (TempEval) tasks such as Semantic Evaluation (SemEval)-2007 Task 15 (**TempEval-1**) (Verhagen et al., 2007), SemEval-2010 Task 13 (**TempEval-2**) (Verhagen et al., 2010), SemEval-2013 Task 1 (**TempEval-3**) (UzZaman et al., 2013), which split TIE into three main

Gas prices <**EVENT** **eid**="e18" **class**="OCCURRENCE">fell </**EVENT**>nearly two cents a gallon <**SIGNAL** **sid**="s137">over</**SIGNAL**><**TIMEX3** **tid**="t25" **type**="DURATION" **value**="P2W" **temporalFunction**="true" **functionInDocument**="NONE" **endPoint**="t19">the last two weeks</**TIMEX3**>.
<**MAKEINSTANCE** **eventID**="e18" **eiid**="ei167" **tense**="PAST" **aspect**="NONE" **polarity**="POS" **pos**="VERB"/>
<**TLINK** **lid**="l4" **relType**="IS_INCLUDED" **eventInstanceID**="ei167" **relatedToTime**="t25" **signalID**="s137"/>

Figure 5.2: A sentence annotated with TimeML.

She has had similar pain intermittently for last year.
<**TIMEX3** **id**="T6" **start**="920" **end**="929" **text**="last year" **type**="DURATION" **val**="p1y" **mod**="NA"/>
<**EVENT** **id**="E25" **start**="888" **end**="900" **text**="similar pain" **modality**="FACTUAL" **polarity**="POS" **type**="PROBLEM"/>
<**TLINK** **id**="TL40" **fromID**="E25" **fromText**="similar pain" **toID**="T6" **toText**="last year" **type**="OVERLAP"/>

Figure 5.3: An annotated sentence from i2b2 corpus.

tasks: (i) timex extraction, (ii) event extraction and (iii) temporal relation extraction. The third task is further categorized into four subtasks, requiring the recognition of temporal relations between (a) events and timexes within the same sentence (b) events and the document creation time (**DCT**) (c) main events in adjacent sentences, and (d) two events where one syntactically dominates the other. Moreover, the 2012 Informatics for Integrating Biology and the Bedside (**i2b2**) shared task established the first clinical TempEval (**CliTempEval-1**) corpus (Sun et al., 2013a), which was provided for the participants to develop and evaluate models dealing with the above tasks (Sun et al., 2013b). A temporally annotated sentence from i2b2 corpus is shown in Figure 5.3. In addition to i2b2 corpus, the Temporal Histories of Your Medical Event (**THYME**) project built another temporally annotated corpus (Styler IV et al., 2014b), which served as the data set in clinical TempEval: SemEval-2015 task 6 (**CliTempEval-2**) (Bethard et al., 2015), SemEval-2016 task 12 (**CliTempEval-3**) (Bethard et al., 2016), SemEval-2017 Task 12 (**CliTempEval-4**) (Bethard et al., 2017). In the next section, we present the literature review for TIE.

5.2 Literature Review

5.2.1 Time Expression Extraction

The earliest methods for timex extraction (**TEE**) were from MUC-6 (Grishman and Sundheim, 1996), where researchers mainly employed hand-designed rules such as regular expressions to match the fixed-format timexes (Gaizauskas et al., 1995; Fisher et al., 1995; Aberdeen et al., 1995; Eriguchi and Kitani, 1996). MUC-7 (Marsh and Perzanowski, 1998), a follow-up of MUC-6, improved TEE with new approaches such as finite state transducers (Borthwick et al., 1998; Mikheev et al., 1998) and HMMs (Miller et al., 1998). The encouraging performance of ML algorithms in combination with rules received great attention from the community.

In the early 2000s, Mani and Wilson (2000) designed the first timex tagger TempEx, which used POS tag-based heuristics for extracting timexes. TempEx tagger ignored the informative temporal information expressed by prepositions (e.g., Monday vs by Monday) in timexes. As an enhancement, Schilder and Habel (2001) employed finite state transducers coupled with hand-crafted rules for TEE. The ACE organised the shared task: the 2004 edition of the Temporal Expression Recognition and Normalization Evaluation (Negri and Marseglia, 2004), extended the category of timexes covering absolute timexes, relative timexes, and phrases that contain temporal words such as daily, former, future, making the TEE more challenging. To deal with such timexes, Verhagen et al. (2005a) developed GUTime, which extended TempEx (Mani and Wilson, 2000) to construct an automatic temporal annotation tool for TimeML (Pustejovsky et al., 2005). In addition to rule-based methods, ML-based methods such as SVM (Hacioglu et al., 2005) and CRFs (Ahn et al., 2005) were employed in TEE.

In the early 2010s, Strötgen and Gertz (2010) developed HeidelTime that depends on regular expressions to extract timexes. They further incorporated knowledge resources and linguistic clues for timex normalisation, achieving the highest performance in TempEval-2 (Verhagen et al., 2010). Similarly to HeidelTime, SUTime (Chang and Manning, 2013) also adopted rules to extract timexes. Both HeidelTime and SUTime focused on the timexes in English only. Unlike TempEval-2, TempEval-3 (UzZaman et al., 2013) covered both English and Spanish for TEE, where a wide range of approaches were investigated including ML-based (Llorens et al., 2010; Filannino et al., 2013; Jung and Stent,

2013; Kolya et al., 2013), rule-based (Strötgen and Gertz, 2010; Chang and Manning, 2013; Zavarella and Tanev, 2013; Chambers, 2013) and their combination (Kolomiyets and Moens, 2013; Bethard, 2013). To enable language-independent TEE, Strötgen and Gertz (2015) extended HeidelTime to cover more than 200 languages without requiring language skills nor training data.

Apart from general domain, TEE has been receiving attention from other domains such as medicine. The i2b2 (Sun et al., 2013b) was the first to provide a temporally annotated corpus consisting of clinical records. Following i2b2, SemEval-2015 organized the CliTempEval-2 (Bethard et al., 2015) to bring past temporal shared tasks (Verhagen et al., 2007, 2010; Sun et al., 2013b; UzZaman et al., 2013) to the clinical domain, using clinical notes and pathology reports from the Mayo Clinic. In this task, Velupillai et al. (2015) achieved the best performance using SVM. In CliTempEval-3 (Bethard et al., 2016), Lee et al. (2016a) applied the HMM-SVM sequence tagger (Joachims et al., 2009) in combination with various features (e.g., lexical, syntactic, dictionary-based, discourse, etc.), outperforming the other participating systems (Hansart et al., 2016; Khalifa et al., 2016; MacAvaney et al., 2017; Chikka, 2016; Sarath et al., 2016; Grouin and Moriceau, 2016). Differently from CliTempEvals (Sun et al., 2013b; Bethard et al., 2015, 2016), CliTempEval-4 (Bethard et al., 2017) focused on the adaptation from colon cancer to brain cancer domain, requiring systems to be trained on data from colon cancer patients, and then make predictions on data from brain cancer patients. MacAvaney et al. (2017) approached TEE by employing CRFs and decision trees in combination with features (e.g., lexical, syntactic, semantic, distributional, and rule-based features), achieving the best performance among all systems (Leeuwenberg and Moens, 2017; Huang et al., 2017; Lamurias et al., 2017; Long et al., 2017; Tourille et al., 2017b; Sarath et al., 2016). In particular, Tourille et al. (2017b) explored neural models by combining RNN (Goller and Kuchler, 1996) with SVM for TEE.

As neural networks have achieved great success in many NLP tasks (Miwa and Bansal, 2016; Christopoulou et al., 2018; Ju et al., 2019a), there is more work with neural models for identifying timexes. Lin et al. (2017b) presented a CNN-based TEE model, which is depicted in Figure 5.4, establishing a new state-of-the-art on the THYME corpus (Styler IV et al., 2014b). More recently, (Goyal and Durrett, 2019) generated synthetic data consisting of pairs of timexes, then

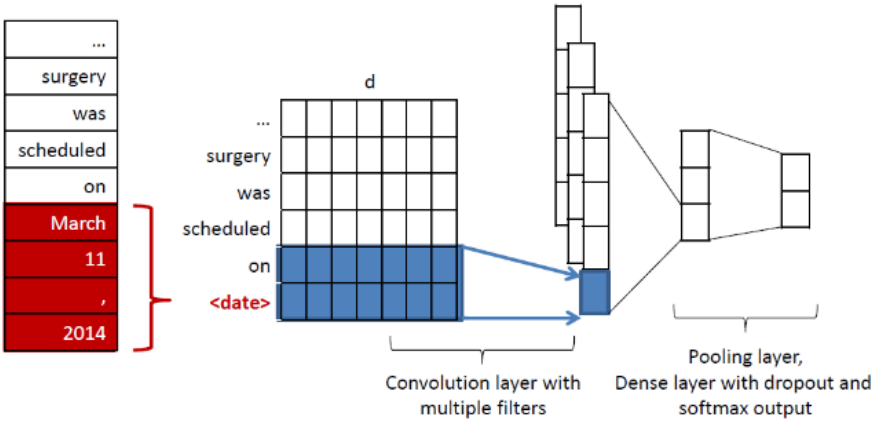

Figure 5.4: CNN with encoded time expressions (Lin et al., 2017b).

trained a character LSTM to learn time embeddings through classifying the temporal relations between timexes. Those time embeddings were evaluated in the context of temporal event ordering, showing the effectiveness in the downstream temporal tasks.

5.2.2 Event Extraction

According to TimeML (Ingria and Pustejovsky, 2002), an event is a cover term for situations that happen or occur, including predicates describing states or circumstances in which something obtains or holds true. Briefly, an event is something that happens or something that can be attached to a timestamp. The meaning of an event is domain-specific and application-dependent. In clinical NLP, events can be clinically relevant activities (Sun et al., 2013b), e.g. treatment, medication, diseases, etc. In general NLP, however, events have to be anchorable in time and they are conveyed in the form of finite verbs and their nominalisation. In general, event extraction is cast as a sequence labelling task, i.e. NER. Therefore, methods of NER include rule-based, learning-based, neural-based that were discussed in Chapter 2 can be used for identifying events. Work dealing with event extraction were mainly from TempEvals (Verhagen et al., 2010; UzZaman et al., 2013; Bethard et al., 2015, 2016, 2017), where various rule-based methods (Zavarella and Tanev, 2013), learning-based (Llorens et al., 2010; Velupillai et al., 2015; MacAvaney et al., 2017) methods and neural methods (Li and Huang, 2016; Tourille et al., 2017b) have been developed.

5.2.3 Temporal Relation Extraction

The earliest work for TRE, to the best of our knowledge, was by Reichenbach (1947), who used the concepts of *speech*, *event*, and *reference* and the relations (anteriority and simultaneity) between them. These concepts in combination with relations inspired researchers to temporally order events based on linguistics. Since then, ordering events and relating events to timestamps has been receiving much attention from the area of computational linguistics and NLP (Ebersole, 1952; Garey, 1957; Davidson, 1967; Bronckart and Sinclair, 1973; Erbaugh, 1978).

In the 1980s, Allen (1984) pushed forward TRE by creating a formal set of 13 temporal relations. Later on, Dowty (1986) proposed to use temporal subordinate clauses or tenses to indicate the temporal relationship between the events and states that were described in successive sentences in narrative discourse. Lascarides and Asher (1993) determined the relations between events using complex semantic knowledge such as prior knowledge, making it extremely challenging for practical systems to encode all available knowledge (Hitzeman et al., 1995). Following this work, work for TRE faded away until the emergence of TimeML (Ingria and Pustejovsky, 2002), which was used for annotating TimeBank (Pustejovsky et al., 2003a). TimeBank corpus is used as the data resource for TempEvals.

TempEval-1 (Verhagen et al., 2007) was an initial evaluation shared task focusing only on the categorization of temporal relations in English. Specifically, it introduced three TRE-related tasks:

A predict temporal relation types between the given timexes and events within the same sentence,

B predict temporal relation types between the given document creation time and events across the sentences,

C predict temporal relation types between the given events and events, where events are individually main events located in the adjacent sentences.

The best system over these three tasks was CU-TMP (Bethard and Martin, 2007), which adopted SVM in combination with features derived from annotations and the text.

TempEval-2 (Verhagen et al., 2010) extended TempEval-1 into a multilingual task, covering four TRE subtasks. This included the three tasks proposed in TempEval-1 with an addition of the following task:

D) predict the temporal relation types holding between given events and events,

where one event subordinates another.

Among all participating systems, the TIPSem (Llorens et al., 2010) used CRF models, achieving the best performance in Task ***A***) and ***B***) in Spanish where data for Task ***C***) and ***D***) was unavailable. Moreover, Kumar Kolya et al. (2010) used CRFs over the four tasks, ranking first in tasks ***A***, ***B*** and ***C*** while second in task ***D***. The best system for task ***D*** was Ha et al. (2010), who adopted Markov logic (Richardson and Domingos, 2006) in combination with rich lexical relations beyond basic and syntactic features.

CliTempEval-1 (Sun et al., 2013b) was the first TRE evaluation shared task in the clinical domain. In comparison with TempEval-1 and 2, the TRE track organised by i2b2 was more complicated in the sense that any two entities in a discharge summary can be a candidate pair to assign temporal relations to. In other words, candidate pairs (e-e, e-t, t-t) include both intra-sentence and inter-sentence ones while only pairs of main events in adjacent sentences (e-e) were included in the TempEval-1 and 2. We name this task as task $\boldsymbol{C}^{+}$ in differentiation with ***C***. Additionally, the i2b2 organizer introduced a new task: ***E***) identify timexes and events from different documents, then determine the temporal relations between them.

In CliTempEval-1, Tang et al. (2013) ensembled CRF, SVM and rules to approach task ***E*** and $\boldsymbol{C}^{+}$, outperforming the rest of the systems (Xu et al., 2013; DSouza and Ng, 2013; Grouin et al., 2012).

TempEval-3 (UzZaman et al., 2013) was a follow-up to TempEval-1 and 2, covering English and Spanish. Besides task ***A***, ***B***, $\boldsymbol{C}^{+}$ and ***E***, TempEval-3 adopted a full set of temporal relations, which were reduced in the TempEval-1 and 2. In addition, TempEval-3 used a different test set and introduced a new set of evaluation metrics to indicate the identification and categorization of temporal relations. In particular, the new evaluation metrics are defined in the following equations:

$$\text{Precision} = \frac{\text{Sys}^{-}_{relation} \cap \text{Ref}^{+}_{relation}}{\text{Sys}^{-}_{relation}} \tag{5.1}$$

$$\text{Recall} = \frac{\text{Ref}^{-}_{relation} \cap \text{Sys}^{+}_{relation}}{\text{Ref}^{-}_{relation}}, \tag{5.2}$$

where G^{+} is the closure of G and G^{-} excludes the redundant relations, which can be inferred through other relations (UzZaman et al., 2013). In particular, temporal closure, indicated by Allen (1984), is a reasoning mechanism that derives

new implicit temporal relations from explicit temporal relations (UzZaman and Allen, 2011). Based on the temporal closure, if event A happens before B and event B happens before C, then A happens before C. Such properties are known as temporal transitivity (Allen, 1984).

SemEval-2015 Task 4 (**TempEval-4**) (Minard et al., 2015), which aimed to build timelines from written news in English, introduced two tasks:
F: build the timeline for each target event that occurs cross documents from raw text sources
G: given the gold entities, build the timeline for each given gold entity that occurs cross documents.
A timeline for a specific target entity consists of an ordered list of the events in which that entity participates. To focus on temporal relations between events, the organizer additionally proposed two subtasks corresponding to tasks ***F*** and ***G*** by defining that events do not need to be associated with a time anchor. For task ***G*** and its subtask, the top system GPLSIUA (Navarro and Saquete, 2015) adopted TIPSem (Llorens et al., 2010) in combination with K-means clustering method (Lloyd, 1957) to construct timelines for each gold entity, achieving 25.36% and 23.25% in terms of F1-score, respectively. Unlike task ***G***, the highest performances in task ***F*** and its subtask were much lower with the corresponding F1-score of 7.28% [1] and 1.69% (Caselli et al., 2015).

CliTempEval-2 was a follow-up of CliTempEval-1, which brought TRE related tasks to the clinical domain, using clinical notes and pathology reports from the MayoClinic. This shared task focused on the identification of temporal relations between events and timexes only, which included task ***B*** and $\boldsymbol{C^+}$. In particular, task $\boldsymbol{C^+}$ in CliTempEval-2 dealt with the relation type "CONTAINS" only, which indicates that an event or timex is temporally contained in (i.e., occurred during) another event or timex. Each task here included two cases: with and without given gold temporal entities. Velupillai et al. (2015) presented a CRF-based method that takes input from a rule-based system (Bethard, 2013), yielding the highest F1-score in task ***B*** on both cases.

In task $\boldsymbol{C^+}$, however, the rule-based baseline from the organiser produced the highest performance with an F1-score of approximately 10% without using gold entities while around 26% F1-score was obtained when using gold entities. One reason for the low performances was that the task $\boldsymbol{C^+}$ in the clinical domain often

[1] The task participants did not submit a paper with the description of the system.

spans many sentences, while almost all of the relations in TempEval-3 were across adjacent sentences at maximum. Considering the small number of participants, it was likely because those participants had to complete system development and the time-consuming data use agreement process in a short time (six months or less), making it difficult to cover multiple subtasks (Bethard et al., 2015).

CliTempEval-3 presented the same tasks with CliTempEval-2 but shortened the data access procedure by using the available training and development data from Bethard et al. (2015) for model development. As a result, greater improvements were made especially for task $\boldsymbol{C^+}$, with the best F1-score of 47.9% (Lee et al., 2016a) when using raw texts. When using gold entities, the best system (Lee et al., 2016a) produced 57.3% in terms of F1-score, significantly reducing the gap between systems and human annotations.

CliTempEval-4 (Bethard et al., 2017) conducted the same shared tasks with CliTempEval-2 and 3, but introduced a new aspect: domain adaptation, which required the systems to be trained on annotated timelines for colon cancer domain and predict timelines on brain cancer domain. Among all participants, the LIMSI-COT system (Tourille et al., 2017b) obtained the highest performances in task $\boldsymbol{B}$ with an F1-score of 51% and 59% in the corresponding unsupervised and supervised domain adaptation settings, respectively. Furthermore, Tourille et al. (2017b) also ranked first with 32% F1-score in task $\boldsymbol{C^+}$ in the supervised domain adaptation setting. In the unsupervised domain adaptation setting, the best system developed by MacAvaney et al. (2017) achieved 34% in terms of F1-score. In comparison with human annotations, the LIMSI-COT system (Tourille et al., 2017b) was around 33% lower in terms of F1-score than inter-annotator agreement in task $\boldsymbol{C^+}$ in both supervised and unsupervised settings. As for task $\boldsymbol{B}$, Tourille et al. (2017b) reached comparable performances in the unsupervised setting but higher performances in the supervised setting.

In addition to TempEvals, much work (Verhagen et al., 2005b; Noro et al., 2006; Chambers et al., 2007; Bethard and Martin, 2008; Yoshikawa et al., 2009) had been done to improve TRE using different approaches. In the early 2010s, Ling and Weld (2010) proposed to use probabilistic inferences that used the temporal transitivity for extracting point-wise constraints on the end-point of event-intervals. Laokulrat et al. (2014) presented a two-stage approach (as shown in Figure 5.5) that used the pairwise predictions from the first stage as the features for the second stage to classify relations. Such a multi-step approach is able

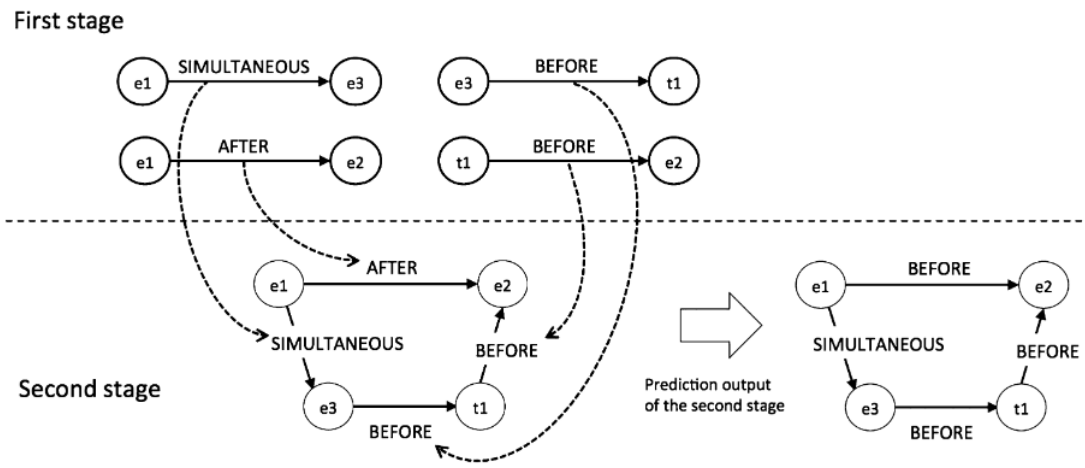

Figure 5.5: The two-step approach. The output from the first stage is treated as features for the second stage.The final output is predicted using label information of nearby relations. (Laokulrat et al., 2014)

to incorporate knowledge obtained from nearby entities by making use of time graphs where nodes represent entities and edges represent temporal relations. Lin et al. (2015) used separate SVM-based models to classify intra-sentence relations between e-e and e-t, while another rule-based method was applied to classify inter-sentence relations.

To utilize neural networks, Tourille et al. (2017a) employed the BiLSTM-Softmax architecture to extract narrative container relations on THYME corpus. On the same corpus, Dligach et al. (2017) utilized both CNN and LSTM with the addition of "tag insertion", establishing a new state-of-the-art. The "tag insertion" inserted special tokens (e.g. <e1>and </e1>) to mark the positions of target entities.

Compared with clinical corpora, inter-sentence temporal relations in general corpora are sparse. To enrich inter-sentence relation annotations, Chambers et al. (2014) selected 36 documents from TimeBank corpus (Pustejovsky et al., 2003a) to create the TimeBank-Dense corpus, where relations for all pairs in the adjacent sentences were annotated. Figure 5.6 shows an example of the annotation differences between TimeBank and TimeBank-Dense. On TimeBank-Dense corpus, Cheng and Miyao (2017) adopted BiLSTM along with shortest dependency paths (SDPs) to classify both intra- and inter-sentence temporal relations without using external knowledge and manually annotated attributes of entities (class, tense, polarity, etc.). In particular, for the target of entities in different sentences, a

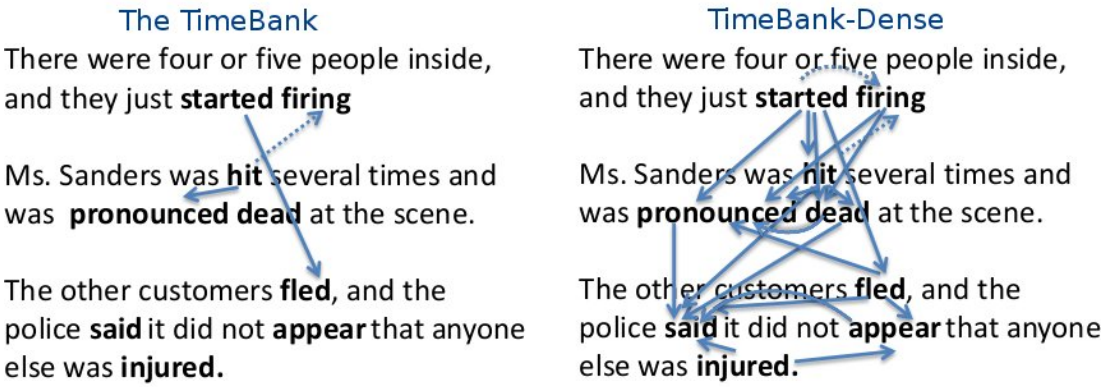

Figure 5.6: An example of annotation differences between TimeBank (Pustejovsky et al., 2003a) and TimeBank-Dense (Chambers et al., 2014). Solid and dotted arrows represent "BEFORE" and "INCLUDED_IN" relations. Relations with document creation time are not listed.

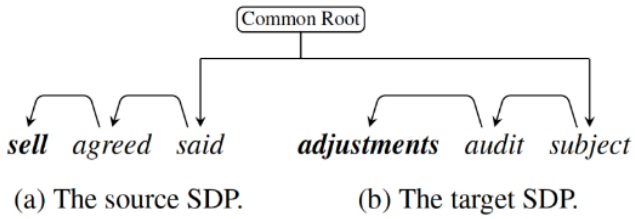

Figure 5.7: An example of the SDP representation of a cross-sentence relation between sentences.

"common root" assumption, depicted in Figure 5.7, was used to extend SDP representations for cross-sentence relations. Based on SDP, Meng et al. (2017) presented a BiLSTM-based model in combination with dependencies for identifying temporal both intra- and inter-sentence relations. This model was used by Meng and Rumshisky (2018) to make pairwise predictions for all intra-sentence pair candidates using features including dependencies, window-sized contexts, event attributes, normalised time values. Inspired by Neural Turing Machine (Graves et al., 2014), Meng and Rumshisky (2018) further introduced a global context layer (GCL) to store predictions in narrative order, and retrieve them for use when relevant entities come in, thus enabling a uniform consideration of all pairs in a wider context. The GCL, as shown in Figure 5.8, uses long-term memory and attention mechanisms to resolve long-distance dependencies, thus enabling self-correction and incorporating global context information. In addition to SDP-based neural models, Ning et al. (2018) employed constrained conditional models

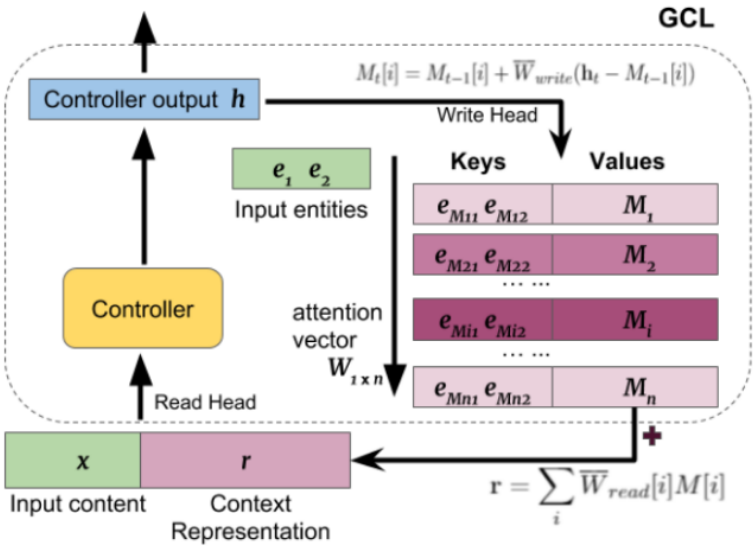

Figure 5.8: The network architecture of GCL by (Meng and Rumshisky, 2018). Input entity representations are compared to the Key section of GCL memory. Slots with the same or similar entities get more attention.

to jointly extract temporal and causal relations, aiming to utilise the fact "a cause must occur earlier than its effect". Leeuwenberg and Moens (2018) proposed to predict the events' start and end-points that were relevant to the DCT. Then all events in the document were temporally ordered according to their corresponding start and end-points.

More recently, with the advert of BERT (Devlin et al., 2019), Lin et al. (2019) presented a sentence-agnostic model that takes in a window-based sequence instead of a linguistic sentence to identify both intra- and inter-sentence relations. Meanwhile, Liu et al. (2019) presented the attention-based models for extracting intra-sentence relations on THYME (Styler IV et al., 2014b) corpus. Goyal and Durrett (2019) presented a novel framework (see Figure 5.9) to classify the temporal relations between events and timexes. In particular, the time embeddings used in Figure 5.9 were produced using the model described in Figure 5.10.

Unlike previous work, Yan et al. (2019a) focused on predicting whether two given entities participate in a relation at a given time spot. Specifically, they created the WIKI-TIME (Yan et al., 2019b) corpus, which includes the valid period of a certain relation of two entities in the knowledge base. Then they proposed a novel model (as shown in Figure 5.11) to incorporate both temporal information encoding and sequential reasoning, achieving better performances

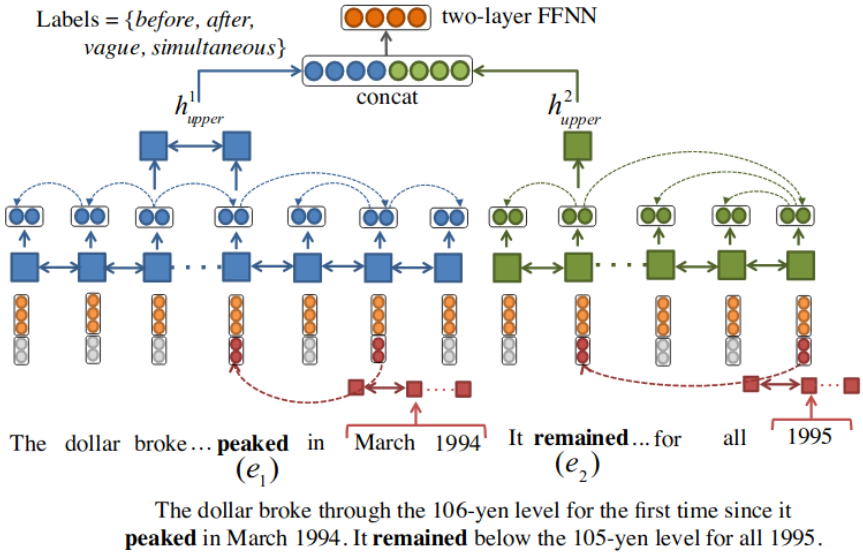

Figure 5.9: Model architecture in Goyal and Durrett (2019). In the example, *peaked* and *remained* are target events. The sentences are passed through the lower LSTM, then the outputs corresponding to the events' dependency paths are fed to the upper LSTMs, which produce input to feed forward and classification layers. Time expressions are embedded with a character-level model and broadcasted to events that they modify. In the architecture, FFNN represents feed forward neural networks.

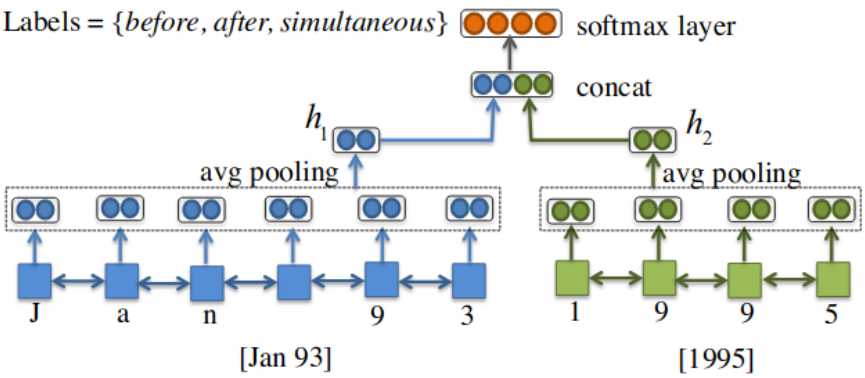

Figure 5.10: The model architecture for timex embedding (Goyal and Durrett, 2019). The output of character biLSTMs is used as input to classification. These vectors serve as time embeddings in the downstream tasks.

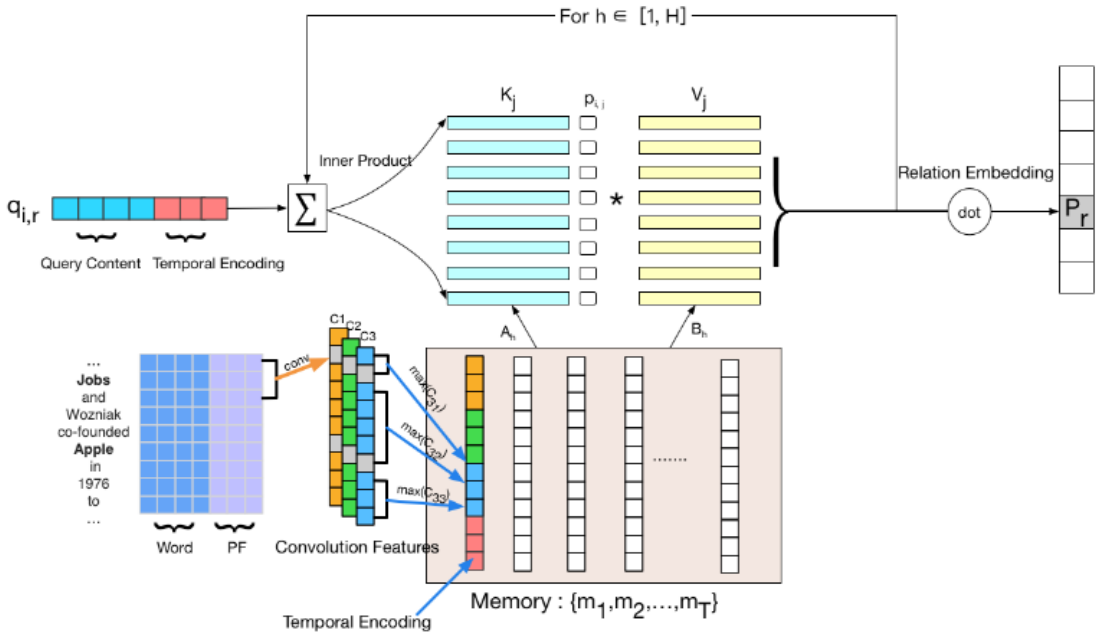

Figure 5.11: Model architecture proposed by Yan et al. (2019a).

compared with state-of-the-art models on the WIKI-TIME data set.

5.2.4 Resources

We summarise the most commonly used corpora for temporal relation identification in the following.

5.2.4.1 TimeBank

The TimeBank corpus (Pustejovsky et al., 2003a) was annotated using TimeML (Pustejovsky et al., 2005). It contains 183 news articles. Events, timexes and temporal relations between event-event, event-timex, event-DCT and timex-timex were all annotated. A total of 14 temporal relation types were included in TimeBank. We list the statistics for each temporal relation type in Table 5.1.

5.2.4.2 TimeBank-Dense

TimeBank-Dense (TBD) (Chambers et al., 2014) contains 36 news articles, which are a subset of TimeBank corpus. There are six temporal relations: BEFORE, AFTER, SIMULTANEOUS, IS_INCLUDED, INCLUDES and VAGUE. Compared with TimeBank, relations for pairs in adjacent sentences were all annotated.

Relation Type	Statistics	Relation Type	Statistics
BEFORE	1408	IBEFORE	34
BEGUN_BY	70	BEGINS	61
DURING	302	DURING_INV	1
IS_INCLUDED	1357	INCLUDES	582
SIMULTANEOUS	671	IDENTITY	743
AFTER	897	IAFTER	39
ENDED_BY	177	ENDS	76

Table 5.1: Category-wise statistics of TimeBank corpus.

Item	B	A	I	II	S	V	All
TBD	2590	2104	836	1060	215	5190	12715
TBD3 Train	1444	1148	473	629	120	3013	6827
TBD3 Dev	242	218	37	73	20	359	949
TBD3 Test	589	428	116	159	39	900	2231

Table 5.2: Category-wise statistics of TimeBank-Dense Corpus. B, A, I, II, S, V represent BEFORE, AFTER, INCLUDES, IS_INCLUDED, SIMULTANEOUS, VAGUE, respectively (Chambers et al., 2014).

In addition, all the temporal entities have a temporal relation with the document creation time. The statistics of this corpus are shown in Table 5.2. A simplified version of TBD (i.e. TBD3), which has been widely used in recent work (Chambers et al., 2014; Meng and Rumshisky, 2018; Leeuwenberg and Moens, 2018), excluded relations between entities that were not included in the UzZaman et al. (2013). In comparison purpose, we choose TBD3 corpus for our experiments to investigate intra- and inter-sentence relation extraction.

5.2.4.3 THYME Corpus

THYME corpus was annotated using "THYME Guidelines to ISO-TimeML" (THYME-TimeML) (Styler IV et al., 2014a) which was developed in the context of THYME project. The aim of THYME project is to create robust gold standards for semantic information in clinical notes, as well as to develop state-of-the-art algorithms on this data set. THYME-TimeML extended ISO-TimeML (Pustejovsky et al., 2010), which was an inheriting framework from TimeML (Boguraev et al., 2005). THYME corpus consists of clinical and pathology notes of patients with colon cancer from Mayo Clinic. In addition, it contains five types of temporal relations, whose statistics are listed in Table 5.3.

Relation Type	Statistics	Relation Type	Statistics
CONTAINS	5112	OVERLAP	1205
BEFORE	1004	BEGINS-ON	488
END-ON	126		

Table 5.3: Category-wise statistics of THYME Corpus.

Item	TimeBank	TBD3	I2b2	THYME
Event	7935	1494	30062	15769
Timex	3712	246	4186	1426
Relations	6418	10007	61371	7935
- Event-Event	3481	6088	26070	-
- Event-Timex	27976	3495	34886	-
- Timex-Timex	140	424	415	-

Table 5.4: Annotations of events, timexes and temporal relations in the four corpora. The statistics of temporal relations in THYME were absent due to no access to THYME corpus.

5.2.4.4 I2b2 Corpus

The 2012 i2b2 corpus (Sun et al., 2013a) was annotated using a simplified version of THYME-TimeML (Styler IV et al., 2014a). The corpus contains 14 temporal relations and consists of 310 discharge summaries: 190 summaries for training and 120 for testing. Within each document, two types of temporal relations were annotated: (1) event-section time, which links every event from the patient history section to the admission date and every event from the hospital course section to the discharge date; and (2) the other relation links events/timexes either from the same sentence or from different sentences using *BEFORE* , *AFTER* , and *OVERLAP* relations. We list the statistics of annotations including events, timexes, relations between events-events, timexes-timexes, events-timexes in Table 5.4 for TimeBank, TBD3, THYME besides i2b2.

5.3 Method

In this section, we describe our approach in response to our third research question (*RQ3*) and the corresponding hypothesis (*H2*):

RQ3: How cross-sentence TRE can be improved?
H2: Incorporation of latent information (i.e., event arguments) between temporal

entities can improve TRE.

We consider extracting both temporal entities and their temporal relations in a pipeline manner. In particular, temporal entity extraction can be directly addressed by our model introduced in Chapter 3. In this chapter, we therefore focus on the temporal relation extraction, which assumes temporal entities are given. We model TRE by including both intra-sentence and inter-sentence temporal relations at once. Note that we consider both temporal entities and their temporal relations in a pipeline manner. the first part can be directly addressed by our model introduced in Chapter 3. We therefore focus on the temporal relation extraction with given entities in this chapter.

Linguistically, an event in general involves the corresponding subject and object, which are considered as arguments. Arguments shared among events are informative clues for TRE since they can convey co-reference and discourse information, which will be helpful for cross-sentence TRE. As illustrated in Figure 5.12, the subject and object of "challenge" co-refer to the subject and object of "fight", respectively. Co-reference between arguments is helpful in classifying the temporal relationship between them. To incorporate such information, we propose a neural TRE model that incorporates the arguments of events as additional context for TRE. Figure 5.13 describes our model architecture, which consists of four components: context encoder, entity encoder, pair encoder and classifier. The context encoder is used to inject context information to word representations, which are subsequently fed to the entity encoder. The entity encoder deals with entity representation, which is a concatenation of subject representation, object representation in combination with the corresponding word representation. The pair encoder processes each pair through combining three representations: a concatenation of two target entity representations, position embedding and a pair-context representation which sums the representation of the word and its dependency type along the shortest path, which connects two target entities. Our classifier takes in the pair representation and outputs the prediction. We detail each component in the next section.

5.3.1 Context Encoder

Given a sequence $S = w_1, w_2, \ldots, w_N$ containing N words, the context encoder takes in the raw sequence S and outputs word representations denoted by $\mathbf{w_1}$, $\mathbf{w_2}$, ..., $\mathbf{w_N}$. In detail, we apply a BiLSTM over word embeddings to generate

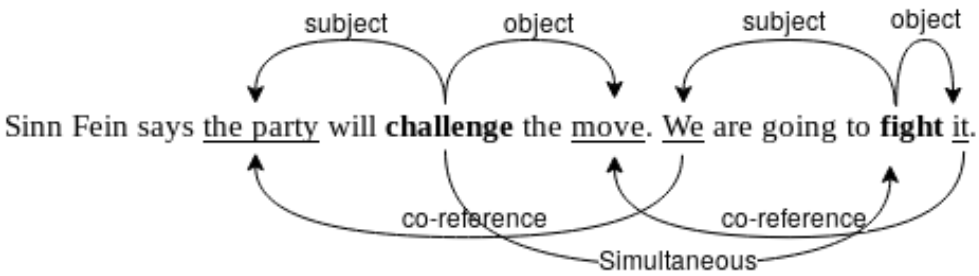

Figure 5.12: An example containing two events "challenge" and "fight" and their temporal relation "Simultaneous".

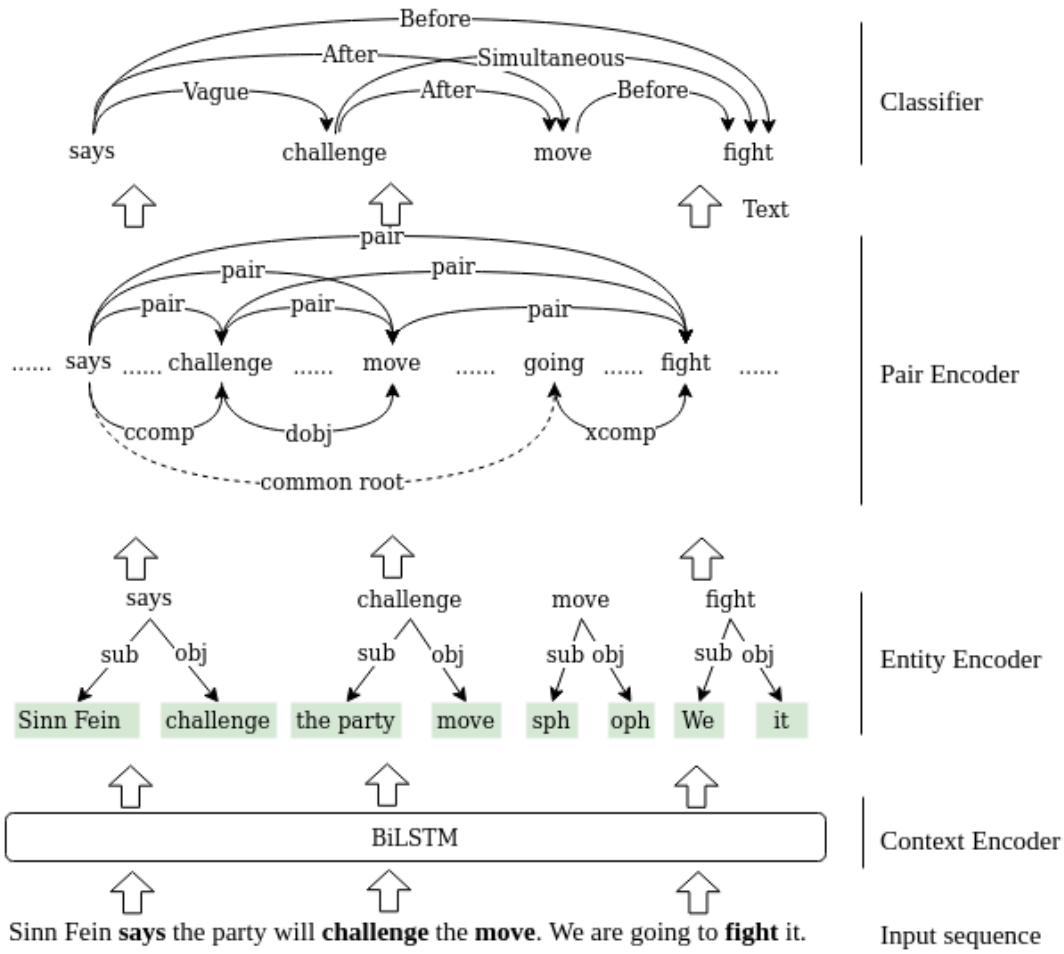

Figure 5.13: The model architecture of the temporal relation extraction. Dot arrows denote the dependencies. "sph" (subject placeholder) and "oph" (object placeholder) represent the placeholder for the absent subject and object, respectively. "ccomp", "dobj", "xcomp", "sub" and "obj" refer to dependency types. We use the common root assumption following Cheng and Miyao (2017) to represent the SDPs between cross-sentence pair candidates.

contextualized word representations. The word embeddings are a concatenation of character embeddings, ELMo embeddings (Peters et al., 2018) and Wikipedia word embeddings (Miwa and Bansal, 2016). The output of BiLSTM which stacks the forward and backward LSTM hidden states is used as our word representation. Note that the word representation encodes context information.

5.3.2 Entity Encoder

The entity encoder copes with entity representation, which is a concatenation of three parts: word representation, subject representation and object representation. Each part consists of a span representation and its semantic type embedding. The embeddings for entity types $\mathbf{et_i}$ and argument types $\mathbf{st_i}$, $\mathbf{at_i}$ are obtained from two embedding layers, respectively. In particular, entity types are included in the data while the argument types (i.e., a subset of dependency types) are obtained using the Stanford CoreNLP parser (Manning et al., 2014).

To produce the representation for the first part, we append the entity type embedding to an average of word representations using Equation 5.3, where *start* and *end* are the word indices in the sequence S.

$$\mathbf{e_i} = \frac{1}{end - start + 1} \sum_{i=start}^{end} \mathbf{w_i} \tag{5.3}$$

Similarly, we use Equation 5.3 to obtain the subject span representation, which is further combined with its type embedding to produce the subject representation. The object representation is calculated in the same way with the subject representation. When locating the arguments, we find that arguments corresponding to the events do not always occur in the same sentence. For example, in Figure 5.13, the subject for the event "move" is hidden in other distant sentences. In addition, some events have only subjects or objects. In response to the missing arguments, we define two place holders for the corresponding subject (sph) and object (oph), respectively. Unlike events, timexes do not have arguments. We also use the two place holders to represent their "arguments".

5.3.3 Pair Encoder

Our pair encoder uses the entity representation and additional context information to produce the representation for each pair. Each pair representation is a

concatenation of: pair of entity representations, position embedding, SDP embedding. We use a different embedding layer to calculate the position embedding.

Given the entity representation, the pair encoder firstly combines the target entity representation to form the pair representation, which is further concatenated with the position embedding following Miwa and Bansal (2016). To enrich the context for each pair, we append the SDP representation, which is a sum of word and its dependency type representations. To produce the SDP representation for each pair, we firstly collect the words on the SDP that connects the head words of two target entities. For each target entity, the word that has most dependency links is selected as the headword. Then we enrich word representation obtained from the context encoder by appending its dependency type representation. The enriched word representations are summed up to serve as the SDP representation. Before classification, we apply a linear layer to the output of pair encoder dimension reduction.

5.3.4 Classifier

Given pair representations, we employ softmax to classify the temporal relation for each pair. We consider both left-to-right and right-to-left directions, leading to two predictions to each pair. To resolve the conflicts, we choose the prediction that has a higher probability as the final prediction for the pair.

5.3.5 Training

We employ mini-batch training and update the model parameters using back-propagation through time (BPTT) (Werbos, 1990) with Adam (Kingma and Adam, 2015). The model parameters include weights, bias, and embeddings of entity types, positions, dependency types and place holders. During the training stage, we apply early stopping, L2-regularization and dropout (Hinton et al., 2012). Two dropouts are employed to the input of ELMo embeddings and input of the Softmax classifier, respectively. We fix the batch size and hidden units of LSTM with 1 and 2048, respectively. We initialise the embeddings of place holders as zeros. Hyper parameters including dropouts, Adam learning rate, gradient clipping and weight decay (L2) are all tuned using grid searching.

Hyper Parameters	Range	Best
First dropout	[0.3, 0.4, 0.5, 0.6, 0.7]	0.5
Second dropout	[0.3, 0.4, 0.5, 0.6, 0.7]	0.3
Learning rate	[0.001, 0.005, 0.01, 0.02]	0.001
Gradient clipping	[5, 10, 15, 20]	10
Weight decay (L2)	[10^{-8} – 10^{-3}]	10^{-4}

Table 5.5: Value range and best value of tuned hyper parameters in TBD3 corpus.

5.4 Evaluation

We evaluated our model on the TBD3 corpus, which is detailed in Section 5.2.4. We followed Meng and Rumshisky (2018) for data splitting, which divides the data into training, development and test sets with 22, 5 and 9 documents, respectively. In addition, we flipped each pair to augment pair candidates. If the temporal relation holding between two entities is *BEFORE*, then the relation type for the flipped pair is *AFTER*. As mentioned in Section 5.2.4, all pairs in the same and adjacent sentences have temporal relations. All events also have temporal relations with their corresponding DCT. To consider all the pairs including event-DCT and the rest at once, we append the sentence that contains the DCT to every two adjacent sentences as one instance. We enumerated all the combinations between any two entities contained in one instance to generate pair candidates. Tokenization and dependency parsing were conducted by Stanford CoreNLP parser (Manning et al., 2014).

Precision (P), recall (R) and F1-score (F1), defined in Section 2.5 in Chapter 2, were used as evaluation metrics in our task. Our model was implemented with Pytorch (Paszke et al., 2017). For LSTM, we initialized hidden states, cell states and all the biases as zero except for the forget gate that was set as 1. The other hyper parameters were chosen using the grid search. The details are listed in Table 5.5.

5.5 Results and Discussion

Table 5.6 presents the comparisons of our model with related work including the state-of-the-art model by Leeuwenberg and Moens (2018). Our model outperformed the state-of-the-art models with 58.9% in terms of micro F1-score,

Work	**F1-score (%)**			
	E-D [14%]	**E-E [64%]**	**E-T [19%]**	**Overall**
With gold temporal entities				
Leeuwenberg and Moens (2018)	-	-	-	58.6
Meng and Rumshisky (2018)	48.9	57	48.7	54.6
Cheng and Miyao (2017)	54.6	52.9	47.1	52.0
Mirza and Tonelli (2016)	53.4	51.9	46.8	51.1
Cassidy et al. (2014)	55.3	49.4	49.4	50.7
Ours	57.23	60.76	54.13	**58.9**
With predicted temporal entities				
Ours	51.51	55.30	48.44	**52.95**

Table 5.6: Comparisons of our model with state-of-the-art models on the test set of TBD3 Corpus. We use the model from Ju et al. (2018) to obtain predicted temporal entities.

achieving a new state-of-the-art for temporal relation extraction on TBD3 corpus.

Table 5.7 lists the intra- and inter-sentence performances of ablation tests on the development set of TBD3 corpus. As observed from Table 5.7, the integration of argument and SDP components improve the performances of cross-sentence relations. When removing SDPs, the performances for the intra-sentence relations keep the same while the inter-sentence performances drop. The reason for the same intra-sentence relations is that some events serve as the arguments, which can indicate the temporal order between them. For example, if event A is the object for event B, it is likely that event A happens after event B. This phenomenon is common in intra-sentence relations in the development set. We note that this phenomenon can be captured using SDPs since arguments constitute the corresponding SDPs. In other words, the information of arguments is overlapping with the information of SDPs. When removing SDPs, it supplements the context information for intra-sentence pairs where argument information is still available. However, it lowers the performances for inter-sentence relations since the connection between arguments, which are expressed by SDPs, is interrupted. In other words, we need SDPs to connect the arguments to convey their context information for cross-sentence pairs. When keeping the SDP component only, we got a slight drop in intra-sentence relations but higher performances in comparison with the model that integrates argument component only. Thus, we obtained better F1-score than the model without SDPs. When excluding all components, our model produced the lowest F1-score, demonstrating the effectiveness of SDP

Component	**F1-score (%)**		
	Intra-sentence	**Inter-sentence**	**Overall**
Our model	53.74	54.34	**54.16**
- SDP	53.74	50.75	51.63
- Argument	53.38	52.84	53.00
- SDP, Argument	51.94	50.08	51.11

Table 5.7: Ablation test of our model on the development set of TBD3 corpus.

Relation Type	**B**	**A**	**I**	**II**	**V**	**S**
B	**328**	10	18	2	209	0
A	4	**307**	6	9	122	2
I	10	19	**54**	0	63	1
II	13	10	0	**32**	68	2
V	148	106	37	20	**586**	2
S	3	6	1	1	24	**4**

Table 5.8: The statistics of category-wise predictions on the test set of TBD3 corpus. B, A, I, II, S, V represent BEFORE, AFTER, INCLUDES, IS_INCLUDED, SIMULTANEOUS, VAGUE, respectively. The first column represents the predictions while the first row represents the gold standard.

and argument components.

Table 5.8 reports the category-wise statistics including the correct predictions and errors on the test set of TBD3 corpus. Our model performs best in identifying *BEFORE*, *AFTER* and *VAGUE* relations and yields the lowest performances in *SIMULTANEOUS* and *IS_INCLUDED* relations which have sparse training examples. For all the relation types, most errors were caused by the misclassification between *VAGUE* and *NON-VAGUE* (e.g., *BEFORE*, *AFTER*) relation types. There are mainly two reasons account for this. One reason is that 43% of events are ambiguous (Reimers et al., 2016) such as in terms of event lasting time. When annotating the temporal relation for a pair of events, the "VAGUE" category will be assigned to the pair if it meets the following cases (Chambers et al., 2014):

- the annotator looked at the pair and decided that multiple valid relations could apply

- the annotator looked at the pair and decided that no temporal relation exists,

- the annotator failed to look at the pair, so a single relation may exist.

Such "VAGUE" relations confuse our model in making correct predictions between *VAGUE* and *NON-VAGUE*. Another reason is that some event arguments are hidden in other sentences rather than the sentence that contains the corresponding events. When searching the event arguments, we only consider one sentence instead of multiple sentences, thus leading to the argument absence. As a result, the performance was decreased. To attach the hidden co-reference to the events, we will consider the information from the whole documents as future work. In addition, we will incorporate the co-reference between arguments in future to enrich the context information for pairs of events/timexes. To evaluate the generalization ability of the model in different domains, more data sets such as THYME will be used in our task.

5.6 Summary

In this chapter, we firstly explained the definitions and concepts for TIE. Then, we presented an overview of event extraction, timex extraction and their temporal relation extraction. In addition, the commonly used TIE data resources together with their statistics were described. Moreover, we proposed a neural model to extract both intra- and inter-sentence relations using arguments of events. Experimental results and analysis were detailed, showing the effectiveness of event arguments in TRE. In addition, we pointed out the possible improvements for the model, which will be detailed in the next chapter.

Chapter 6

Conclusions

In this chapter, we present the answers to our research questions listed in Section 1.2 in Chapter 1 and describe how they are accomplished. In addition, we present possible directions to advance our research.

6.1 Evaluation of Research Questions

In response to the first research question (*RQ1*), we established the first research object (*RO1*):

RQ1: What are the state-of-the-art methods in extracting both flat and nested entities? *RO1:* To conduct a comprehensive literature review including methods, resources and tools for NER.

To achieve the first research object, we walked through the history of NER, which dates back to the 1990s. At that time, work dealing with NER focused on the extraction of flat entities. The research for nested entities was not recognised until 2003. Since many methods of flat NER can be adapted to nested NER, we hence analysed literature for both flat and nested NER in Section 2.2 and 2.3 of Chapter 2. In addition, we investigated available resources and tools for NER in Section 2.4, which enabled us to select ACE2005 (Walker et al., 2006), GENIA (Kim et al., 2003) and JNLPBA (Kim et al., 2004a) as our experimental data sets. Section 2.5 of Chapter 2 summarised evaluation metrics for NER, of which the precision, recall and F1-score were chosen to evaluate our model.

By reviewing the past methods for NER, we found that existing work ignored the dependencies between nested entities, which are informative clues to detect nested entities. To answer our second research question (RQ2), we proposed a neural layered model that automatically stacks flat NER layers (i.e., BiLSTM-CRF) to cope with nested entities. We presented our nested NER method in Chapter 3, which starts with an overview of the model architecture, and is followed by model explanations. Besides, we provided three ways to prepare the input of the model, resulting in three model variants, which were described in Section 3.5. Model training was detailed in Section 3.6.

RQ2: How nested NER can be improved using NNs?
RO2: To design neural NER models without feature-engineering and external knowledge bases.

To answer the third research question (RQ3), we came up with the following research objective (RO3) and presented the corresponding work in Chapter 4.

RQ3: How to measure the model generalisation?
RO3: To conduct evaluations in the settings of different domains and task-specific applications.

Section 4.1 presents the domain-specific evaluation with both flat and nested NER settings. Specifically, the model was firstly evaluated on a flat annotation corpus JNLPBA, showing that the model is able to achieve comparable performance with state-of-the-art flat NER models. Our model was then evaluated on nested corpora in both general and biomedical domains. Experimental results from these corpora demonstrate that our model can be generalised to different domains. Section 4.2 presents the task-specific evaluation with flat and nested NER settings, where the model was used as one of the components in each task. In detail, we conducted NER in the neuroscience domain under flat NER setting to help curation of neuroscience entities. For nested NER setting, we evaluated the model by identifying pertinent and potentially complex information about chronic obstructive pulmonary disease phenotypes from clinical textual data. In addition, the model was used for extracting adverse drug event and medication information from clinical records. We also adapted the model to extract elements

of patient/population, intervention, comparator, and outcomes, which were further used in the scientific abstract screening task. Overall results of these tasks showed that the model in general can be applied to different task-specific NLP applications. However, we note that the model improved the overall performances of medication related entities but did not improve the extraction of entities with adverse drug event and reason categories, which remain challenging in the clinical domain. As a whole, Chapter 4 directly answers RQ3.

Time is one crucial dimension in interpreting texts, which requires temporally relating entities. To investigate temporal information extraction, we decomposed our fourth research question into three subquestions. To answer the first subquestion (RQ4-1), we established our fourth research objective (RO4):

RQ4: How to temporally relate named entities?
RQ4-1: What are the state-of-the-art methods for TRE?
RO4: To conduct a comprehensive literature review focusing on temporal entity and their relation extraction.

As an initial step to achieve RO4, we firstly gave a thorough literature review in Section 5.2, which includes the extraction of temporal entities and their temporal relations. In addition, we summarised the commonly used corpora in temporal information extraction, which enables us to select TimeBank-Dense (Chambers et al., 2014) corpus as experimental data set.

To answer the second research subquestion (RQ4-2) of RQ4, we established the fifth objective (RO5). To achieve RO5, we proposed a neural model that incorporates event arguments to improve the extraction of both intra- and inter-sentence relations. The details of the proposed model were discussed in Section 5.3. Moreover, we conducted evaluations on TimeBank-Dense corpus to answer the third research subquestion (RQ4-3), which was established for the sixth research objective (RO6). Experimental results and analysis from Section 5.4 showed the effectiveness of event arguments in extracting temporal relations.

RQ4-2: How TRE can be improved using NNs?
RO5: To design a novel model to extract temporal relations including both intra- and inter-sentence relations.

RQ4-3: How to measure the ability of the TRE model in extracting both sentence- and document-level temporal relations?
RO6: To conduct evaluations in the settings of both intra- and inter-sentence temporal relations.

After accomplishing our research objectives, we obtained the findings summarised above to support the research hypotheses that we formulated in the beginning:

H1: Utilisation of inner entities can improve the detection of outer entities using NNs.
H2: Incorporation of latent information (i.e., event arguments) between temporal entities can improve TRE.

6.2 Future Work

Our research presented in the thesis can be enhanced in the following directions:

- Extension to detect discontinuous and overlapping entities. In addition to nested entities, discontinuous and overlapping entities (entities that partially overlap with each other) are also common in many domains. The model is designed to identify nested entities by utilising inner entities to improve the extraction of outer entities. For nested entities, the BIO tagging scheme is used for labeling entities on different nesting levels. To enable the extraction of discontinuous and overlapping entities, we can additionally introduce BD, ID, BH and IH labels to represent the **B**eginning of **D**iscontinuous body, **I**nside of **D**iscontinuous body, **B**eginning of **H**ead, and **I**nside of **I**nside of **H**ead (Metke-Jimenez and Karimi, 2015). With extended BIO tagging scheme, the model is able to extract flat, nested, discontinuous and overlapping entities.

- Integration of external resources for NER. Although external resources including knowledge bases and additional corpora are excluded in our neural NER model, incorporation of such resources will further improve our neural model, especially in the domains where annotated data are limited.

One promising way is the utilisation of un-annotated data resources. to augment the informative features which are extracted by the model, thus helping NER performances. Another promising way is to make the most of the neural models which are trained in other domains where annotated data are available, thus enabling the model to do NER in the domains where data resource is expensive.

- Multi-task learning for TRE. As TRE involves reasoning related causality and event co-reference, we can jointly model temporal relation and causality extraction, aiming to improve both tasks. Since some events are mentioned multiple times, therefore, finding the co-referred events is helpful to decide the temporal relations. Based on this observation, we can model TRE and co-reference resolution together to improve both tasks.

- Incorporation of external temporally annotated corpora. Existing available temporal corpora are limited for cross-sentence temporal relations. To the best of our knowledge, TimeBank-Dense corpus is the only available data set in the general domain that contains densely cross-sentence temporal relations. However, TimeBank-Dense contains only 36 news articles which are far from enough to train a good neural model. Therefore, the integration of other temporal resources such as knowledge bases and data resources will be helpful to better model the task of extracting temporal relations.

Bibliography

Global strategy for prevention, diagnosis and management of copd 2018 report. Technical report. Accessed 23 October 2018.

Aberdeen, J., Burger, J., Day, D., Hirschman, L., Robinson, P., and Vilain, M. (1995). Mitre: Description of the alembic system used for muc-6. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, pages 141–155, Stroudsburg, PA, USA. Association for Computational Linguistics.

Abney, S. (2002). Bootstrapping. In *Proceedings of the 40th annual meeting of the association for computational linguistics*.

Afzal, N., Mallipeddi, V. P., Sohn, S., Liu, H., Chaudhry, R., Scott, C. G., Kullo, I. J., and Arruda-Olson, A. M. (2018). Natural language processing of clinical notes for identification of critical limb ischemia. *International journal of medical informatics*, 111:83–89.

Ahmed, I. and Sathyaraj, R. (2015). Named entity recognition by using maximum entropy. *International Journal of Database Theory and Application*, 8(2):43–50.

Ahn, D., Fissaha Adafre, S., and De Rijke, M. (2005). Towards task-based temporal extraction and recognition. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Alex, B., Haddow, B., and Grover, C. (2007). Recognising nested named entities in biomedical text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 65–72, Stroudsburg,

PA, USA. Association for Computational Linguistics, Association for Computational Linguistics.

Allen, I. E. and Olkin, I. (1999). Estimating time to conduct a meta-analysis from number of citations retrieved. *Jama*, 282(7):634–635.

Allen, J. F. (1984). Towards a general theory of action and time. *Artificial intelligence*, 23(2):123–154.

Allen, J. F. and Ferguson, G. (1994). Actions and events in interval temporal logic. *Journal of logic and computation*, 4(5):531–579.

Allen, J. F. and Hayes, P. J. (1989). Moments and points in an interval-based temporal logic. *Computational Intelligence*, 5(3):225–238.

Alnazzawi, N., Thompson, P., Batista-Navarro, R., and Ananiadou, S. (2015). Using text mining techniques to extract phenotypic information from the phenochf corpus. In *BMC medical informatics and decision making*, volume 15, page S3. BioMed Central.

Ambert, K. H., Cohen, A. M., Burns, G. A., Boudreau, E., and Sonmez, K. (2013). Virk: an active learning-based system for bootstrapping knowledge base development in the neurosciences. *Frontiers in neuroinformatics*, 7:38.

Aphinyanaphongs, Y. and Aliferis, C. F. (2003). Text categorization models for retrieval of high quality articles in internal medicine. In *AMIA Annual Symposium Proceedings*, volume 2003, pages 31–35.

Aphinyanaphongs, Y., Tsamardinos, I., Statnikov, A., Hardin, D., and Aliferis, C. F. (2005). Text categorization models for high-quality article retrieval in internal medicine. *J Am Med Inform Assoc*, 12(2):207–216.

Aronson, A. R., Mork, J. G., Gay, C. W., Humphrey, S. M., and Rogers, W. J. (2004). The NLM indexing initiative's medical text indexer. 107:268–272.

Beaulieu-Jones, B. K., Greene, C. S., et al. (2016). Semi-supervised learning of the electronic health record for phenotype stratification. *Journal of biomedical informatics*, 64:168–178.

Bekhuis, T. and Demner-Fushman, D. (2010). Towards automating the initial screening phase of a systematic review. In Safran, C., Reti, S., and Marin, H., editors, *World Congress on Medical Informatics (MEDINFO)*, volume 160 of *Stud Health Technol Inform*, pages 146–150.

Bekhuis, T. and Demner-Fushman, D. (2012). Screening nonrandomized studies for medical systematic reviews: a comparative study of classifiers. *Artif Intell Med*, 55(3):197–207.

Bekhuis, T., Tseytlin, E., Mitchell, K. J., and Demner-Fushman, D. (2014). Feature engineering and a proposed decision-support system for systematic reviewers of medical evidence. *PLOS ONE*, 9(1):e86277.

Beller, E., Clark, J., Tsafnat, G., Adams, C., Diehl, H., Lund, H., Ouzzani, M., Thayer, K., Thomas, J., Turner, T., et al. (2018). Making progress with the automation of systematic reviews: principles of the international collaboration for the automation of systematic reviews (ICASR). *Systematic reviews*, 7(1):77.

Belousov, M., Milosevic, N., Dixon, W. G., and Nenadic, G. (2017). Extracting adverse drug reactions and their context using sequence labelling ensembles in tac2017. In *Proceedings of Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA.

Bender, O., Och, F. J., and Ney, H. (2003). Maximum entropy models for named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 148–151, Stroudsburg, PA, USA. Association for Computational Linguistics.

Benikova, D., Biemann, C., Kisselew, M., and Pado, S. (2014). Germeval 2014 named entity recognition shared task: companion paper.

Berger, A. L., Pietra, V. J. D., and Pietra, S. A. D. (1996). A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Bethard, S. (2013). ClearTK-TimeML: A minimalist approach to TempEval 2013. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 10–14, Atlanta, Georgia, USA. Association for Computational Linguistics.

Bethard, S., Derczynski, L., Savova, G., Pustejovsky, J., and Verhagen, M. (2015). SemEval-2015 task 6: Clinical TempEval. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 806–814, Denver, Colorado. Association for Computational Linguistics.

Bethard, S. and Martin, J. H. (2007). CU-TMP: Temporal relation classification using syntactic and semantic features. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 129–132, Prague, Czech Republic. Association for Computational Linguistics.

Bethard, S. and Martin, J. H. (2008). Learning semantic links from a corpus of parallel temporal and causal relations. In *Proceedings of ACL-08: HLT, Short Papers*, pages 177–180, Columbus, Ohio. Association for Computational Linguistics.

Bethard, S., Savova, G., Chen, W.-T., Derczynski, L., Pustejovsky, J., and Verhagen, M. (2016). SemEval-2016 task 12: Clinical TempEval. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1052–1062, San Diego, California. Association for Computational Linguistics.

Bethard, S., Savova, G., Palmer, M., and Pustejovsky, J. (2017). SemEval-2017 task 12: Clinical TempEval. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 565–572, Vancouver, Canada. Association for Computational Linguistics.

Bing, L., Chaudhari, S., Wang, R., and Cohen, W. (2015). Improving distant supervision for information extraction using label propagation through lists. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 524–529.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *J Mach Learn Res*, 3:993–1022.

Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl_1):D267–D270.

Boguraev, B., Castano, J., Gaizauskas, R., Ingria, B., Katz, G., Knippen, B., Littman, J., Mani, I., Pustejovsky, J., Sanfilippo, A., et al. (2005). TimeML 1.2.1–a formal specification language for events and temporal expressions. *Rapport technique*, 1(1).

Borthwick, A. and Grishman, R. (1999). *A maximum entropy approach to named entity recognition*. PhD thesis, Citeseer.

Borthwick, A., Sterling, J., Agichtein, E., and Grishman, R. (1998). Nyu: Description of the mene named entity system as used in muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*.

Bottou, L. and Cun, Y. L. (2004). Large scale online learning. In *Advances in neural information processing systems*, pages 217–224.

Boudin, F., Nie, J.-Y., Bartlett, J. C., Grad, R., Pluye, P., and Dawes, M. (2010a). Combining classifiers for robust PICO element detection. *BMC Medical Informatics and Decision Making*, 10(1):29.

Boudin, F., Shi, L., and Nie, J.-Y. (2010b). Improving medical information retrieval with pico element detection. In *European Conference on Information Retrieval*, pages 50–61. Springer.

Boyer, R. S. and Moore, J. S. (1991). Mjrtya fast majority vote algorithm. In *Automated Reasoning*, pages 105–117. Springer.

Brbić, M., Piškorec, M., Vidulin, V., Kriško, A., Šmuc, T., and Supek, F. (2017). Phenotype inference from text and genomic data. In Altun, Y., Das, K., Mielikäinen, T., Malerba, D., Stefanowski, J., Read, J., Žitnik, M., Ceci, M., and Džeroski, S., editors, *Machine Learning and Knowledge Discovery in Databases*, pages 373–377, Cham. Springer International Publishing.

Breitenstein, M. K., Liu, H., Maxwell, K. N., Pathak, J., and Zhang, R. (2018). Electronic health record phenotypes for precision medicine: Perspectives and caveats from treatment of breast cancer at a single institution. *Clinical and translational science*, 11(1):85–92.

Brockmeier, A. J., Ju, M., Przybyła, P., and Ananiadou, S. (2019). Improving reference prioritisation with pico recognition. *BMC Medical Informatics and Decision Making*, 19(1):256.

Bronckart, J.-P. and Sinclair, H. (1973). Time, tense and aspect. *Cognition*, 2(1):107–130.

Bui, D. D. A., Del Fiol, G., Hurdle, J. F., and Jonnalagadda, S. (2016). Extractive text summarization system to aid data extraction from full text in systematic review development. *Journal of Biomedical Informatics*, 64:265–272.

Bundschus, M., Dejori, M., Stetter, M., Tresp, V., and Kriegel, H.-P. (2008). Extraction of semantic biomedical relations from text using conditional random fields. *BMC bioinformatics*, 9(1):207.

Byrne, K. (2007). Nested named entity recognition in historical archive text. In *ICSC*, pages 589–596. IEEE Computer Society.

Cai, J. and Strube, M. (2010). End-to-end coreference resolution via hypergraph partitioning. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 143–151, Beijing, China. Coling 2010 Organizing Committee.

Cao, P., Chen, Y., Liu, K., Zhao, J., and Liu, S. (2018). Adversarial transfer learning for Chinese named entity recognition with self-attention mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 182–192, Brussels, Belgium. Association for Computational Linguistics.

Carroll, R. J., Eyler, A. E., and Denny, J. C. (2011). Naïve electronic health record phenotype identification for rheumatoid arthritis. In *AMIA annual symposium proceedings*, volume 2011, page 189. American Medical Informatics Association.

Caselli, T., Fokkens, A., Morante, R., and Vossen, P. (2015). SPINOZA_VU: An NLP pipeline for cross document TimeLines. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 787–791, Denver, Colorado. Association for Computational Linguistics.

Cassidy, T., McDowell, B., Chambers, N., and Bethard, S. (2014). An annotation framework for dense event ordering. In *Proceedings of the 52nd Annual*

Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 501–506, Baltimore, Maryland. Association for Computational Linguistics.

Cazzola, M., Calzetta, L., Rogliani, P., and Matera, M. G. (2017). The challenges of precision medicine in copd. *Molecular diagnosis & therapy*, 21(4):345–355.

Chambers, N. (2013). NavyTime: Event and time ordering from raw text. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 73–77, Atlanta, Georgia, USA. Association for Computational Linguistics.

Chambers, N., Cassidy, T., McDowell, B., and Bethard, S. (2014). Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics*, 2:273–284.

Chambers, N., Wang, S., and Jurafsky, D. (2007). Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 173–176, Prague, Czech Republic. Association for Computational Linguistics.

Chang, A. and Manning, C. D. (2013). SUTime: Evaluation in TempEval-3. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 78–82, Atlanta, Georgia, USA. Association for Computational Linguistics.

Changpinyo, S., Hu, H., and Sha, F. (2018). Multi-task learning for sequence tagging: An empirical study. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2965–2977, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Che, Z., Kale, D., Li, W., Bahadori, M. T., and Liu, Y. (2015). Deep computational phenotyping. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 507–516. ACM.

Chen, Q., Peng, Y., and Lu, Z. (2018). Biosentvec: creating sentence embeddings for biomedical texts. *arXiv preprint arXiv:1810.09302*.

Cheng, F. and Miyao, Y. (2017). Classifying temporal relations by bidirectional LSTM over dependency paths. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–6, Vancouver, Canada. Association for Computational Linguistics.

Chikka, V. R. (2016). CDE-IIITH at SemEval-2016 task 12: Extraction of temporal information from clinical documents using machine learning techniques. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1237–1240, San Diego, California. Association for Computational Linguistics.

Chiu, B., Crichton, G., Korhonen, A., and Pyysalo, S. (2016a). How to train good word embeddings for biomedical NLP. *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 166–174.

Chiu, B., Crichton, G., Korhonen, A., and Pyysalo, S. (2016b). How to train good word embeddings for biomedical nlp. In *Proceedings of the 15th workshop on biomedical natural language processing*, pages 166–174.

Chiu, J. P. and Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Cho, H., Okazaki, N., Miwa, M., and Tsujii, J. (2010). Nersuite: a named entity recognition toolkit. *Tsujii Laboratory, Department of Information Science, University of Tokyo, Tokyo, Japan.*

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Choi, S., Ryu, B., Yoo, S., and Choi, J. (2012). Combining relevancy and methodological quality into a single ranking for evidence-based medicine. *Inf Sci*, 214:76–90.

Christopoulou, F., Miwa, M., and Ananiadou, S. (2018). A walk-based model on entity graphs for relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 81–88, Melbourne, Australia. Association for Computational Linguistics.

Chung, G. Y. (2009a). Sentence retrieval for abstracts of randomized controlled trials. *BMC Medical Informatics and Decision Making*, 9(1):10.

Chung, G. Y. and Coiera, E. (2007). A study of structured clinical abstracts and the semantic classification of sentences. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 121–128. Association for Computational Linguistics.

Chung, G. Y.-C. (2009b). Towards identifying intervention arms in randomized controlled trials: extracting coordinating constructions. *Journal of Biomedical Informatics*, 42(5):790–800.

Cocos, A. and Masino, A. J. (2017). Combining rule-based and neural network systems for extracting adverse reactions from drug labels. In *Proceedings of Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA.

Cohan, A., Dernoncourt, F., Kim, D. S., Bui, T., Kim, S., Chang, W., and Goharian, N. (2018). A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Cohen, A. M. (2006). An effective general purpose approach for automated biomedical document classification. In *AMIA Annual Symposium Proceedings*, volume 2006, page 161.

Cohen, A. M. (2008). Optimizing feature representation for automated systematic review work prioritization. In *AMIA Annual Symposium Proceedings*, volume 2008, pages 121–125.

Cohen, A. M. (2011). Performance of support-vector-machine-based classification on 15 systematic review topics evaluated with the WSS@ 95 measure. *Journal of the American Medical Informatics Association: JAMIA*, 18(1):104.

Cohen, A. M., Ambert, K., and McDonagh, M. (2009). Cross-topic learning for work prioritization in systematic review creation and update. *J Am Med Inform Assoc*, 16(5):690–704.

Cohen, A. M., Ambert, K., and McDonagh, M. (2010). A prospective evaluation of an automated classification system to support evidence-based medicine and systematic review. In *AMIA Annual Symposium Proceedings*, volume 2010, page 121.

Cohen, A. M., Hersh, W. R., Peterson, K., and Yen, P. Y. (2006). Reducing workload in systematic review preparation using automated citation classification. *J Am Med Inform Assoc*, 13(2):206–219.

Cohen, A. M., Smalheiser, N. R., McDonagh, M. S., Yu, C., Adams, C. E., Davis, J. M., and Yu, P. S. (2015). Automated confidence ranked classification of randomized controlled trial articles: an aid to evidence-based medicine. *J Am Med Inform Assoc*, 22(3):707–717.

Collier, N., Groza, T., Smedley, D., Robinson, P. N., Oellrich, A., and Rebholz-Schuhmann, D. (2015a). Phenominer: from text to a database of phenotypes associated with omim diseases. *Database*, 2015.

Collier, N., Oellrich, A., and Groza, T. (2015b). Concept selection for phenotypes and diseases using learn to rank. In *J. Biomedical Semantics*.

Collier, N., Tran, M.-V., Le, H.-Q., Ha, Q.-T., Oellrich, A., and Rebholz-Schuhmann, D. (2013). Learning to recognize phenotype candidates in the auto-immune literature using svm re-ranking. In *PloS one*.

Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011a). Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011b). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Comrie, B. (1985). *Tense*, volume 17. Cambridge university press.

Consortium, L. D. et al. (2004). The ace 2004 evaluation plan. Technical report, Technical report, LDC.

Cortes, C. and Vapnik, V. (1995). Machine learning. *Support vector networks*, 20(3):25.

Cucchiarelli, A. and Velardi, P. (2001). Unsupervised named entity recognition using syntactic and semantic contextual evidence. *Computational Linguistics*, 27(1):123–131.

Culotta, A. and McCallum, A. (2005). Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751.

Dagan, I. and Engelson, S. P. (1995). Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier.

Dalal, S. R., Shekelle, P. G., Hempel, S., Newberry, S. J., Motala, A., and Shetty, K. D. (2013). A pilot study using machine learning and domain knowledge to facilitate comparative effectiveness review updating. *Med Decis Making*, 33(3):343–355.

Dalton, J., Dietz, L., and Allan, J. (2014). Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374. ACM.

Dandala, B., Mahajan, D., and Devarakonda, M. V. (2017). Ibm research system at tac 2017: Adverse drug reactions extraction from drug labels. In *Proceedings of Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA.

Dasgupta, S., Kalai, A. T., and Monteleoni, C. (2005). Analysis of perceptron-based active learning. In *International Conference on Computational Learning Theory*, pages 249–263. Springer.

Davidson, D. (1967). The logical form of action sentences.

De Bruijn, B., Carini, S., Kiritchenko, S., Martin, J., and Sim, I. (2008). Automated information extraction of key trial design elements from clinical trial publications. In *AMIA Annual Symposium Proceedings*, volume 2008, page 141. American Medical Informatics Association.

Del Fiol, G., Michelson, M., Iorio, A., Cotoi, C., and Haynes, R. B. (2018). A deep learning method to automatically identify reports of scientifically rigorous clinical research from the biomedical literature: Comparative analytic study. *Journal of Medical Internet Research*, 20(6).

Demner-Fushman, D. and Lin, J. (2005). Knowledge extraction for clinical question answering: Preliminary results. In *Proceedings of the AAAI-05 Workshop on Question Answering in Restricted Domains*, pages 9–13. AAAI Press (American Association for Artificial Intelligence) Pittsburgh, PA.

Demner-Fushman, D. and Lin, J. (2007). Answering clinical questions with knowledge-based and statistical techniques. *Computational Linguistics*, 33(1):63–103.

Dernoncourt, F., Lee, J. Y., and Szolovits, P. (2016). Neural networks for joint sentence classification in medical paper abstracts. *arXiv preprint arXiv:1612.05251*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dligach, D., Miller, T., Lin, C., Bethard, S., and Savova, G. (2017). Neural temporal relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 746–751, Valencia, Spain. Association for Computational Linguistics.

Dong, D., Wu, H., He, W., Yu, D., and Wang, H. (2015). Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Donmez, P. and Carbonell, J. G. (2008). Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 619–628. ACM.

Donmez, P. and Carbonell, J. G. (2010). From active to proactive learning methods. In *Advances in Machine Learning I*, pages 97–120. Springer.

Dowty, D. R. (1986). The effects of aspectual class on the temporal structure of discourse: semantics or pragmatics? *Linguistics and philosophy*, 9(1):37–61.

DSouza, J. and Ng, V. (2013). Classifying temporal relations in clinical data: a hybrid, knowledge-rich approach. *Journal of biomedical informatics*, 46:S29–S39.

Ebersole, F. B. (1952). Verb tenses as expressors and indicators. *Analysis*, 12(5):101–113.

Ekbal, A. and Bandyopadhyay, S. (2007). A hidden markov model based named entity recognition system: Bengali and hindi as case studies. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 545–552. Springer.

Ekbal, A. and Bandyopadhyay, S. (2010). Named entity recognition using support vector machine: A language independent approach. *International Journal of Electrical, Computer, and Systems Engineering*, 4(2):155–170.

Elsner, M., Charniak, E., and Johnson, M. (2009). Structured generative models for unsupervised named-entity clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 164–172. Association for Computational Linguistics.

Erbaugh, M. (1978). Acquisition of temporal and aspectual distinctions in mandarin.

Eriguchi, Y. and Kitani, T. (1996). Ntt data: Description of the erie system used for muc-6. In *Proceedings of a Workshop on Held at Vienna, Virginia: May 6-8, 1996*, TIPSTER '96, pages 469–470, Stroudsburg, PA, USA. Association for Computational Linguistics.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134.

Feng, X., Huang, L., Tang, D., Ji, H., Qin, B., and Liu, T. (2016). A language-independent neural network for event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 66–71, Berlin, Germany. Association for Computational Linguistics.

Filannino, M., Brown, G., and Nenadic, G. (2013). ManTIME: Temporal expression identification and normalization in the TempEval-3 challenge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 53–57, Atlanta, Georgia, USA. Association for Computational Linguistics.

Finkel, J., Dingare, S., Nguyen, H., Nissim, M., Manning, C., and Sinclair, G. (2004). Exploiting context for biomedical entity recognition: from syntax to the web. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*.

Finkel, J. R. and Manning, C. D. (2009). Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 141–150, Singapore. Association for Computational Linguistics, Association for Computational Linguistics.

Fisher, D., Soderland, S., Feng, F., and Lehnert, W. (1995). Description of the umass system as used for muc-6. In *Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, pages 127–140, Stroudsburg, PA, USA. Association for Computational Linguistics.

Florez, E., Precioso, F., Riveill, M., and Pighetti, R. (2018). Named entity

recognition using neural networks for clinical notes. In *International Workshop on Medication and Adverse Drug Event Detection*, pages 7–15.

Fragkou, P. (2017). Applying named entity recognition and co-reference resolution for segmenting english texts. *Progress in Artificial Intelligence*, 6(4):325–346.

French, L., Lane, S., Xu, L., and Pavlidis, P. (2009a). Automated recognition of brain region mentions in neuroscience literature. *Frontiers in neuroinformatics*, 3:29.

French, L., Lane, S., Xu, L., and Pavlidis, P. (2009b). Automated recognition of brain region mentions in neuroscience literature. *Frontiers in neuroinformatics*, 3:29.

French, L., Lane, S., Xu, L., and Pavlidis, P. (2009c). Automated recognition of brain region mentions in neuroscience literature. *Frontiers in neuroinformatics*, 3:29.

French, L., Lane, S., Xu, L., Siu, C., Kwok, C., Chen, Y., Krebs, C., and Pavlidis, P. (2012). Application and evaluation of automated methods to extract neuroanatomical connectivity statements from free text. *Bioinformatics*, 28(22):2963–2970.

French, L. and Pavlidis, P. (2012). Using text mining to link journal articles to neuroanatomical databases. *Journal of Comparative Neurology*, 520(8):1772–1783.

Friedman, C., Alderson, P. O., Austin, J. H., Cimino, J. J., and Johnson, S. B. (1994). A general natural-language text processor for clinical radiology. *Journal of the American Medical Informatics Association*, 1(2):161–174.

Friedman, C. and Hripcsak, G. (1998). Evaluating natural language processors in the clinical domain. *Methods of information in medicine*, 37(04/05):334–344.

Frunza, O., Inkpen, D., and Matwin, S. (2010). Building systematic reviews using automatic text classification techniques. In *International Conference on Computational Linguistics (COLING)*, pages 303–311.

Frunza, O., Inkpen, D., Matwin, S., Klement, W., and O'Blenis, P. (2011). Exploiting the systematic review protocol for classification of medical abstracts. *Artif Intell Med*, 51(1):17–25.

Fu, X., Batista-Navarro, R., Rak, R., and Ananiadou, S. (2015). Supporting the annotation of chronic obstructive pulmonary disease (copd) phenotypes with text mining workflows. *Journal of biomedical semantics*, 6(1):8.

Gaizauskas, R., Wakao, T., Humphreys, K., Cunningham, H., and Wilks, Y. (1995). University of sheffield: Description of the lasie systemas used for muc-6. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Garey, H. B. (1957). Verbal aspect in french. *Language*, 33(2):91–110.

Gehrmann, S., Dernoncourt, F., Li, Y., Carlson, E. T., Wu, J. T., Welt, J., Foote, Jr., J., Moseley, E. T., Grant, D. W., Tyler, P. D., and Celi, L. A. (2018). Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PLOS ONE*, 13(2):1–19.

Gkoutos, G. V., Schofield, P. N., and Hoehndorf, R. (2017). The anatomy of phenotype ontologies: principles, properties and applications. *Briefings in Bioinformatics*, 19(5):1008–1021.

Goller, C. and Kuchler, A. (1996). Learning task-dependent distributed representations by backpropagation through structure. In *Neural Networks, 1996., IEEE International Conference on*, volume 1, pages 347–352. IEEE.

Goyal, T. and Durrett, G. (2019). Embedding time expressions for deep temporal ordering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4400–4406, Florence, Italy. Association for Computational Linguistics.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Gridach, M. (2017). Character-level neural network for biomedical named entity recognition. *Journal of biomedical informatics*, 70:85–91.

Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: A brief history. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 1*, COLING '96, pages 466–471, Stroudsburg, PA, USA. Association for Computational Linguistics.

Grouin, C., Grabar, N., HAMON, T., ROSSET, S., TANNIER, X., and ZWEIGENBAUM, P. (2012). A tale of temporal relations between clinical concepts and temporal expressions: towards a representation of the clinical patients timeline. In *UZUNER, O., SUN, W. et RUMSHISKY, A., éditeurs: i2b2/VA Workshop Proc, Chicago, IL. i2b2.*

Grouin, C. and Moriceau, V. (2016). LIMSI at SemEval-2016 task 12: machine-learning and temporal information to identify clinical events and time expressions. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1225–1230, San Diego, California. Association for Computational Linguistics.

Groza, T., Köhler, S., Doelken, S., Collier, N., Oellrich, A., Smedley, D., Couto, F. M., Baynam, G., Zankl, A., and Robinson, P. N. (2015). Automatic concept recognition using the human phenotype ontology reference and test suite corpora. *Database*, 2015.

Gu, B. (2006). Recognizing nested named entities in GENIA corpus. In *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology*, LNLBioNLP '06, pages 112–113, New York, New York. Association for Computational Linguistics.

Gu, X., Ding, C., Li, S. K., and Xu, W. (2017). Bupt-pris system for tac 2017 event nugget detection, event argument linking and adr tracks. In *Proceedings of Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA.

GuoDong, Z. (2004). Recognizing names in biomedical texts using hidden markov model and svm plus sigmoid. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 1–7. Association for Computational Linguistics.

Gupta, N., Singh, S., and Roth, D. (2017). Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2671–2680, Copenhagen, Denmark. Association for Computational Linguistics.

Ha, E., Baikadi, A., Licata, C., and Lester, J. (2010). NCSU: Modeling temporal relations with Markov logic and lexical ontology. In *Proceedings of the*

5th International Workshop on Semantic Evaluation, pages 341–344, Uppsala, Sweden. Association for Computational Linguistics.

Hacioglu, K., Chen, Y., and Douglas, B. (2005). Automatic time expression labeling for english and chinese text. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 548–559. Springer.

Han, A. L.-F., Zeng, X., Wong, D. F., and Chao, L. S. (2015). Chinese named entity recognition with graph-based semi-supervised learning model. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 15–20.

Han, M. K., Agusti, A., Calverley, P. M., Celli, B. R., Criner, G., Curtis, J. L., Fabbri, L. M., Goldin, J. G., Jones, P. W., MacNee, W., et al. (2010). Chronic obstructive pulmonary disease phenotypes: the future of copd. *American journal of respiratory and critical care medicine*, 182(5):598–604.

Hansart, C., De Meyere, D., Watrin, P., Bittar, A., and Fairon, C. (2016). CENTAL at SemEval-2016 task 12: a linguistically fed CRF model for medical and temporal information extraction. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1286–1291, San Diego, California. Association for Computational Linguistics.

Hansen, M. J., Rasmussen, N. Ø., and Chung, G. (2008). A method of extracting the number of trial participants from abstracts describing randomized controlled trials. *Journal of Telemedicine and Telecare*, 14(7):354–358.

Hara, K. and Matsumoto, Y. (2007). Extracting clinical trial design information from medline abstracts. *New Generation Computing*, 25(3):263–275.

Hashimoto, K., Kontonatsios, G., Miwa, M., and Ananiadou, S. (2016). Topic detection using paragraph vectors to support active learning in systematic reviews. *J Biomed Inform*, 62:59–65.

Hazarika, D., Poria, S., Vij, P., Krishnamurthy, G., Cambria, E., and Zimmermann, R. (2018). Modeling inter-aspect dependencies for aspect-based sentiment analysis. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 266–270, New Orleans, Louisiana. Association for Computational Linguistics.

He, R., Lee, W. S., Ng, H. T., and Dahlmeier, D. (2017). An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397, Vancouver, Canada. Association for Computational Linguistics.

Heaney, L. G. and McGarvey, L. P. (2017). Personalised medicine for asthma and chronic obstructive pulmonary disease. *Respiration*, 93(3):153–161.

Higgins, J. P. and Deeks, J. J. (2011). Selecting studies and collecting data. In Higgins, J. P. and Green, S., editors, *Cochrane Handbook for Systematic Reviews of Interventions*, chapter 7. The Cochrane Collaboration. John Wiley & Sons. Version 5.1.0 [updated March 2011].

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

Hitzeman, J., Moens, M., and Grover, C. (1995). Algorithms for analysing the temporal structure of discourse. In *Seventh Conference of the European Chapter of the Association for Computational Linguistics*, Dublin, Ireland. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Howard, B. E., Phillips, J., Miller, K., Tandon, A., Mav, D., Shah, M. R., Holmgren, S., Pelch, K. E., Walker, V., Rooney, A. A., Macleod, M., Shah, R. R., and Thayer, K. (2016). SWIFT-Review: a text-mining workbench for systematic review. *Syst Rev*, 5(87).

Hsu, W., Speier, W., and Taira, R. K. (2012). Automated extraction of reported statistical analyses: towards a logical representation of clinical trial literature. In *AMIA Annual Symposium Proceedings*, volume 2012, page 350. American Medical Informatics Association.

Huang, C., Zaïane, O., Trabelsi, A., and Dziri, N. (2018). Automatic dialogue generation with expressed emotions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, Volume 2 (Short Papers), pages 49–54, New Orleans, Louisiana. Association for Computational Linguistics.

Huang, P.-Y., Huang, H.-H., Wang, Y.-W., Huang, C., and Chen, H.-H. (2017). NTU-1 at SemEval-2017 task 12: Detection and classification of temporal events in clinical data with domain adaptation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1010–1013, Vancouver, Canada. Association for Computational Linguistics.

Huang, X., Lin, J., and Demner-Fushman, D. (2006). Evaluation of PICO as a knowledge representation for clinical questions. In *AMIA annual symposium proceedings*, volume 2006, page 359. American Medical Informatics Association.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *ArXiv*, abs/1508.01991.

Ingria, R. and Pustejovsky, J. (2002). Timeml specification 1.0.

Iqbal, E., Mallah, R., Rhodes, D., Wu, H., Romero, A., Chang, N., Dzahini, O., Pandey, C., Broadbent, M., Stewart, R., et al. (2017). Adept, a semantically-enriched pipeline for extracting adverse drug events from free-text electronic health records. *PloS one*, 12(11):e0187121.

Jensen, K., Soguero-Ruiz, C., Mikalsen, K. O., Lindsetmo, R.-O., Kouskoumvekaki, I., Girolami, M., Skrovseth, S. O., and Augestad, K. M. (2017). Analysis of free text in electronic health records for identification of cancer patient trajectories. *Scientific reports*, 7:46226.

Ji, H., Nothman, J., Dang, H. T., and Hub, S. I. (2016). Overview of tac-kbp2016 tri-lingual edl and its impact on end-to-end cold-start kbp. *Proceedings of TAC*.

Ji, H., Nothman, J., Hachey, B., and Florian, R. (2015). Overview of TAC-KBP2015 tri-lingual entity discovery and linking. In *Proceedings of the 2015 Text Analysis Conference, TAC 2015, Gaithersburg, Maryland, USA, November 16-17, 2015, 2015*. NIST.

Ji, X., Ritter, A., and Yen, P.-Y. (2017). Using ontology-based semantic similarity to facilitate the article screening process for systematic reviews. *Journal of Biomedical Informatics*, 69:33–42.

Ji, X. and Yen, P.-Y. (2015). Using medline elemental similarity to assist in the article screening process for systematic reviews. *JMIR medical informatics*, 3(3).

Jiang, J. and Zhai, C. (2007). Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic. Association for Computational Linguistics.

Jin, D. and Szolovits, P. (2018). Pico element detection in medical text via long short-term memory neural networks. In *Proceedings of the BioNLP 2018 workshop*, pages 67–75.

Joachims, T., Finley, T., and Yu, C.-N. J. (2009). Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59.

Johnson, A. E., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035.

Jonnalagadda, S. and Petitti, D. (2013). A new iterative method to reduce workload in systematic review process. *Int J Comput Biol Drug Des*, 6(1-2):5–17.

Jonnalagadda, S. R., Goyal, P., and Huffman, M. D. (2015). Automating data extraction in systematic reviews: a systematic review. *Systematic reviews*, 4(1):78.

Ju, M., Miwa, M., and Ananiadou, S. (2018). A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459, New Orleans, Louisiana. Association for Computational Linguistics.

Ju, M., Nguyen, N. T. H., Miwa, M., and Ananiadou, S. (2019a). An ensemble of neural models for nested adverse drug events and medication extraction with subwords. *Journal of the American Medical Informatics Association*.

Ju, M., Short, A., Thompson, P., Diar Bakerly, N., Gkoutos, G., Tsaprouni, L., and Ananiadou, S. (2019b). Annotating and detecting phenotypic information

for chronic obstructive pulmonary disease. *Journal of the American Medical Informatics Association*.

Jung, H. and Stent, A. (2013). ATT1: Temporal annotation using big windows and rich syntactic and semantic features. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 20–24, Atlanta, Georgia, USA. Association for Computational Linguistics.

Kanoulas, E., Li, D., Azzopardi, L., and Spijker, R. (2017). CLEF 2017 technologically assisted reviews in empirical medicine overview. In *CEUR Workshop Proceedings*, volume 1866, pages 1–29.

Kanoulas, E., Li, D., Azzopardi, L., and Spijker, R. (2018). Clef 2018 technologically assisted reviews in empirical medicine overview. In *CEUR Workshop Proceedings*, volume 2125, pages 1–34.

Karystianis, G., Buchan, I., and Nenadic, G. (2014). Mining characteristics of epidemiological studies from medline: a case study in obesity. *Journal of biomedical semantics*, 5(1):22.

Karystianis, G., Thayer, K., Wolfe, M., and Tsafnat, G. (2017). Evaluation of a rule-based method for epidemiological document classification towards the automation of systematic reviews. *Journal of Biomedical Informatics*, 70:27–34.

Katiyar, A. and Cardie, C. (2018). Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871, New Orleans, Louisiana. Association for Computational Linguistics.

Kazama, J., Makino, T., Ohta, Y., and Tsujii, J. (2002). Tuning support vector machines for biomedical named entity recognition. In *Proceedings of the ACL-02 workshop on Natural language processing in the biomedical domain-Volume 3*, pages 1–8. Association for Computational Linguistics.

Kazama, J. and Torisawa, K. (2008). Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *proceedings of ACL-08: HLT*, pages 407–415.

Kelly, C. and Yang, H. (2013). A system for extracting study design parameters from nutritional genomics abstracts. *Journal of Integrative Bioinformatics*, 10(2):82–93.

Khabsa, M., Elmagarmid, A., Ilyas, I., Hammady, H., and Ouzzani, M. (2016). Learning to identify relevant studies for systematic reviews using random forest and external information. *Mach Learn*, 102(3):465–482.

Khalifa, A., Velupillai, S., and Meystre, S. (2016). UtahBMI at SemEval-2016 task 12: Extracting temporal information from clinical text. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1256–1262, San Diego, California. Association for Computational Linguistics.

Khordad, M., Mercer, R. E., and Rogan, P. (2011). Improving phenotype name recognition. In *Canadian Conference on Artificial Intelligence*, pages 246–257. Springer.

Khordad, M., Mercer, R. E., and Rogan, P. K. (2012). A machine learning approach for phenotype name recognition. In *COLING*.

Kim, J.-D., Ohta, T., Tateisi, Y., and Tsujii, J. (2003). GENIA corpusa semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl$_1$) : $i180 - -i182$.

Kim, J.-D., Ohta, T., Tsuruoka, Y., Tateisi, Y., and Collier, N. (2004a). Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Citeseer.

Kim, J.-D., Ohta, T., Tsuruoka, Y., Tateisi, Y., and Collier, N. (2004b). Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Association for Computational Linguistics.

Kim, S., Song, Y., Kim, K., Cha, J.-W., and Lee, G. G. (2006). Mmr-based active machine learning for bio named entity recognition. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 69–72. Association for Computational Linguistics.

Kim, S. N., Martinez, D., Cavedon, L., and Yencken, L. (2011). Automatic classification of sentences to support evidence based medicine. In *BMC Bioinformatics*, volume 12, page S5. BioMed Central.

Kingma, D. and Adam, J. B. (2015). A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Kiritchenko, S., de Bruijn, B., Carini, S., Martin, J., and Sim, I. (2010). ExaCT: automatic extraction of clinical trial characteristics from journal publications. *BMC Medical Informatics and Decision Making*, 10(1):56.

Klinger, R. and Friedrich, C. M. (2009). Feature subset selection in conditional random fields for named entity recognition. In *Proceedings of the International Conference RANLP-2009*, pages 185–191.

Kocbek, S. and Groza, T. (2017). Extracting disease-phenotype relations from text with disease-phenotype concept recognisers and association rule mining. In *2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)*, pages 358–363. IEEE.

Kolitsas, N., Ganea, O.-E., and Hofmann, T. (2018). End-to-end neural entity linking. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 519–529, Brussels, Belgium. Association for Computational Linguistics.

Kolomiyets, O. and Moens, M.-F. (2013). KUL: Data-driven approach to temporal parsing of newswire articles. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 83–87, Atlanta, Georgia, USA. Association for Computational Linguistics.

Kolya, A. K., Kundu, A., Gupta, R., Ekbal, A., and Bandyopadhyay, S. (2013). JU_CSE: A CRF based approach to annotation of temporal expression, event and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 64–72, Atlanta, Georgia, USA. Association for Computational Linguistics.

Korkontzelos, I., Nikfarjam, A., Shardlow, M., Sarker, A., Ananiadou, S., and Gonzalez, G. H. (2016). Analysis of the effect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts. *Journal of biomedical informatics*, 62:148–158.

Kozareva, Z. (2006). Bootstrapping named entity recognition with automatically generated gazetteer lists. In *Student Research Workshop*.

Kryściński, W., Paulus, R., Xiong, C., and Socher, R. (2018). Improving abstraction in text summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1808–1817, Brussels, Belgium. Association for Computational Linguistics.

Kumar Kolya, A., Ekbal, A., and Bandyopadhyay, S. (2010). JU_CSE_TEMP: A first step towards evaluating events, time expressions and temporal relations. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 345–350, Uppsala, Sweden. Association for Computational Linguistics.

Kuru, O., Can, O. A., and Yuret, D. (2016). CharNER: Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 911–921, Osaka, Japan. The COLING 2016 Organizing Committee.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 282–289.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Lamurias, A., Sousa, D., Pereira, S., Clarke, L., and Couto, F. M. (2017). ULISBOA at SemEval-2017 task 12: Extraction and classification of temporal expressions and events. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1019–1023, Vancouver, Canada. Association for Computational Linguistics.

Laokulrat, N., Miwa, M., and Tsuruoka, Y. (2014). Exploiting timegraphs in temporal relation classification. In *Proceedings of TextGraphs-9: the workshop on Graph-based Methods for Natural Language Processing*, pages 6–14.

Lascarides, A. and Asher, N. (1993). Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and Philosophy*, 16(5):437–493.

Laws, F. and Schätze, H. (2008). Stopping criteria for active learning of named entity recognition. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 465–472, Stroudsburg, PA, USA. Association for Computational Linguistics.

Le, P. and Titov, I. (2018). Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1595–1604, Melbourne, Australia. Association for Computational Linguistics.

LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer.

Lee, H.-J., Xu, H., Wang, J., Zhang, Y., Moon, S., Xu, J., and Wu, Y. (2016a). UTHealth at SemEval-2016 task 12: an end-to-end system for temporal information extraction from clinical notes. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1292–1297, San Diego, California. Association for Computational Linguistics.

Lee, J. Y., Dernoncourt, F., and Szolovits, P. (2017a). Transfer learning for named-entity recognition with neural networks. *arXiv preprint arXiv:1705.06273*.

Lee, K., He, L., Lewis, M., and Zettlemoyer, L. (2017b). End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, Copenhagen, Denmark. Association for Computational Linguistics.

Lee, S., Song, Y., Choi, M., and Kim, H. (2016b). Bagging-based active learning model for named entity recognition with distant supervision. In *2016 International Conference on Big Data and Smart Computing (BigComp)*, pages 321–324. IEEE.

Leeuwenberg, A. and Moens, M.-F. (2017). KULeuven-LIIR at SemEval-2017 task 12: Cross-domain temporal information extraction from clinical records. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1030–1034, Vancouver, Canada. Association for Computational Linguistics.

Leeuwenberg, A. and Moens, M.-F. (2018). Temporal information extraction by predicting relative time-lines. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1237–1246, Brussels, Belgium. Association for Computational Linguistics.

Lefebvre, C., Glanville, J., Wieland, L. S., Coles, B., and Weightman, A. L. (2013). Methodological developments in searching for studies for systematic reviews: past, present and future? *Syst Rev*, 2(78).

Li, F., Liu, W., and Yu, H. (2018a). Extraction of information related to adverse drug events from electronic health record notes: design of an end-to-end model based on deep learning. *JMIR medical informatics*, 6(4):e12159.

Li, F., Zhang, M., Tian, B., Chen, B., Fu, G., and Ji, D. (2017a). Recognizing irregular entities in biomedical text via deep neural networks. *Pattern Recognition Letters*, 105:105–113.

Li, J., Sun, A., Han, J., and Li, C. (2018b). A survey on deep learning for named entity recognition. *arXiv preprint arXiv:1812.09449*.

Li, M., Nguyen T. H., N., and Ananiadou, S. (2017b). Proactive learning for named entity recognition. In *BioNLP 2017*, pages 117–125, Vancouver, Canada,. Association for Computational Linguistics.

Li, P. and Huang, H. (2016). UTA DLNLP at SemEval-2016 task 12: Deep learning based natural language processing system for clinical information identification from clinical notes and pathology reports. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1268–1273, San Diego, California. Association for Computational Linguistics.

Liao, W. and Veeramachaneni, S. (2009). A simple semi-supervised algorithm for named entity recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, SemiSupLearn '09, pages 58–65, Stroudsburg, PA, USA. Association for Computational Linguistics.

Lin, B. Y., Xu, F., Luo, Z., and Zhu, K. (2017a). Multi-channel bilstm-crf model for emerging named entity recognition in social media. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 160–165.

Lin, C., Dligach, D., Miller, T. A., Bethard, S., and Savova, G. K. (2015). Multilayered temporal modeling for the clinical domain. *Journal of the American Medical Informatics Association*, 23(2):387–395.

Lin, C., Miller, T., Dligach, D., Bethard, S., and Savova, G. (2017b). Representations of time expressions for temporal relation extraction with convolutional neural networks. In *BioNLP 2017*, pages 322–327.

Lin, C., Miller, T., Dligach, D., Bethard, S., and Savova, G. (2019). A BERT-based universal model for both within- and cross-sentence clinical temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 65–71, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Lin, Y.-F., Tsai, T.-H., Chou, W.-C., Wu, K.-P., Sung, T.-Y., and Hsu, W.-L. (2004). A maximum entropy approach to biomedical named entity recognition. In *Proceedings of the 4th International Conference on Data Mining in Bioinformatics*, BIOKDD'04, pages 56–61, Berlin, Heidelberg. Springer-Verlag.

Ling, X. and Weld, D. S. (2010). Temporal information extraction. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Liu, F., Zhao, J., Lv, B., Xu, B., and Yu, H. (2005). Product named entity recognition based on hierarchical hidden Markov model. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Liu, L., Hu, X., Song, W., Fu, R., Liu, T., and Hu, G. (2018). Neural multitask learning for simile recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1543–1553, Brussels, Belgium. Association for Computational Linguistics.

Liu, L., Utiyama, M., Finch, A., and Sumita, E. (2016). Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan. The COLING 2016 Organizing Committee.

Liu, S., Wang, L., Chaudhary, V., and Liu, H. (2019). Attention neural model for temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 134–139, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Llorens, H., Chambers, N., UzZaman, N., Mostafazadeh, N., Allen, J., and Pustejovsky, J. (2015). SemEval-2015 task 5: QA TempEval - evaluating temporal information understanding with question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 792–800, Denver, Colorado. Association for Computational Linguistics.

Llorens, H., Saquete, E., and Navarro, B. (2010). Tipsem (english and spanish): Evaluating crfs and semantic roles in tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 284–291. Association for Computational Linguistics.

Lloyd, S. (1957). Least square quantization in pcm. bell telephone laboratories paper. published in journal much later: Lloyd, sp: Least squares quantization in pcm. *IEEE Trans. Inform. Theor.(1957/1982)*, 18.

Long, Y., Li, Z., Wang, X., and Li, C. (2017). XJNLP at SemEval-2017 task 12: Clinical temporal information ex-traction with a hybrid model. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1014–1018, Vancouver, Canada. Association for Computational Linguistics.

Lowell, D., Lipton, Z. C., and Wallace, B. C. (2018). How transferable are the datasets collected by active learners? *arXiv preprint arXiv:1807.04801*.

Lu, W. and Roth, D. (2015). Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867, Lisbon, Portugal. Association for Computational Linguistics.

Luan, Y., Wadden, D., He, L., Shah, A., Ostendorf, M., and Hajishirzi, H. (2019). A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Ma, X. and Hovy, E. H. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

MacAvaney, S., Cohan, A., and Goharian, N. (2017). GUIR at SemEval-2017 task 12: A framework for cross-domain clinical temporal information extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1024–1029, Vancouver, Canada. Association for Computational Linguistics.

Malaviya, C., Ferreira, P., and Martins, A. F. T. (2018). Sparse and constrained attention for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 370–376, Melbourne, Australia. Association for Computational Linguistics.

Malouf, R. (2002). Markov models for language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Manda, P., Beasley, L., and Mohanty, S. (2018). Taking a dive: Experiments in deep learning for automatic ontology-based annotation of scientific literature. *BioRxiv*, page 365874.

Mani, I. and Wilson, G. (2000). Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 69–76, Hong Kong. Association for Computational Linguistics.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60. Association for Computational Linguistics.

Mao, J., Moore, L. R., Blank, C. E., Wu, E. H.-H., Ackerman, M., Ranade, S., and Cui, H. (2016). Microbial phenomics information extractor (micropie): a natural language processing tool for the automated acquisition of prokaryotic phenotypic characters from text sources. *BMC bioinformatics*, 17(1):528.

Marciniak, M. and Mykowiecka, A. (2015). Nested term recognition driven by word connection strength. *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication*, 21(2):180–204.

Marinho, Z., Mendes, A., Miranda, S., and Nogueira, D. (2019). Hierarchical nested named entity recognition. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 28–34, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Màrquez, L., Villarejo, L., Martí, M. A., and Taulé, M. (2007). Semeval-2007 task 09: Multilevel semantic annotation of catalan and spanish. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 42–47, Stroudsburg, PA, USA. Association for Computational Linguistics.

Marsh, E. and Perzanowski, D. (1998). MUC-7 evaluation of IE technology: Overview of results. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, Fairfax, Virginia, USA. Association for Computational Linguistics.

Marshall, I. J., Kuiper, J., and Wallace, B. C. (2015a). Automating risk of bias assessment for clinical trials. *IEEE Journal of Biomedical and Health Informatics*, 19(4):1406–1412.

Marshall, I. J., Kuiper, J., and Wallace, B. C. (2015b). RobotReviewer: evaluation of a system for automatically assessing bias in clinical trials. *J Am Med Inform Assoc*, 23(1):193–201.

Marshall, I. J., Noel-Storr, A., Kuiper, J., Thomas, J., and Wallace, B. C. (2018). Machine learning for identifying randomized controlled trials: an evaluation and practitioner's guide. *Res Synth Methods*.

Matwin, S., Kouznetsov, A., Inkpen, D., Frunza, O., and O'Blenis, P. (2010). A new algorithm for reducing the workload of experts in performing systematic reviews. *J Am Med Inform Assoc*, 17(4):446–453.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics.

McCallum, A. K. (2002). MALLET: A machine learning for language toolkit.

McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133.

McDonald, R., Crammer, K., and Pereira, F. (2005). Flexible text segmentation with structured multilabel classification. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 987–994, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Mei, H., Bansal, M., and Walter, M. R. (2017). Coherent dialogue with attention-based language models. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Meng, Y. and Rumshisky, A. (2018). Context-aware neural model for temporal information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 527–536, Melbourne, Australia. Association for Computational Linguistics.

Meng, Y., Rumshisky, A., and Romanov, A. (2017). Temporal information extraction for question answering using syntactic dependencies in an LSTM-based architecture. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 887–896, Copenhagen, Denmark. Association for Computational Linguistics.

Metke-Jimenez, A. and Karimi, S. (2015). Concept extraction to identify adverse drug reactions in medical forums: A comparison of algorithms. *ArXiv*, abs/1504.06936.

Mikheev, A., Grover, C., and Moens, M. (1998). Description of the ltg system used for muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*.

Mikheev, A., Moens, M., and Grover, C. (1999). Named entity recognition without gazetteers. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 1–8. Association for Computational Linguistics.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. (2018). Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Millard, L. A., Flach, P. A., and Higgins, J. P. (2015). Machine learning to assist risk-of-bias assessments in systematic reviews. *Int J Epidemiol*, 45(1):266–277.

Miller, S., Crystal, M., Fox, H., Ramshaw, L., Schwartz, R., Stone, R., Weischedel, R., et al. (1998). Bbn: Description of the sift system as used for muc-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*.

Minard, A.-L., Speranza, M., Agirre, E., Aldabe, I., van Erp, M., Magnini, B., Rigau, G., and Urizar, R. (2015). SemEval-2015 task 4: TimeLine: Cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 778–786, Denver, Colorado. Association for Computational Linguistics.

Miravitlles, M., Calle, M., and Soler-Cataluna, J. J. (2012). Clinical phenotypes of copd: identification, definition and implications for guidelines. *Archivos de Bronconeumología (English Edition)*, 48(3):86–98.

Miravitlles, M., Soler-Cataluña, J. J., Calle, M., and Soriano, J. B. (2013). Treatment of copd by clinical phenotypes: putting old evidence into clinical practice. *European Respiratory Journal*, 41(6):1252–1256.

Mirza, P. (2014). Extracting temporal and causal relations between events. In *Proceedings of the ACL 2014 Student Research Workshop*, pages 10–17, Baltimore, Maryland, USA. Association for Computational Linguistics.

Mirza, P. and Tonelli, S. (2014). An analysis of causality between events and its relation to temporal information. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2097–2106, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Mirza, P. and Tonelli, S. (2016). CATENA: CAusal and TEmporal relation extraction from NAtural language texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 64–75, Osaka, Japan. The COLING 2016 Organizing Committee.

Mishra, S. and Diesner, J. (2016). Semi-supervised named entity recognition in noisy-text. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 203–212.

Miwa, M. and Bansal, M. (2016). End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics, Association for Computational Linguistics.

Miwa, M., Thomas, J., O'Mara-Eves, A., and Ananiadou, S. (2014). Reducing systematic review workload through certainty-based screening. *J Biomed Inform*, 51:242–253.

Moen, S. and Ananiadou, T. S. S. (2013). Distributional semantics resources for biomedical text processing. *Proceedings of LBM*, pages 39–44.

Mohit, B. and Hwa, R. (2005). Syntax-based semi-supervised named entity tagging. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 57–60. Association for Computational Linguistics.

Muis, A. O. and Lu, W. (2017). Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2608–2618, Copenhagen, Denmark. Association for Computational Linguistics.

Müller, H.-M., Rangarajan, A., Teal, T. K., and Sternberg, P. W. (2008). Textpresso for neuroscience: searching the full text of thousands of neuroscience research papers. *Neuroinformatics*, 6(3):195–204.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Naghavi, M., Abajobir, A. A., Abbafati, C., Abbas, K. M., Abd-Allah, F., Abera, S. F., Aboyans, V., Adetokunboh, O., Afshin, A., Agrawal, A., et al. (2017).

Global, regional, and national age-sex specific mortality for 264 causes of death, 1980–2016: a systematic analysis for the global burden of disease study 2016. *The Lancet*, 390(10100):1151–1210.

Navarro, B. and Saquete, E. (2015). GPLSIUA: Combining temporal information and topic modeling for cross-document event ordering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 820–824, Denver, Colorado. Association for Computational Linguistics.

Negri, M. and Marseglia, L. (2004). Recognition and normalization of time expressions: Itc-irst at tern 2004. *Rapport interne, ITC-irst, Trento*.

Ng, J.-P., Chen, Y., Kan, M.-Y., and Li, Z. (2014). Exploiting timelines to enhance multi-document summarization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 923–933, Baltimore, Maryland. Association for Computational Linguistics.

Nguyen, N. T. H., Gabud, R. S., and Ananiadou, S. (2019). Copious: A gold standard corpus of named entities towards extracting species occurrence from biodiversity literature. *Biodiversity Data Journal*, 7:e29626.

Ni, Y., Kennebeck, S., Dexheimer, J. W., McAneney, C. M., Tang, H., Lingren, T., Li, Q., Zhai, H., and Solti, I. (2014). Automated clinical trial eligibility prescreening: increasing the efficiency of patient identification for clinical trials in the emergency department. *Journal of the American Medical Informatics Association*, 22(1):166–178.

Nikfarjam, A., Sarker, A., OConnor, K., Ginn, R., and Gonzalez, G. (2015). Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, 22(3):671–681.

Ning, Q., Feng, Z., Wu, H., and Roth, D. (2018). Joint reasoning for temporal and causal relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2278–2288, Melbourne, Australia. Association for Computational Linguistics.

Niu, Y. and Hirst, G. (2004). Analysis of semantic classes in medical text for question answering. In *Proceedings of the Conference on Question Answering in Restricted Domains*.

Noreen, E. W. (1989). *Computer-intensive methods for testing hypotheses*. Wiley New York.

Noro, T., Inui, T., Takamura, H., and Okumura, M. (2006). Time period identification of events in text. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 1153–1160, Sydney, Australia. Association for Computational Linguistics.

Nye, B., Li, J. J., Patel, R., Yang, Y., Marshall, I. J., Nenkova, A., and Wallace, B. C. (2018). A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. pages 197–207.

Oellrich, A., Collier, N., Smedley, D., and Groza, T. (2015). Generation of silver standard concept annotations from biomedical texts with special relevance to phenotypes. *PloS one*, 10(1):e0116040.

O'Mara-Eves, A., Thomas, J., McNaught, J., Miwa, M., and Ananiadou, S. (2015). Using text mining for study identification in systematic reviews: a systematic review of current approaches. *Syst Rev*, 4(5).

O'Reilly, C., Iavarone, E., and Hill, S. L. (2017). A framework for collaborative curation of neuroscientific literature. *Frontiers in neuroinformatics*, 11:27.

Organization, W. H. et al. (1972). *International drug monitoring: the role of national centres, report of a WHO meeting [held in Geneva from 20 to 25 September 1971]*. World Health Organization.

Oxman, A. D., Sackett, D. L., Guyatt, G. H., Browman, G., Cook, D., Gerstein, H., Haynes, B., Hayward, R., Levine, M., Nishikawa, J., et al. (1993). Users' guides to the medical literature: I. how to get started. *JAMA*, 270(17):2093–2095.

Özkaya, S. and Diri, B. (2011). Named entity recognition by conditional random fields from turkish informal texts. In *2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU)*, pages 662–665. IEEE.

Pacific Northwest Evidence-based Practice, OHSU Center for Evidence-Based Policy. Drug effectiveness review project (derp) systematic drug class review gold standard data. http://skynet.ohsu.edu/~cohenaa/systematic-drug-class-review-data.html.

Pan, S. J., Toh, Z., and Su, J. (2013). Transfer joint embedding for cross-domain named entity recognition. *ACM Transactions on Information Systems (TOIS)*, 31(2):7.

Pasunuru, R. and Bansal, M. (2019). Dstc7-avsd: Scene-aware video-dialogue systems with dual attention. In *DSTC7 at AAAI2019 workshop*.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

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

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Piñero, J., Bravo, À., Queralt-Rosinach, N., Gutiérrez-Sacristán, A., Deu-Pons, J., Centeno, E., García-García, J., Sanz, F., and Furlong, L. I. (2016). Disgenet: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic acids research*, page gkw943.

Przybyła, P., Brockmeier, A. J., Kontonatsios, G., Le Pogam, M.-A., McNaught, J., von Elm, E., Nolan, K., and Ananiadou, S. (2018). Prioritising references for systematic reviews with robotanalyst: A user study. *Research synthesis methods*, 9(3):470–488.

Przybya, P., Brockmeier, A. J., and Ananiadou, S. (2019). Quantifying risk factors in medical reports with a context-aware linear model. *Journal of the American Medical Informatics Association*, 26(6):537–546.

Pustejovsky, J. (2002). Terqas: time and event recognition for question answering systems. In *ARDA Workshop*.

Pustejovsky, J., Hanks, P., Sauri, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., et al. (2003a). The timebank corpus. In *Corpus linguistics*, volume 2003, page 40. Lancaster, UK.

Pustejovsky, J., Ingria, B., Sauri, R., Castano, J., Littman, J., Gaizauskas, R., Setzer, A., Katz, G., and Mani, I. (2005). The specification language TimeML. *The language of time: A reader*, pages 545–557.

Pustejovsky, J., Lee, K., Bunt, H., and Romary, L. (2010). Iso-timeml: An international standard for semantic annotation. In *LREC*, volume 10, pages 394–397.

Pustejovsky, J., Mani, I., and Organizers (2003b). Timeml annotation graphical organizer.

Qu, L., Ferraro, G., Zhou, L., Hou, W., and Baldwin, T. (2016). Named entity recognition for novel types by transfer learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 899–905, Austin, Texas. Association for Computational Linguistics.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding with unsupervised learning. Technical report, Technical report, OpenAI.

Radford, W., Carreras, X., and Henderson, J. (2015). Named entity recognition with document-specific KB tag gazetteers. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 512–517, Lisbon, Portugal. Association for Computational Linguistics.

Rathbone, J., Hoffmann, T., and Glasziou, P. (2015). Faster title and abstract screening? Evaluating Abstrackr, a semi-automated online screening program for systematic reviewers. *Syst Rev*, 4(80).

Ratinov, L. and Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning*, pages 147–155. Association for Computational Linguistics.

Rei, M., Crichton, G., and Pyysalo, S. (2016). Attending to characters in neural sequence labeling models. In *Proceedings of COLING 2016, the 26th International*

Conference on Computational Linguistics: Technical Papers, pages 309–318, Osaka, Japan. The COLING 2016 Organizing Committee.

Reichenbach, H. (1947). The tenses of verbs. *Time: From Concept to Narrative Construct: a Reader*, pages 1–12.

Reimers, N., Dehghani, N., and Gurevych, I. (2016). Temporal anchoring of events for the TimeBank corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2195–2204, Berlin, Germany. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2017). Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *ArXiv*, abs/1707.06799.

Richardet, R., Chappelier, J.-C., Telefont, M., and Hill, S. (2015a). Large-scale extraction of brain connectivity from the neuroscientific literature. *Bioinformatics*, 31(10):1640–1647.

Richardet, R., Chappelier, J.-C., Telefont, M., and Hill, S. (2015b). Large-scale extraction of brain connectivity from the neuroscientific literature. *Bioinformatics*, 31(10):1640–1647.

Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine Learning*, 62(1):107–136.

Richardson, W. S., Wilson, M. C., Nishikawa, J., and Hayward, R. S. (1995). The well-built clinical question: a key to evidence-based decisions. *ACP Journal Club*, 123(3):A12–A12.

Riloff, E., Jones, R., et al. (1999). Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479.

Roberts, K. E., Demner-Fushman, D., and Tonning, J. M. (2017). Overview of the tac 2017 adverse reaction extraction from drug labels track. In *Proceedings of Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA.

Ruder, S., Ghaffari, P., and Breslin, J. G. (2016). A hierarchical model of reviews for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 999–1005, Austin, Texas. Association for Computational Linguistics.

Rush, A. M., Chopra, S., and Weston, J. (2015). A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Russell, S. and Norvig, P. (2010). Artificial intelligence: A modern approach. third edit.

Saeidi, M., Bouchard, G., Liakata, M., and Riedel, S. (2016). SentiHood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1546–1556, Osaka, Japan. The COLING 2016 Organizing Committee.

Salton, G. and Buckley, C. (1990). Improving retrieval performance by relevance feedback. *J Assoc Inf Sci Technol*, 41(4):288–297.

Sarath, P., Manikandan, R., and Niwa, Y. (2016). Hitachi at SemEval-2016 task 12: A hybrid approach for temporal information extraction from clinical notes. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1231–1236, San Diego, California. Association for Computational Linguistics.

Sarntivijai, S., Vasant, D., Jupp, S., Saunders, G., Bento, A. P., Gonzalez, D., Betts, J., Hasan, S., Koscielny, G., Dunham, I., et al. (2016). Linking rare and common disease: mapping clinical disease-phenotypes to ontologies in therapeutic target validation. *Journal of biomedical semantics*, 7(1):8.

Sasaki, Y., Tsuruoka, Y., McNaught, J., and Ananiadou, S. (2008). How to make the most of ne dictionaries in statistical ner. *BMC bioinformatics*, 9(11):S5.

Savova, G. K., Tseytlin, E., Finan, S., Castine, M., Miller, T., Medvedeva, O., Harris, D., Hochheiser, H., Lin, C., Chavan, G., et al. (2017). Deepphe: a natural language processing system for extracting cancer phenotypes from clinical records. *Cancer research*, 77(21):e115–e118.

Schardt, C., Adams, M. B., Owens, T., Keitz, S., and Fontelo, P. (2007). Utilization of the PICO framework to improve searching PubMed for clinical questions. *BMC Medical Informatics and Decision Making*, 7(1):16.

Scheffer, T., Decomain, C., and Wrobel, S. (2001). Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pages 309–318. Springer.

Schilder, F. and Habel, C. (2001). From temporal expressions to temporal information: Semantic tagging of news messages. In *Proceedings of the ACL 2001 workshop on temporal and spatial information processing*.

Segreti, A., Stirpe, E., Rogliani, P., and Cazzola, M. (2014). Defining phenotypes in copd: an aid to personalized healthcare. *Molecular diagnosis & therapy*, 18(4):381–388.

Sekine, S. and Isahara, H. (2000). Irex: Ir & ie evaluation project in japanese. In *LREC*, pages 1977–1980. Citeseer.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Settles, B. (2004). Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*.

Settles, B. (2009). Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Settles, B. and Craven, M. (2008). An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1070–1079. Association for Computational Linguistics.

Shardlow, M., Ju, M., Li, M., O'Reilly, C., Iavarone, E., McNaught, J., and Ananiadou, S. (2018). A text mining pipeline using active and deep learning aimed at curating information in computational neuroscience. *Neuroinformatics*.

Shemilt, I., Khan, N., Park, S., and Thomas, J. (2016). Use of cost-effectiveness analysis to compare the efficiency of study identification methods in systematic reviews. *Syst Rev*, 5(140).

Shemilt, I., Simon, A., Hollands, G. J., Marteau, T. M., Ogilvie, D., O'Mara-Eves, A., Kelly, M. P., and Thomas, J. (2014). Pinpointing needles in giant haystacks: use of text mining to reduce impractical screening workload in extremely large scoping reviews. *Res Synth Methods*, 5(1):31–49.

Shen, D., Zhang, J., Su, J., Zhou, G., and Tan, C.-L. (2004). Multi-criteria-based active learning for named entity recognition. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shen, D., Zhang, J., Zhou, G., Su, J., and Tan, C.-L. (2003). Effective adaptation of a hidden markov model-based named entity recognizer for biomedical domain. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine-Volume 13*, pages 49–56, Sapporo, Japan. Association for Computational Linguistics, Association for Computational Linguistics.

Shen, Y., Yun, H., Lipton, Z., Kronrod, Y., and Anandkumar, A. (2017). Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, Vancouver, Canada. Association for Computational Linguistics.

Sheng, E. and Natarajan, P. (2018). A byte-sized approach to named entity recognition. *arXiv preprint arXiv:1809.08386*.

Sim, I., Tu, S. W., Carini, S., Lehmann, H. P., Pollock, B. H., Peleg, M., and Wittkowski, K. M. (2014). The ontology of clinical research (OCRe): an informatics foundation for the science of clinical research. *Journal of biomedical informatics*, 52:78–91.

Singh, G., Marshall, I. J., Thomas, J., Shawe-Taylor, J., and Wallace, B. C. (2017). A neural candidate-selector architecture for automatic structured clinical text annotation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1519–1528. ACM.

Singh, T. D., Nongmeikapam, K., Ekbal, A., and Bandyopadhyay, S. (2009). Named entity recognition for manipuri using support vector machine. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 2*, pages 811–818.

Singhal, A., Simmons, M., and Lu, Z. (2016). Text mining genotype-phenotype relationships from biomedical literature for database curation and precision medicine. *PLoS computational biology*, 12(11):e1005017.

Small, K., Wallace, B., Trikalinos, T., and Brodley, C. E. (2011). The constrained weight space SVM: learning with ranked features. In *International Conference on Machine Learning (ICML-11)*, pages 865–872.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, pages 2951–2959, USA. Curran Associates Inc.

Søgaard, A. and Goldberg, Y. (2016). Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany. Association for Computational Linguistics.

Sohrab, M. G. and Miwa, M. (2018). Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

Song, L., Zhang, Y., Wang, Z., and Gildea, D. (2018). N-ary relation extraction using graph-state LSTM. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2226–2235, Brussels, Belgium. Association for Computational Linguistics.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Stevenson, M. and Gaizauskas, R. (2000). Using corpus-derived name lists for named entity recognition. In *Proceedings of the sixth conference on Applied natural language processing*, pages 290–295. Association for Computational Linguistics.

Stöckel, A., Paassen, B., Dickfelder, R., Göpfert, J. P., Brazda, N., Müller, H. W., Cimiano, P., Hartung, M., and Klinger, R. (2015). Scie: Information extraction for spinal cord injury preclinical experiments-a webservice and open source toolkit. *bioRxiv*, page 013458.

Stone, M. and Arora, R. (2017). Identifying nominals with no head match co-references using deep learning. *CoRR*, abs/1710.00936.

Stratos, K. (2017). Entity identification as multitasking. In *Proceedings of the 2nd Workshop on Structured Prediction for Natural Language Processing*, pages 7–11, Copenhagen, Denmark. Association for Computational Linguistics.

Strötgen, J. and Gertz, M. (2010). Heideltime: High quality rule-based extraction and normalization of temporal expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 321–324, Stroudsburg, PA, USA. Association for Computational Linguistics.

Strötgen, J. and Gertz, M. (2015). A baseline temporal tagger for all languages. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 541–547.

Strubell, E., Verga, P., Belanger, D., and McCallum, A. (2017). Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark. Association for Computational Linguistics.

Styler IV, W. F., Bethard, S., Finan, S., Palmer, M., Pradhan, S., de Groen, P. C., Erickson, B., Miller, T., Lin, C., Savova, G., et al. (2014a). Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

Styler IV, W. F., Bethard, S., Finan, S., Palmer, M., Pradhan, S., de Groen, P. C., Erickson, B., Miller, T., Lin, C., Savova, G., and Pustejovsky, J. (2014b). Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, 2:143–154.

Suleiman, D., Awajan, A., and Al-Madi, N. (2017). Deep learning based technique for plagiarism detection in arabic texts. In *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, pages 216–222. IEEE.

Summerscales, R., Argamon, S., Hupert, J., and Schwartz, A. (2009). Identifying treatments, groups, and outcomes in medical abstracts. In *The Sixth Midwest Computational Linguistics Colloquium (MCLC 2009)*.

Summerscales, R. L., Argamon, S., Bai, S., Hupert, J., and Schwartz, A. (2011). Automatic summarization of results from clinical trials. In *2011 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 372–377. IEEE.

Sun, W., Rumshisky, A., and Uzuner, O. (2013a). Annotating temporal information in clinical narratives. *Journal of biomedical informatics*, 46:S5–S12.

Sun, W., Rumshisky, A., and Uzuner, O. (2013b). Evaluating temporal relations in clinical text: 2012 i2b2 Challenge. *Journal of the American Medical Informatics Association*, 20(5):806–813.

Takeuchi, K. and Collier, N. (2002). Use of support vector machines in extended named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Takeuchi, K. and Collier, N. (2005). Bio-medical entity extraction using support vector machines. *Artificial Intelligence in Medicine*, 33(2):125–137.

Tang, B., Wu, Y., Jiang, M., Chen, Y., Denny, J. C., and Xu, H. (2013). A hybrid system for temporal information extraction from clinical text. *Journal of the American Medical Informatics Association*, 20(5):828–835.

Thomas, J., McNaught, J., and Ananiadou, S. (2011). Applications of text mining within systematic reviews. *Res Synth Methods*, 2(1):1–14.

Thompson, P. and Ananiadou, S. (2017). Extracting gene-disease relations from text to support biomarker discovery. In *Proceedings of the 2017 International Conference on Digital Health*, pages 180–189. ACM.

Tiftikci, M., Özgür, A., He, Y., and Hur, J. (2017). Extracting adverse drug reactions using deep learning and dictionary based approaches. In *Proceedings of Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA.

Timsina, P., Liu, J., and El-Gayar, O. (2016). Advanced analytics for the automation of medical systematic reviews. *Inf Syst Front*, 18(2):237–252.

Tjong Kim Sang, E. F. (2002). Introduction to the conll-2002 shared task: Language-independent named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–4, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tjong Kim Sang, E. F. and De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Tokui, S., Oono, K., Hido, S., and Clayton, J. (2015a). Chainer: a next-generation open source framework for deep learning. In *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, volume 5.

Tokui, S., Oono, K., Hido, S., and Clayton, J. (2015b). Chainer: a next-generation open source framework for deep learning. In *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, volume 5, pages 1–6.

Toral, A. and Munoz, R. (2006). A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. In *Proceedings of the Workshop on NEW TEXT Wikis and blogs and other dynamic text sources*.

Tourille, J., Ferret, O., Névéol, A., and Tannier, X. (2017a). Neural architecture for temporal relation extraction: A bi-LSTM approach for detecting narrative containers. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 224–230, Vancouver, Canada. Association for Computational Linguistics.

Tourille, J., Ferret, O., Tannier, X., and Névéol, A. (2017b). LIMSI-COT at SemEval-2017 task 12: Neural architecture for temporal information extraction from clinical narratives. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada. Association for Computational Linguistics.

Trieu, H.-L., Nguyen, N. T. H., Miwa, M., and Ananiadou, S. (2018). Investigating domain-specific information for neural coreference resolution on biomedical texts. In *Proceedings of the BioNLP 2018 workshop*, pages 183–188, Melbourne, Australia. Association for Computational Linguistics.

Tsafnat, G., Glasziou, P., Choong, M. K., Dunn, A., Galgani, F., and Coiera, E. (2014). Systematic review automation technologies. *Systematic reviews*, 3(1):74.

Tsafnat, G., Glasziou, P., Karystianis, G., and Coiera, E. (2018). Automated screening of research studies for systematic reviews using study characteristics. *Systematic Reviews*, 7(1):64.

Tsuruoka, Y. and Tsujii, J. (2004). Improving the performance of dictionary-based approaches in protein name recognition. *Journal of biomedical informatics*, 37(6):461–470.

Tsuruoka, Y., Tsujii, J., and Ananiadou, S. (2008). Accelerating the annotation of sparse named entities by dynamic sentence selection. *BMC bioinformatics*, 9(11):S8.

Tu, H., Ma, Z., Sun, A., Xu, Z., and Wang, X. (2017). Entity recognition by distant supervision with soft list constraint. In *International Conference on Advanced Data Mining and Applications*, pages 681–694. Springer.

Underwood, E. (2016). International brain projects proposed.

UzZaman, N. and Allen, J. F. (2011). Temporal evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 351–356, Stroudsburg, PA, USA. Association for Computational Linguistics.

UzZaman, N., Llorens, H., Derczynski, L., Allen, J., Verhagen, M., and Pustejovsky, J. (2013). SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA. Association for Computational Linguistics.

Van Driel, M. A., Bruggeman, J., Vriend, G., Brunner, H. G., and Leunissen, J. A. (2006). A text-mining analysis of the human phenome. *European journal of human genetics*, 14(5):535.

Vasques, X., Richardet, R., Hill, S. L., Slater, D., Chappelier, J.-C., Pralong, E., Bloch, J., Draganski, B., and Cif, L. (2015). Automatic target validation based on neuroscientific literature mining for tractography. *Frontiers in neuroanatomy*, 9:66.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017a). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017b). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, Long Beach, CA, USA.

Velupillai, S., Mowery, D. L., Abdelrahman, S., Christensen, L., and Chapman, W. (2015). BluLab: Temporal information extraction for the 2015 clinical TempEval challenge. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 815–819, Denver, Colorado. Association for Computational Linguistics.

Velupillai, S., Suominen, H., Liakata, M., Roberts, A., Shah, A. D., Morley, K., Osborn, D., Hayes, J., Stewart, R., Downs, J., et al. (2018). Using clinical natural language processing for health outcomes research: Overview and actionable suggestions for future advances. *Journal of biomedical informatics*, 88:11–19.

Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., and Pustejovsky, J. (2007). SemEval-2007 task 15: TempEval temporal relation identification. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 75–80, Prague, Czech Republic. Association for Computational Linguistics.

Verhagen, M., Mani, I., Sauri, R., Littman, J., Knippen, R., Jang, S. B., Rumshisky, A., Phillips, J., and Pustejovsky, J. (2005a). Automating temporal annotation with tarsqi. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 81–84.

Verhagen, M., Mani, I., Sauri, R., Littman, J., Knippen, R., Jang, S. B., Rumshisky, A., Phillips, J., and Pustejovsky, J. (2005b). Automating temporal annotation with TARSQI. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 81–84, Ann Arbor, Michigan. Association for Computational Linguistics.

Verhagen, M., Saurí, R., Caselli, T., and Pustejovsky, J. (2010). Semeval-2010 task

13: Tempeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, Uppsala, Sweden. Association for Computational Linguistics.

Vintar, Š. (2004). Comparative evaluation of c-value in the treatment of nested terms. In *Workshop Description*.

Voita, E., Talbot, D., Moiseev, F., Sennrich, R., and Titov, I. (2019). Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

von Däniken, P. and Cieliebak, M. (2017). Transfer learning and sentence level features for named entity recognition on tweets. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 166–171.

Wagner, G., Nussbaumer-Streit, B., Greimel, J., Ciapponi, A., and Gartlehner, G. (2017). Trading certainty for speed - how much uncertainty are decisionmakers and guideline developers willing to accept when using rapid reviews: an international survey. *BMC Med Res Methodol*, 17(1):121.

Walker, C., Strassel, S., Medero, J., and Maeda, K. (2006). ACE 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57.

Wallace, B. C. and Marshall, I. J. (2016). Extracting PICO sentences from clinical trial reports using supervised distant supervision. *J Mach Learn Res*, 17:1–25.

Wallace, B. C., Small, K., Brodley, C. E., Lau, J., and Trikalinos, T. A. (2012). Deploying an interactive machine learning system in an evidence-based practice center: abstrackr. *ACM SIGHIT Symposium on International Health Informatics*, page 819.

Wallace, B. C., Trikalinos, T. A., Lau, J., Brodley, C., and Schmid, C. H. (2010). Semi-automated screening of biomedical citations for systematic reviews. *BMC Bioinf*, 11(1):55.

Wang, B., Liakata, M., Tsakalidis, A., Georgakopoulos Kolaitis, S., Papadopoulos, S., Apostolidis, L., Zubiaga, A., Procter, R., and Kompatsiaris, Y. (2017).

TOTEMSS: Topic-based, temporal sentiment summarisation for twitter. In *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 21–24, Tapei, Taiwan. Association for Computational Linguistics.

Wang, B. and Lu, W. (2018). Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 204–214, Brussels, Belgium. Association for Computational Linguistics.

Wang, B., Lu, W., Wang, Y., and Jin, H. (2018a). A neural transition-based model for nested mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1011–1017, Brussels, Belgium. Association for Computational Linguistics.

Wang, J., Zhao, L., Ye, Y., and Zhang, Y. (2018b). Adverse event detection by integrating twitter data and vaers. *Journal of biomedical semantics*, 9(1):19.

Wang, Y. (2009). Annotating and recognising named entities in clinical notes. In *Proceedings of the ACL-IJCNLP 2009 Student Research Workshop*, ACLstudent '09, pages 18–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Wang, Y. and Patrick, J. (2009). Cascading classifiers for named entity recognition in clinical notes. In *Proceedings of the workshop on biomedical information extraction*, pages 42–49. Association for Computational Linguistics.

Watanabe, T. and Sumita, E. (2015). Transition-based neural constituent parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1169–1179, Beijing, China. Association for Computational Linguistics.

Wei, Q., Chen, T., Xu, R., He, Y., and Gui, L. (2016). Disease named entity recognition by combining conditional random fields and bidirectional recurrent neural networks. *Database*, 2016.

Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, 3(1):9.

Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Wouters, E., Wouters, B., Augustin, I., and Franssen, F. (2017). Personalized medicine and chronic obstructive pulmonary disease. *Current opinion in pulmonary medicine*, 23(3):241–246.

Wu, D., Lee, W. S., Ye, N., and Chieu, H. L. (2009). Domain adaptive bootstrapping for named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1523–1532, Singapore. Association for Computational Linguistics.

Wu, H., Toti, G., Morley, K. I., Ibrahim, Z. M., Folarin, A., Jackson, R., Kartoglu, I., Agrawal, A., Stringer, C., Gale, D., et al. (2018). Semehr: A general-purpose semantic search system to surface semantic data from clinical notes for tailored care, trial recruitment, and clinical research. *Journal of the American Medical Informatics Association*, 25(5):530–537.

Wunnava, S., Qin, X., Kakar, T., Rundensteiner, E. A., and Kong, X. (2018). Bidirectional lstm-crf for adverse drug event tagging in electronic health records. In *International Workshop on Medication and Adverse Drug Event Detection*, pages 48–56.

Xu, D., Yadav, V., and Bethard, S. (2018a). Uarizona at the made1.0 nlp challenge. *Proceedings of machine learning research (PMLR)*, 90:57.

Xu, G., Wang, C., and He, X. (2018b). Improving clinical named entity recognition with global neural attention. In Cai, Y., Ishikawa, Y., and Xu, J., editors, *Web and Big Data*, pages 264–279, Cham. Springer International Publishing.

Xu, J., Lee, H.-J., Ji, Z., Wang, J., Wei, Q., and Xu, H. (2017a). Uth_ccb system for adverse drug reaction extraction from drug labels at tac-adr 2017. In *Proceedings of Text Analysis Conference (TAC)*, Gaithersburg, Maryland, USA.

Xu, M., Jiang, H., and Watcharawittayakul, S. (2017b). A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1237–1247, Vancouver, Canada. Association for Computational Linguistics.

Xu, M., Jiang, H., and Watcharawittayakul, S. (2017c). A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1237–1247, Vancouver, Canada. Association for Computational Linguistics.

Xu, R., Garten, Y., Supekar, K. S., Das, A. K., Altman, R. B., and Garber, A. M. (2007). Extracting subject demographic information from abstracts of randomized clinical trial reports. In *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems*, Stud Health Technol Inform, pages 550–554. IOS Press.

Xu, S., An, X., Zhu, L., Zhang, Y., and Zhang, H. (2015). A crf-based system for recognizing chemical entity mentions (cems) in biomedical literature. *Journal of cheminformatics*, 7(1):S11.

Xu, Y., Wang, Y., Liu, T., Tsujii, J., and Chang, E. I.-C. (2013). An end-to-end system to identify temporal relation in discharge summaries: 2012 i2b2 challenge. *Journal of the American Medical Informatics Association*, 20(5):849–858.

Yadav, P., Steinbach, M., Kumar, V., and Simon, G. (2018). Mining electronic health records (ehrs): a survey. *ACM Computing Surveys (CSUR)*, 50(6):85.

Yan, J., He, L., Huang, R., Li, J., and Liu, Y. (2019a). Relation extraction with temporal reasoning based on memory augmented distant supervision. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1019–1030, Minneapolis, Minnesota. Association for Computational Linguistics.

Yan, J., He, L., Huang, R., Li, J., and Liu, Y. (2019b). Wiki dataset.

Yang, J., Liang, S., and Zhang, Y. (2018a). Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3879–3889, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yang, X., Bian, J., and Wu, Y. (2018b). Detecting medications and adverse drug events in clinical notes using recurrent neural networks. In *International workshop on medication and adverse drug event detection*, pages 1–6.

Yang, Z., Salakhutdinov, R., and Cohen, W. W. (2017). Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*.

Yao, L., Liu, H., Liu, Y., Li, X., and Anwar, M. W. (2015). Biomedical named entity recognition based on deep neutral network. *Int. J. Hybrid Inf. Technol*, 8(8):279–288.

Ye, Z. and Ling, Z.-H. (2018). Hybrid semi-Markov CRF for neural sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 235–240, Melbourne, Australia. Association for Computational Linguistics.

Yeleswarapu, S., Rao, A., Joseph, T., Saipradeep, V. G., and Srinivasan, R. (2014). A pipeline to extract drug-adverse event pairs from multiple data sources. *BMC medical informatics and decision making*, 14(1):13.

Yoshikawa, K., Riedel, S., Asahara, M., and Matsumoto, Y. (2009). Jointly identifying temporal relations with Markov logic. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 405–413, Suntec, Singapore. Association for Computational Linguistics.

Zavarella, V. and Tanev, H. (2013). FSS-TimEx for TempEval-3: Extracting temporal information from text. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 58–63, Atlanta, Georgia, USA. Association for Computational Linguistics.

Zeng, Z., Deng, Y., Li, X., Naumann, T., and Luo, Y. (2018). Natural language processing for ehr-based computational phenotyping. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(1):139–153.

Zhang, B., Whitehead, S., Huang, L., and Ji, H. (2018a). Global attention for name tagging. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 86–96, Brussels, Belgium. Association for Computational Linguistics.

Zhang, J., Shen, D., Zhou, G., Su, J., and Tan, C.-L. (2004). Enhancing HMM-based biomedical named entity recognition by studying special phenomena. *Journal of biomedical informatics*, 37(6):411–422.

Zhang, N., Deng, S., Sun, Z., Chen, X., Zhang, W., and Chen, H. (2018b). Attention-based capsule networks with dynamic routing for relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 986–992, Brussels, Belgium. Association for Computational Linguistics.

Zhang, Q., Fu, J., Liu, X., and Huang, X. (2018c). Adaptive co-attention network for named entity recognition in tweets. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhang, S., Jiang, H., Xu, M., Hou, J., and Dai, L. (2015). The fixed-size ordinally-forgetting encoding method for neural network language models. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 495–500, Beijing, China. Association for Computational Linguistics.

Zhang, Y., Marshall, I., and Wallace, B. C. (2016). Rationale-augmented convolutional neural networks for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 795. NIH Public Access.

Zhang, Y. L. and Yang, Q. (2017). A survey on multi-task learning. *ArXiv*, abs/1707.08114.

Zhao, H., Yang, Y., Zhang, Q., and Si, L. (2018). Improve neural entity recognition via multi-task data selection and constrained decoding. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 346–351, New Orleans, Louisiana. Association for Computational Linguistics.

Zhao, J., Bysani, P., and Kan, M.-Y. (2012). Exploiting classification correlations for the extraction of evidence-based practice information. In *AMIA Annual*

Symposium Proceedings, volume 2012, page 1070. American Medical Informatics Association.

Zhao, J., Kan, M.-Y., Procter, P. M., Zubaidah, S., Yip, W. K., and Li, G. M. (2010). Improving search for evidence-based practice using information extraction. In *AMIA Annual Symposium Proceedings*, volume 2010, page 937. American Medical Informatics Association.

Zhou, G., Zhang, J., Su, J., Shen, D., and Tan, C. (2004). Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, 20(7):1178–1190.

Zhou, X., Li, L., Dong, D., Liu, Y., Chen, Y., Zhao, W. X., Yu, D., and Wu, H. (2018). Multi-turn response selection for chatbots with deep attention matching network. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1118–1127, Melbourne, Australia. Association for Computational Linguistics.

Zhou, Z.-H. (2017). A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53.

Zhuo, J., Cao, Y., Zhu, J., Zhang, B., and Nie, Z. (2016). Segment-level sequence modeling using gated recursive semi-Markov conditional random fields. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1413–1423, Berlin, Germany. Association for Computational Linguistics.

Zukov-Gregoric, A., Bachrach, Y., Minkovsky, P., Coope, S., and Maksak, B. (2017). Neural named entity recognition using a self-attention mechanism. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 652–656. IEEE.