2022

A study on developing novel methods for relation extraction

DARSHINI MAHENDRAN

DISSERTATION

A STUDY ON DEVELOPING NOVEL METHODS FOR RELATION EXTRACTION

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

DARSHINI MAHENDRAN

BS, University of Peradeniya, Sri Lanka - Aug 2011 to Dec 2015

Director: Bridget T McInnes, Ph.D.,

Associate Professor, Department of Computer Science

Virginia Commonwealth University

Richmond, Virginia

May, 2022

Acknowledgements

First, I would like to thank my advisor Dr. Bridget McInnes for the patient guidance, and encouragement. I was fortunate to have an advisor who looked out for me all time, whose words always motivated me, and ideas never failed to inspire me.

I would like to thank Dr. Nastassja Lewinski and Dr. Christina Tang, who are my collaborators, and mentors. I have always enjoyed working with them and learned a lot from them. A special thanks to Dr. Alberto Cano Rojas and Dr. Tom Arodz for being part of my Dissertation Committee.

A very special gratitude goes out to all of my co-authors, my fellow doctoral students, lab members and friends in particular, Dr. Sam Henry, Amy Olex, Clint Cuffy, Nicholas Rodriguez, Megan Davis, Megan Charity, Andriy Mulyar, Gabby Gurdin, Luke G Maffey, and Jorge Vargas. Also, I would like to acknowledge the VCU College of Engineering and the staff for providing me the opportunity to be part of this wonderful community and meet so many interesting people.

Finally, I must express my very profound gratitude to my family. Thank you all for your encouragement!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Abstract

DISSERTATION

A STUDY ON DEVELOPING NOVEL METHODS FOR RELATION EXTRACTION

By Darshini Mahendran

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2022.

Director: Bridget T McInnes, Ph.D.,
Associate Professor, Department of Computer Science

Relation Extraction (RE) is a task of Natural Language Processing (NLP) to detect and classify the relations between two entities. Relation extraction in the biomedical and scientific literature domain is challenging as text can contain multiple pairs of entities in the same instance. During the course of this research, we developed an RE framework (RelEx), which consists of five main RE paradigms: rule-based, machine learning-based, Convolutional Neural Network (CNN)-based, Bidirectional Encoder Representations from Transformers (BERT)-based, and Graph Convolutional Networks (GCNs)-based approaches. RelEx's rule-based approach uses co-location information of the entities to determine whether a relation exists between a selected entity and the other entities. RelEx's machine learning-based approach consists of traditional feature representations into traditional machine learning algorithms. RelEx's CNN-based approach consists of three CNN architectures: Segment-CNN,

single-label Sentence-CNN, and multi-label Sentence-CNN. RelEx's BERT-based approach utilizes BERT's contextualized word embeddings into a feed-forward neural network. Finally, RelEx's GCN-based approach consists of two GCN-based architectures: GCN-Vanilla, GCN-BERT. We evaluated variations of these approaches in two different domains across four distinct relation types.

Overall our findings showed that the rule-based approach is applicable for data with fewer instances in the training data. In contrast, the CNN-based, BERT-based, and GCN-based approaches perform better with labeled data with many training instances. These approaches automatically identify patterns in the data efficiently, whereas rule-based approaches require expert knowledge to generate rules. The CNN-based, BERT-based approaches capture the local contextual information within a sentence or document by embedding both semantic and syntactic information in a learned representation. However, their ability to capture the long-range dependency global information in a text is limited. GCN-based approaches capture the global association information by performing convolution operations on neighbor nodes in a graph and incorporating information from neighbors. Combining GCN with BERT integrates the local contextual and global association information of the words and generates better representations for the words.

CHAPTER 1

INTRODUCTION

Today we live in the Information age, during which data is growing at an exponential rate [1]. The rapid growth of data across all domains has caused retrieving relevant information from data a very expensive task [2]. Moreover, the data available is usually in an unstructured or highly heterogeneous format [3]. Information Extraction (IE) is a task of Natural Language Processing (NLP) that extracts structured information from raw unstructured text, which a machine or a program can easily interpret. NLP is a field of Artificial Intelligence (AI) that allows machines to interpret human language. It is the driving force behind applications such as chatbots, voice assistants, and search engines. NLP, a combination of linguistics and computer science, studies the rules and structure of language to create systems that analyze and extract meaning from text and speech. However, extracting information from unstructured data requires not only a considerable amount of manual effort but is also a time-consuming task [3]. Therefore, the need for IE systems that can detect and extract information automatically has grown recently. IE systems have various applications in different industries such as business, medical institutions, military, and research institutions.

IE consists of several subtasks such as Named Entity Recognition (NER), Relation Extraction (RE), and Event Extraction (EE). RE automatically detects and classifies relations between entities in a text. Identifying and extracting relations is important for many downstream applications such as question answering [4], summarization [5], and information retrieval [6]. The task of RE includes extracting sentences that contain the entity pair which can hold a semantic relation and then

predicting whether a certain relation exists between the specified entity pair [7]. For example, Fig. 1 shows an example of a sentence from a clinical dataset with multiple entities: *Problems*, *Treatments*, and *Tests*.

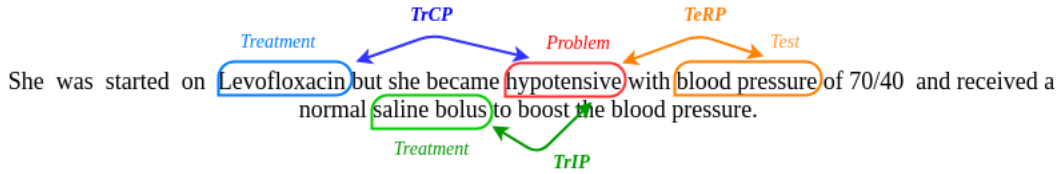

Fig. 1. Sentence from i2b2-2010 relation corpus depicting multiple pairs of medical entities in the same sentence.

Many approaches have been explored with RE over the years. Lately, deep learning approaches have gained popularity as they require less human inputs but yield better accuracy when trained with a huge amount of data. In earlier days, traditional non-deep learning methods such as pattern-based, rule-based, and machine-learning were used to extract relations. Rule-based and pattern-based approaches use predefined rules and patterns to identify relations between words, whereas machine-learning approaches utilize algorithms to automatically learn patterns. Supervised techniques for machine learning required a large amount of training data for learning, and the manually constructed features did not capture all the relevant information [8]. In recent years, deep learning techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), which achieved outstanding results at many NLP tasks, were used for RE [3]. More recently contextualized deep neural language models such as ELMo [9], Bidirectional Encoder Representations from Transformers (BERT) [10], and Generative Pre-trained Transformer-2 (OpenAI GPT-2) [11] were introduced and they yielded significant improvements in extracting relations. These techniques capture the local contextual information in a sentence or document well by embedding semantic and syntactic information in a

learned representation. However, their ability to capture the long-range dependency global information in a text is limited. Utilizing the global association information between words outside the sentence boundaries can help generate better representations. Graph Neural Networks (GNNs) are deep learning models that operate in the graph domain and capture global information between words/ phrases. Recent research based on the graph has been receiving more attention due to the great expressive power of graphs [12]. The variant of GNN that gained popularity recently is Graph Convolutional Networks (GCNs) [13]. GCN captures the global context information by performing convolution operations on neighbor nodes in a graph and incorporating information from neighbors. GCNs can preserve global structure information of a graph in graph embeddings [14].

In this dissertation, we propose novel methods to identify and extract relations in text automatically. We discuss rule-based, machine-learning based, CNN-based, and GCN-based methods. Dissertation consists of 12 chapters: Chapter 2 provides the background information relevant to understanding the concepts and algorithms used in our work. Chapter 3 provides an overview of the related work. Chapter 8 summarizes all the data we used in our work. Chapter 5 describes our RE system and the details of each approach. Chapters 6 to 11 present the details of our experiments, their contributions to the field, and the overall conclusions of our experiments to date. Finally, Chapter 12 discusses our ideas for the future.

CHAPTER 2

BACKGROUND

In this chapter, we discuss concepts and algorithms used in our work.

2.1 Feature Representation

Words in a text provide context but machine learning or deep learning algorithms are not capable of processing information in the string format. The raw text needs to be converted to numerical form for the algorithms to process the information. Feature representation refers to generating a representation of language for this purpose. These representations can be subdivided into three categories: 1) traditional features, 2) static word embeddings, and 3) contextualized word embeddings. Traditional feature representations are categorical representations, whereas both the static and contextualized embedding representations are learned. In this section, we discuss each of these representations.

2.1.1 Traditional Features

Traditional feature representations utilize categorical information known about language to represent words and terms. Here, tokens are represented as an n-dimensional vector where each element refers to information known about the word. Fig. 2 illustrates how feature vectors are generated: (A) shows how tokens are represented in a vector format, and (B) and (C) show examples of how two sentences are vectorized. This information can be divided into three categories: lexical features, syntactic features, and semantic features.

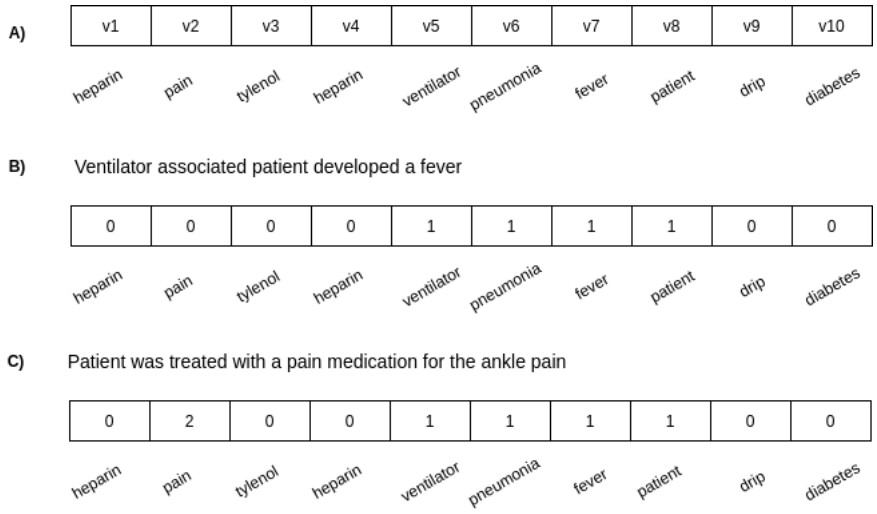

Fig. 2. Illustration of feature vectorization

Lexical features. Words provide context. Lexical features analyze tokens at the word level with respect to their context. They are derived from the token in context with other words used in the sentence or text. Lexical features of a token can include the word itself, its root form (e.g., lemma or stem), its morphological information (e.g., prefix or suffix), its shape (e.g., uppercase, lowercase, title-case), and its surrounding words (i.e., the context).

The underlying idea of contextual information is *a word is known by the company it keeps* [15]. This stems from the understanding that similar words are used in similar contexts. For example, the token *bat* when seen with the context of *baseball* has a different meaning than when seen with the context of *vampire*. The n-gram modeling is used to identify the relevant contexts of a word. The n-grams are ordered sequences of N tokens where N refers to the number of words in a sequence. They can be unigram, bigram, or trigram, where a sequence of tokens is formed from one, two, or three adjacent words, respectively. This is often referred to as a Bag of Words (BOW) model because it represents a token as the bag of other words (n-grams)

seen within some window size of the token in a corpus [16]. Within the feature representation vector, the n-grams are represented numerically. Often this can be a binary representation indicating the presence of the n-gram with respect to the token or a frequency representation indicating how often the token was seen with n-gram as seen in Example B of Fig. 2.

However, with multiple documents inside a corpora, the Term Frequency - Inverse Document Frequency (TF-IDF) [17] takes into account how frequently tokens they appear across multiple documents [18]. TF-IDF is calculated by multiplying two different metrics: Term Frequency (TF), Inverse Document Frequency (IDF). Equations 2.1, 2.2, and 2.3 show the formula to calculate the TF-IDF [19]. TF (Equation 2.2) measures the raw count of a word in a document denoted by $tf(t, d)$, i.e., number of times that word t occurred in the document d. IDF (Equation 2.3) measures how common the word is across multiple documents [19] denoted by $idf(t, D)$ where N stands for the total number of documents in the corpus. Higher the score, higher the relevancy of the word in the specific document.

$$tfidf(t, d, D) = tf(t, d).idf(t, D) \tag{2.1}$$

where:

$$tf(t, d) = log(1 + freq(t, d)) \tag{2.2}$$

and,

$$idf(t, D) = log(\frac{N}{count(d \epsilon D : t \epsilon d}) \tag{2.3}$$

Syntactic features. Syntactic features analyze the structural role of words by considering the order and the position of the word in a sentence or text. Syntactic features include a word's part of speech (POS) as well as its parse information. POS

tags provide the grammatical construct of a token [20]. These tags can be then used to build a parse tree which captures the syntactic relations between the tokens.

Semantic features. Semantic features analyze the meaning of a word and how it is arranged in a sentence contributes to the meaning of the sentence. The relation between two words can differ in three ways: 1) Homonymy - a word that has more than one sense, 2) Polysemy - two senses that are related semantically, 3) Metonymy - use one aspect of a sense to refer to another aspect of the sense. Here, the sense is the semantic representation of a term. For example, the word *bat* could either refer to a *baseballbat* or a *vampirebat* depending on the context in which it is used. Semantic features are often extracted from taxonomies and ontologies such as Unified Medical Language System (UMLS) [21], WordNet [22].

2.1.2 Static Word Embeddings

The disadvantages of traditional feature representations are a lot of feature engineering is required to select the best feature for any task. As a result, the vectors are large, sparse, and noisy. Word embeddings attempt to alleviate this by learning a numerical representations for the words. The basic idea behind these embeddings is that a model learns the probability of words co-occurring together from large corpora. Static word embeddings maps each word in a text to a single vector. Two most commonly used algorithms to generate word embeddings are word2vec [23] and Global Vectors (GloVe) [24] embeddings.

The word2vec algorithm provides direct access to vector representations of words. It has two versions: Skip-Gram and Continuous Bag of Words Model (CBOW) [23] formally defined in Equations 2.4 and 2.5 respectively where w is the window size, and T is the number of words in the corpus. Skip-gram uses the distributed representation of the input word to predict the context, i.e., the model learns the embedding by

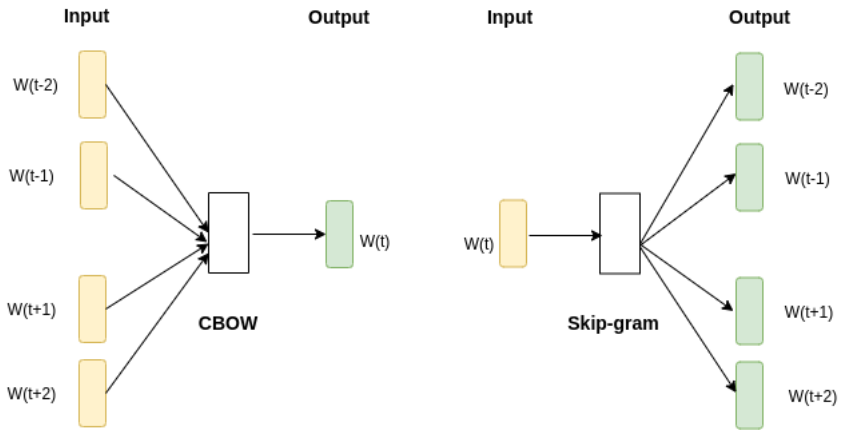

Fig. 3. Architecture of the two word2vec training models

predicting the context. In contrast, CBOW combines the distributed representations of context to predict the word in the middle, i.e., the model learns the embedding by predicting the current word based on its context. Fig. 3 illustrates how both word2vec algorithm models generate embeddings.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{0<j\leq c;} log(p(w_t|w_{t-j})) \tag{2.4}$$

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c; j\neq 0} log(p(w_{t+j}|w_t)) \tag{2.5}$$

Unlike word2vec, GloVe [24] takes advantage of the global count statistics and works based on the co-occurrence ratios between two words. The algorithm takes local context into account by building a co-occurrence matrix using a fixed window size. If the corpus contains W number of words, the size of the co-occurrence matrix X will be $W * W$ where the i^{th} row and j^{th} column of X, x_{ij} denotes how many times word i has co-occurred with word j. Next, they predict the co-occurrence ratios using the word vectors as shown in the Equation 2.6 where P_{ij} refers to the probability of the word j appearing in the context of i. The function F takes the

log of the probability ratios and adds a bias term to capture whether words occur by chance or not. GloVe calculates the loss using word frequency as the infrequent co-occurrences tend to be noisy. Although both algorithms have their differences, surprisingly they both perform very similarly.

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ij}}{P_{jk}} \tag{2.6}$$

$$dot(w_i, \tilde{w}_k) + b_i + \tilde{b}_k = log(X_{ik}) \tag{2.7}$$

The disadvantage of the static word embeddings is a word only has a single vector representation regardless of the context. For example, the word *bank* can indicate a *financial bank* or a *river bank* but the static word embedding representation of both these words would be similar. In other words, all senses of a word share the same representation.

2.1.3 Contextualized Word Embeddings

The disadvantage of static word embeddings is, a word only has a single vector representation regardless of the context in which it is currently being used. To address all this issue, contextualized deep neural language models such as ELMo [9], BERT [10], and OpenAI GPT-2 [11] have been introduced. They attempt to incorporate the context in which a word is currently being used into the representation. This has yielded significant improvements on many NLP tasks [25].

Contextualized word embeddings capture the word semantics in different contexts. The contextualized language models learn sequence-level semantics in a document using the sequence of all words in that document. For example, contextualized word embedding representations of the words *financial bank* and *river bank* would be

different.

2.2 Deep Learning Algorithms

Deep learning models have recently started dominating the NLP field, as they can effectively learn meaningful hidden features without manual feature engineering. Deep learning refers to learning models that utilize multi-layer Artificial Neural Networks (ANNs). These algorithms work well with data that consists of hidden patterns or complex relations among entities. ANNs gained popularity in the past decade, and different variants of simple neural networks achieved success in many research fields. One such network is the Convolutional Neural Networks (CNNs). Different variants of ANNs achieved success in many research fields; however, most of these variants deal with euclidean data while many real-world data are non-euclidean. These data have led to the recent invention of variants – Graph Convolutional Networks (GCNs). Moreover, many ANNs cannot deal with long input sequences well, leading the gradients to vanish or explode. Also, since they process the inputs sequentially, they are slow. In 2017, transformers were introduced to solve these problems [26]. A transformer is a deep learning model that employs Encoder-Decoder architecture, and BERT is a transformer-based model developed by Google for NLP. The proposed models in this work are based on the following deep learning algorithms: CNN, GCN, and BERT. In this section, we discuss each of these these algorithms and their functionality.

2.2.1 Convolutional Neural Networks (CNN)

A typical NLP model that utilizes CNNs, primarily consists of four main layers [27]:

1. embedding layer - to encode words in sentences by real-valued vectors

2. convolution layer - to get local features from each part of the input
3. pooling layer - to extract the most relevant features
4. feed-forward layer - a fully connected layer to perform classification

Here we describe each layer in detail.

Embedding layer. Neural networks learn information through a numerical representation of the data. Word embeddings map a set of words or phrases in a vocabulary to real-valued vectors, which helps to reduce the dimensionality and learn linguistic patterns in the data [28]. They capture both semantic and syntactic properties of words so that semantically similar words produce close embedding vectors. CNNs allow word embeddings to train on the input text itself or use pre-trained word vectors obtained from an external resource. When using pre-trained word embeddings, the weights of the layer is replaced by the pre-trained vector weights.

Convolution layer. The Convolution layer is where CNN learns and calculates the learnable parameters. In other words, if a layer has weight matrices, that is referred to as a "learnable" layer. Convolution is applied to the text to extract local features from the input. The matrix of a size of (dimensions x input length) representing the input relation mention is fed into this layer, and high-level features are extracted. The input channels consist of kernels/filters with randomly initialized weights. The kernels are applied over the text and combined by summing them up and adding a bias for each channel. Since we deal with the text as input, we use a 1D kernel per input channel. 1D kernels are controlled by depth, as the width and height do not change. 1D convolutions compute a weighted sum of the features and select particular combinations of features that are useful. These weights are re-assigned according to the error that is back-propagated. This process can be replicated for various filters with different window sizes to increase or decrease the coverage of the model. For RE, when applying kernels, we consider the n-grams accompanied by the relative

positions of its words [27]. To obtain local features from each part of the sentence, multiple filters of varying lengths are used in all possible continuous combinations of the n-gram. Using a small kernel instead of a fully connected network benefits from weight sharing and reduction in computational costs. Since we move the same kernel over different words, the same weights are shared with a text-oriented description as we convolve on them. Moreover, since the weights are less than a fully connected layer, we have lesser weights to back-propagate on.

Pooling layer. Pooling is applied to extract the global feature from the input to learn the most essential feature. This layer does not include any learnable parameters. Therefore, it does not include any backpropagation learning. It further abstracts the output of the features from the convolution layer by applying an aggregating function. Max-pooling is the most commonly used aggregation function, and as the name suggests, it just picks the maximum value in a specific size window. Average-pooling is another aggregation function used where it averages the windows instead of picking the maximum value. The pooled features are then fed to a fully connected feed-forward neural network to make an inference.

Feed-forward layer. The feed-forward layer is a regular layer of neurons in a neural network. It is densely connected so that each neuron receives input from all the neurons in the previous layer. The pooling layer outputs a sequence, which is then concatenated into a single feature vector to represent the relation mention before it is fed into this layer. This layer more has learnable parameters as it is fully-connected. It is calculated by taking the number of neurons in the current layer and the number of neurons on the previous layer and adding the bias term. This output layer uses a softmax classifier with the number of outputs equal to the number of possible relations between entities. The Dropout technique is applied in between any of the above layers. Dropout is a regularization technique commonly used in neural networks to reduce

interdependent learning amongst the neurons.

2.2.2 Bidirectional Encoder Representations from Transformers (BERT)

In 2018, Google introduced BERT [10], a language model that utilizes an attention mechanism to model semantic relations between words of a text. BERT is the first bidirectionally trained language model, models before that train left-to-right or vice versa. In addition, BERT produces contextual embedding representations of a token. These representations can be fine-tuned for specific domains.

To do this, BERT utilizes a transformer that consists of an encoder to read the input. The language model generation takes part in the encoder, which reads the input. The input representation is the sum of the token, segmentation, and position embeddings. Following are the specific embeddings layer representations, and Fig. 4 shows how the input embeddings are concatenated [10]:

- Token embeddings - Transforms words into vectors of fixed dimensions. A [CLS] token is added to word tokens at the beginning of the first sentence and a [SEP] token at the end of each sentence.
- Segment embeddings - Adds a marker to indicate which sentence the word token is from. If the input is from one sentence only, then outputs a vector corresponding to index 0
- Positional embeddings - Learns a vector representation for each position. Positional information shows the position of the word token in the sentence.

BERT is a transformer-based model that uses an attention mechanism to learn the contextual relations between words in a text. A transformer architecture contains an encoder and a decoder; since BERT generates a language model, it uses an encoder only. BERT does not just predict the next word in the sentence but randomly masks

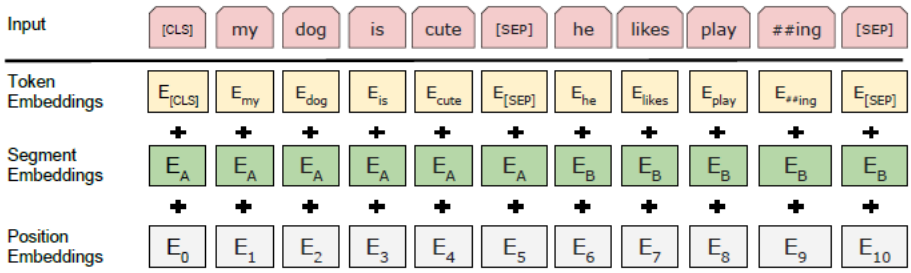

Fig. 4. BERT input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings [10]

words in the sentence, and then it tries to predict them. BERT uses two strategies to train:

1. Masked Language Modeling (MLM) - BERT masks the 15% of the words in the input sequence by replacing the words with a [MASK] token. Then the model tries to predict the masked words based on the context of the surrounding words in both directions of the masked word. Unlike other language models, it considers both the previous and next tokens simultaneously. To prevent the model from trying to predict only when the [MASK] token is present in the input, out of the 15% of the tokens selected for masking [10],
    - 80% of the tokens are replaced with the token [MASK].
    - 10% of the tokens are replaced with a random token.
    - 10% of the tokens are left unchanged.
2. Next Sentence Prediction (NSP) - The model receives pairs of sentences and learns to predict if the second sentence is the subsequent sentence in the original text. This helps the model to learn the relationship between two sentences.

The BERT model is trained with these strategies, and they minimize the combined loss function of both these strategies. BERT is based on stacked layers of

encoders and has two model sizes: BERT-base and BERT-large. BERT base model has 12 encoder layers, whereas BERT large has 24 layers of encoders stacked on top of each other. Following are the configurations of the pre-trained BERT models [10]:

- BERT-Base, Uncased: 12-layers, 768-hidden, 12-attention-heads, 110M parameters
- BERT-Large, Uncased: 24-layers, 1024-hidden, 16-attention-heads, 340M parameters
- BERT-Base, Cased: 12-layers, 768-hidden, 12-attention-heads, 110M parameters
- BERT-Large, Cased: 24-layers, 1024-hidden, 16-attention-heads, 340M parameters

2.2.3 Graph Convolutional Networks (GCN)

Most of these variants deal with euclidean data, while many real-world data are non-euclidean. These data have led to the recent invention of variants – GNNs. GNNs are a deep learning-based method that extends existing neural network methods to operate on the data represented in graph domains [29]. GNNs deal with non-euclidean graph data that contains rich relational information between elements. The following highlight the advantages of GNNs over CNNs [29]:

- Traditional neural networks such as CNNs and RNNs operate on regular euclidean data like images (2D grid) and do not handle non-euclidean types data well because they stack features by a specific order. The graph data do not have a natural order of nodes, and nodes can be traversed in different orders.

- The dependency between two nodes in the graphs is represented by an edge in GNN, whereas they are considered just another feature in the traditional networks.

- Traditional networks learn by the distribution of the data, whereas GNNs generate graphs from non-structural and learn the reasoning, which can be helpful in high-level AI-related research.

GCNs [13] are a recent variant of the basic GNN architectures that are designed to perform inference over data described using a graph. Given a graph $G = (V, E)$, a GCN takes the following as the input [13]: an input feature matrix $N * F$, where N is the number of nodes and F is the number of input features for each node, a feature matrix X, and an $N * N$ matrix representation of the graph structure such as the adjacency matrix A of G. GCNs utilize the *message passing* mechanism, which is performed through matrix operations where the information is passed from one node to another. Each layer of the GCN defines a propagation rule in the form of a matrix, which determines how inputs will be transformed before being sent to the next layer. In this layer, the incoming feature matrix is multiplied by the adjacency matrix as shown in the Equation 2.8

$$f(H^i, A) = \sigma(A\,H^i\,W^i) \tag{2.8}$$

where W^i is the weight matrix for layer i, σ is a non-linear activation function such as the ReLU function, H^i is a hidden layer, f is a propagation rule, and $H^0 = X$, the feature matrix. This helps the features to become increasingly more abstract at each consecutive layer. The basic operations of the GCNs are similar to CNNs. The convolution is applied in CNNs by multiplying the input neurons with weights commonly known as filters or kernels. GCNs perform a similar operation to learn

the features of the neighboring nodes. However, the difference is that the nodes in a GCN are unordered, and the connections between nodes are not uniform (irregular non-euclidean data), whereas CNNs operate on regular euclidean data.

Kipf, et al. [13] presented GCNs in their pioneering work showing it achieved state-of-the-art classification results on several benchmark graph datasets including Stanford Sentiment Treebank (SST-2) [30], Corpus of Linguistic Acceptability (CoLA) [31], and ArangoHate [32].

CHAPTER 3

RELATED WORK

In this chapter, we discuss the works related to our works. Many approaches have been explored and divided into five paradigms: 1) rule-based, 2) machine learning-based 3) deep learning-based, 4) contextualized language model-based, and 5) GCN-based approaches.

3.1 Rule-based approaches

Rule-based approaches use rules and patterns to identify relations between words. They are one of the oldest approaches of NLP [33]. Rule-based technologies are easy to comprehend, maintain, incorporate domain knowledge, and not very challenging to debug; however, these technologies require tedious manual labor to generate rules [33]. Besides, rule-based approaches are task and domain-specific, making it difficult to generalize to new relations and domains.

For example, Li, et al. [34] proposed a rule-based method to link drug names with their attributes using regular expressions to match drug names to a prescription list then co-location information to link the attributes to the drugs. He, et al. [35] also proposed a rule-based method to identify relations between chemical reactions synthesis. They compiled two dictionaries from the training data: 1) a list of possible entities and their mentions, and 2) a list of possible relations between entities. The first dictionary was used to identify the entities, and the second to identify the relations between entities that occur in the same sentence. Lowe, et al. [36] used the ChemicalTagger [37] an open-source tool that uses syntactic information, including

the POS and parses the information, to identify relations between entities. They assigned relations using a predefined set of rules based on the syntactic information between the entities.

Bejan, et al. [38] compared three methods to extract treatment relations from clinical text: use SemRep, simple rule based algorithm using MEDI [1], and combinations of our MEDI rules and SemRep. SemRep was originally developed for processing text from the biomedical research literature at the US National Library of Medicine [2]. MEDI7 is a publicly available resource with information aggregated from four resources: RxNorm, Side Effect Resource (SIDER) [39], MedlinePlus, and Wikipedia. They also constructed a dataset with manually annotated treatment relations which included 6864 randomly selected discharge summaries from the Vanderbilt Synthetic Derivative, a de-identified version of the electronic medical record. They concluded that SemRep was effective at extracting treatment relations from clinical text. However, both the MEDI algorithm and the union ensemble system significantly outperformed SemRep for this task.

3.2 Machine Learning-based Approaches

Machine learning-based approaches use machine learning algorithms and statistical analysis to derive the meaning from a text. Machine learning approaches including Hidden Markov Models (HMM), Conditional Random Field (CRF), Maximum Entropy Models (MaxEnt), Support Vector Machines (SVMs), Naïve Bayes (NB), and Random Forests (RFs) [40] dominated the NLP field until recently. These approaches are trainable and more easily adaptable compared to rule-based approaches; however, they require labeled data and the field expertise for usage and maintenance [33].

---

[1] http://knowledgemap.mc.vanderbilt.edu/research/

[2] http://semrep.nlm.nih.gov/

SVMs dominate most of the machine-learning-based approaches [41, 42, 43, 44]. They mainly focus on identifying traditional features best able to classify relations. For example, Miller, et al. [45] used token context features, character type features, and semantic features as input to SVMs. They represented the entity's context features first as BOW within a window and also with relative positional information. Patrick, et al. [41] utilized: 1) local context features such as words before, after, and between the two entities; and 2) semantic features such as assertion type and its lexicon type of the entities. Demner-Fushman, et al. [43] incorporated semantic information using concepts from the UMLS [21]. They used BOW feature representing that included the entities, the words surrounding the entities, their UMLS Concept Unique Identifier (CUI), semantic type, an assertion value, and co-occurrence of the entity pair. Rink, et al. [46] utilized contextual, similarity, nested relations, Wikipedia, and concept vicinity features to identify relations between entities using an SVM. Zhu, et al. [47] focused on revising the learning algorithm by reformulating the SVM into a composite-kernel framework to achieve better performance. They also explored various superficial word/phrase/concept features, syntactic structures, and additional domain-specific features. De Bruijn, et al. [48] extracted token, word, sentence, document level features, n-grams, and CUI features from the text and augmented them with semantic information from external sources; specifically the UMLS [49], cTAKES [50], and Medline[3]. Also they found the following features and design decisions to be beneficial:

1. Using Charniak parser [51] to parse the input texts and then transferring them into Stanford dependencies [52] added an additional gain in F-score on 5-fold cross validation (CV).

[3]http://mbr.nlm.nih.gov/Download/index.shtml

2. Downsampling the training set to a defined positive class/negative class ratio between reduced a classifier's bias towards the majority class, and improved the overall F-score on 5-fold CV.

3. Using Medline as semi-structured source of knowledge to approximate the relatedness of these concepts yielded improvement during development.

4. Applying bootstrapping on the unlabeled data helped to improve the performance.

Hybrid approaches that incorporate rules and machine learning have also been explored specially many works combine linguistic pattern matching with ML techniques. For example, Grouin, et al. [42], Porumb, et al. [53], Solt, et al. [44], and Minard, et al. [54] utilized hand-built linguistic patterns as features into SVMs. Grouin, et al. [42] trained an SVM and constructed linguistic patterns manually. They reported two advantages of the hybrid approach: (1) linguistic patterns provide a substantial enhancement of the obtained results for classes that do not have enough instances to feed the automatic classifiers, (2) linguistic patterns may confirm automatically-induced relations which helps adding confidence to the obtained results. Minard, et al. [54] also trained an SVM and constructed linguistic patterns manually. Here, for each semantic relation they used a manually constructed set of lexical patterns which are regular expressions describing a set of matching sentences containing medical entities at specified positions with a more or less specific lexical context. Also they used the following features on the SVM for classification: surface features (number and order of tokens), lexical features (tokens, stemmed tokens in concepts, left-right trigrams of concepts, surrounding words of concepts, prepositions between concepts, headword of concepts), syntactic features (POS, presence of a preposition, presence of a coordinating conjunction between concepts and punctuation signs), and semantic

features(UMLS semantic type of tokens in a three-word window on either side of each argument concept, concept types, Levin's class of the verbs [4]).

Solt, et al. [44] built binary classifiers for each class using statistical and ML approaches. For the ML approach they also utilized linguistic kernels along with SVMs and for the statistical approach they utilized the co-occurrence between the words as the features. Following SVM kernels were experimented: Shallow linguistic (SL), Parse tree (SpT, PT), and Dependency graph (APG, kBSPS) kernels.

While Yang, et al. [55] applied heuristic rules to generate candidate pairs of possible related entities that were fed into three ML models (SVM, RF, and Gradient Boosting [56]) to classify the relations. The possible related entities were identified according to their distance from each other as defined by the number of sentence boundaries between the two entities, and multiple classifiers were developed to classify relations based on this distance.

Table 1. Overview of the works that utilized machine-learning based approaches for relation extraction in the clinical domain.

Year	**Paper**	**Dataset**	**ML Algorithm**	**Features**
2010	De Bruijn, et al. [48]	i2b2-2010	SVM	orthographic, syntactic, semantic
2010	Anick, et al. [57]	i2b2-2010	SVM	semantic
2010	Grouin, et al. [42]	i2b2-2010	SVM, pattern matching	surface, lexical, syntactic
2010	Solt, et al. [44]	i2b2-2010	SVM, pattern matching	co-occurrence, syntactic
2011	Minard, et al. [54]	i2b2-2010	SVM, pattern matching	surface, lexical, syntactic, semantic

[4]http://verbs.colorado.edu/%E2%88%BCmpalmer/projects/verbnet.html

3.3 Deep Learning-based Approaches

Deep learning is a field derived from machine learning that use multi-layer ANNs. In recent years, deep learning-based approaches have gained in popularity due to their demonstrated success at tackling complex learning [58]. These approaches typically take word embeddings representations such as word2vec [23] and GloVe [24] embeddings, BERT embeddings as input.

Two common techniques used for RE are CNNs and RNNs. CNNs can capture continuous local features of sequences through the convolution operation, whereas RNNs obtain long-term dependencies through the recursive process. CNNs with convolving filters were initially developed for computer vision, but they are effective for RE. For example, Sahu, et al. [59] and Luo, et al. [60] utilized CNN for relation classification. They applied CNNs to clinical datasets to automatically learn feature representations to reduce the need for engineered features. Sahu, et al. [59] applied CNN to build a Sentence-CNN which learns a single sentence-level representation for each relation, using six discrete features such as exact word (W), distance from the first entity in terms of the number of words (P1), distance from the second entity in terms of the number of words (P2), POS tag of the word, chunk tag of the word, and type of the word. But their sentence-CNN learns a relation representation for the entire sentence and does not explicitly distinguish the segments of the sentence that form the relations. To fix this, Luo, et al. [60] proposed Segment-CNN, which decomposes a sentence into five segments (preceding, concept1, middle, concept2, and succeeding) and uses multiple convolution units to process the segments individually. Also, they utilized only word-embedding features without manual feature engineering.

Chauhan, et al. [61] built a unifying framework across multiple domains using datasets from the general, biomedical, and clinical domains to allow for the system

to be extendable to new datasets. They used a CNN with position embeddings and a ranking loss. They studied each dataset by performing a systematic exploration of modeling, pre-processing, and training methodologies to determine which of these three was the more significant contributor to performance. They found that pre-processing choices are a significant contributor to performance and that omission of such information can further hinder fair comparison. He, et al. [62] proposed a CNN-based architecture for medical relation classification on clinical records. Further, they introduced a multi-pooling operation to capture the position information of local features relative to the concept pair and a novel loss function with a category-level constraint matrix. Generally, the max-pooling operation extracts the most significant feature in a convolutional filter, but they introduced the multi-pooling operation to achieve more local features in each sentence. They concluded feature extraction based on concept pair positioning could improve the efficacy of relation classification.

Lv, et al. [63] proposed the adoption of a CRF model and applied a deep learning model for features optimization by the employment of autoencoder and sparsity limitation. The features they utilized for the CRF can be categorized into four types: words of context, POS of words, concept type, and distance between two concepts. The one-hot vector representation of the features leads to the sparsity problem, and a deep learning model was used to optimize context features of concepts.

Long Short Term Memory (LSTM) networks are a special kind of RNN that stores the information of the long past inputs [64]. Bidirectional LSTMs (BiLSTMs) are derived from LSTMs; LSTMs flows the data in one direction, whereas BiLSTM flows from both directions [58]. Luo, et al. [65] proposed one of the first models based on LSTM with only word2vec word embedding features and no manual feature engineering for clinical RE. They also evaluated both sentence and segment level LSTMs. For segment-level LSTM, they divided the sentence into five segments: be-

fore the first concept (preceding), of the first concept(concept 1), between the two concepts (middle), of the second concept (concept 2), and after the second concept (succeeding). Since their models did not outperform the top systems with manually engineered features that participated in the i2b2-2010 challenge, they suggested that there is still merit in the curated features and domain-specific knowledge.

Christopoulou, et al. [66] developed separate models for intra and inter-sentence RE and combined them using an ensemble method. They developed two intra-sentence models that use BiLSTMs with attention mechanisms to capture dependencies between multiple related pairs in the same sentence. The first model, Weighted BiLSTM, extracted relation patterns that are in the input sequence. The second model, Walk-based model, is extended by stacking a walk layer on top of the former, and it used sentential entity graphs to infer relations between entities. They adopted a neural architecture that uses the transformer network to improve performance for longer sequences for the inter-sentence relations. Ningthoujam, et al. [67] proposed an approach for RE based on the shortest dependency path (SDP) generated from the dependency parsed tree of the sentence using an LSTM model. They developed a dependency parser to extract the shortest dependency path between the entities and fed the SDP-based words, POS, and the types of the entities as the input into the LSTM layer. They claimed that SDP generated by the dependency parser is a better feature representation for the clinical domain. Li, et al. [68] utilized a deep neural network that models SDP between target entities together with the sentence sequence to capture the syntactic features and further improve the performances of clinical RE. First, they used a Bi-LSTM to capture the position features in the sentence sequence, then they used CNN and Bi-LSTM to capture the syntactic context for target entities using SDP information. Finally, they utilized a fully connected layer with a softmax function to classify the relations. Through the experiments, they realized integrating

SDP features improves the accuracy of the prediction because of the following reasons: 1) length of SDP is much shorter than the length of the whole sentence, which reduce noise caused by many other entities; 2) SDP emphasizes more on syntactic structures, which are critical to classification; and 3) dependency relation type represents valuable syntactic relation information between the two neighboring words in the SDP.

CNN and RNN have their advantage, but some recent work has looked at combining CNNs and RNNs to utilize the advantages of these two networks simultaneously. Tang, et al. [69] used a hierarchical attention-based convolutional LSTM (ConvLSTM) model. First, they constructed a sentence as a multi-dimensional hierarchical sequence, then they apply a ConvLSTM network and directly learned local and global context information. Finally, a hierarchical attentive pooling strategy is built to capture the parts of a sentence relevant to the target semantic relation so that the softmax classifier could better classify relations. Here, each word in the sentence is first represented by word itself and word position features. Wei, et al. [70] proposed a model combining CNN and RNN; they utilized word and position embeddings for vector representation. Also, they used SVM as a baseline method to compare the performance of the models; here they utilized Local context and semantic features as features [71].

Chika, et al. [72] proposed a hybrid approach using rule-based deep-learning based techniques. They proposed a deep learning approach that utilizes both word-level and sentence-level representations to extract the relationships between entities. Since deep learning techniques demand a large amount of data for training, they proposed a rule-based approach for classes with fewer samples. They utilized POS tag sequence, point-wise mutual information (PMI) [73], and assertion of the sentence for sentence-level feature extraction. In addition, they utilized pattern of sentence/phrase

and shortest dependency path features for the rule-based extraction. The representations were appended to an embedding layer before being fed into a BiLSTM for classification.

Li, et al. [74] evaluated the effectiveness of advanced deep learning models through four deep learning models with increased complexity for single-domain and multi-domain RE. First, they extracted the following features in the sentence level: first word of the sentence, POS of the word, relative positions from this word to the entities, and they utilized CNNs to transform the feature sequence into a fixed-length representation. Next, they extracted the following features in the instance-level: words and types of both entities, the distance between the target entities, and the number of the entities between the target entities. For the single-domain RE, they built a base model with multilayer perceptron (MLP) and compared it to the model using CapNet [75] and fully shared (FS), shared-private (SP), and adversarial training (ADV) modes [76, 77] for multi-domain RE. The CapNet is a group of neurons that can learn the hierarchical relationships between entities. Next, they tried multi-domain RE to utilize the data from different domains for mutual benefit. In the FS and SP modes, the shared feature extractors learn more domain-specific knowledge but the shared feature extractor of the ADV mode learn less domain-specific knowledge but more shared knowledge. They concluded that CapNet does not achieve better performance than the MLP model, and performance can be improved by adding the data from a different domain.

Recently emerged transformer-based models created a tremendous impact in NLP-based research and many researchers started investigating the novel language models for the clinical RE [78]. Pre-trained contextualized language models have been shown to increase the performance of several NLP tasks, especially BERT. BERT has given state-of-the-art performance for NLP systems in the last few years. Current ap-

proaches primarily utilize BERT-based contextualized embeddings into a single dense feed-forward classification layer with a soft-max output layer for classification. Several BERT-based contextualized embeddings have been trained and fine-tuned on various corpora. Alimova, et al. [79] conducted a comparison between three BERT-based models: BERT-uncased [10], BioBERT [80], and Clinical BERT [81] for extracting relations from clinical texts.

Alimova, et al. [79] proposed a machine learning model with a set of manual engineered features, and for comparison purposes, they utilized three BERT-based models: BERT-uncased, BioBERT, and Clinical BERT. They divided the features for into four categories: (i) distance-based: word distance, char distance, sentence distance, punctuation, position; (ii) word-based: BOW, bag of entities (BOE), entity types; (iii) embedding: entities embeddings, concept embeddings, sentence embedding, similarity; and (iv) knowledge-based: UMLS concept types (UMLS) [5], MeSH concept types (mesh) [6], Occurrence in Food and Drug Administration (FDA) clinical trials [7], Occurrence in biomedical literature. They used BioSentVec embedding [82] for the vector representation. They utilized the entity texts combined with a context between them as an input for the BERT-based models as some results lead to the conclusion that the context between entities plays a crucial role in relation detection. Wei, et al. [83] developed two BERT-based models, Fine-Tuned BERT (FT-BERT) and Feature Combined BERT (FC-BERT), each had a linear classification layer on the top of BERT to predict the relation. For FT-BERT, they represent a candidate relation pair in an input sentence by replacing the entity with its semantic category, and for FC-BERT, they represented the entities using their corresponding entity la-

[5]https://www.nlm.nih.gov/research/umls/index.html

[6]https://www.nlm.nih.gov/mesh/meshhome.html

[7]https://www.fda.gov/

bels. They compared the performance of their models with a CNN-RNN model and a two-step pipeline joint model from their previous work [70] and showed the BERT models outperformed both the baseline models.

Most works focus on the inter-sentential relations in the clinical text but Li, et al. [84] focused on both intra- and inter-sentential semantic relations. They proposed a sequence labeling-based method named Bio-Seq, where multiple specified feature extractors extend the sequence labeling framework to perform feature extractions at different levels, especially at the inter-sentential level. The framework consists of two feature extractors: 1) a document-level feature extractor (DE) where they generate word representations from an entire document, and 2) a hierarchical feature extractor (HE) where one Bi-LSTM generates word representations at the sentence level and the other subsequently concatenates all the word representations into a sequence and enables cross-sentence connections for the word representations. The concatenated word representations generated by the two extractors are then fed into a CRF layer for classification.

Many works explored BERT-based models extensively but not other transformer-based models such as RoBERTa [85] and XLNet [86]. Therefore, Yang, et al. [78] systematically explored three transformer-based models such as BERT, RoBERTa, and XLNet for clinical RE. The transformer language models learn representations, and they utilized two classification strategies (binary vs. multi-class classification). They concluded that the transformer-based models achieved better performance in general, and Specifically, RoBERTa and XLNet achieved the best performance on the 2018 MADE1.0 dataset and n2c2-2018 dataset, respectively.

Recently the success of the pre-trained word embeddings such as Word2Vec and BERT have decreased the usage of frequently used traditional NLP features. Therefore, Hasan, et al. [87] investigated whether traditional NLP features can be combined

with word and sentence embeddings to improve relation extraction. They have explored diverse feature sets and different neural network architectures and evaluated the clinical dataset. Following are the features they utilized: word embeddings, POS embeddings, IOB encoding (format to denote chunks), relative distance, concept embeddings, and dependency tree. In addition, they explored the features using CNN, Bi-LSTM, ResNet [88], and GCN. ResNet provides the benefit of CNN while reducing vanishing gradient problems in deep networks. From the results, they concluded that Bi-LSTM model with static Word2Vec embedding plus traditional syntactic and semantic features is most effective for clinical relation extraction.

Some works combine rule-based or machine-based approaches with deep-learning approaches and propose hybrid models to increase the performance of the models. Ju, et al. [89] designed a neural model to tackle both nested (entities embedded in other entities) and polysemous entities (entities annotated with multiple semantic types). To represent rare and unknown words in entities, they tokenized the words into finer-grained subwords and combined all the models to boost the performance. Also, they implemented a feature-based CRF model and created an ensemble to combine its predictions with those of the neural model. The feature-based model used three groups of features: token-based features, dictionary features, and cluster features. Token-based features consists of orthographic (word shape), lexical (derived using GENIA tagger [90]), and syntactic (derived using GENIA tagger [90]) information. The NN-based model extracted both nested and flat (non-nested) entities without using any linguistic features or external knowledge.

Table 3 provides a summary of the works that utilized deep-learning based approaches for relation extraction in the clinical domain. The table shows the year, authors, state-of-the-art datasets used, DL algorithms utilized, and the features used in the algorithms.

Table 2. Overview of the works that utilized deep-learning based approaches for relation extraction in the clinical domain.

Year	**Paper**	**Dataset**	**DL Algorithm**	**Features**
2016	Sahu, et al. [59]	i2b2-2010	CNN	syntactic
2017	Lv, et al. [63]	i2b2-2010	CRF, Auto encoder	syntactic
2017	Luo, et al. [60]	i2b2-2010	CNN	word2vec embedding
2017	Luo, et al. [65]	i2b2-2010	LSTM	word embedding
2018	Chika, et al. [72]	i2b2-2010	Bi-LSTM	syntactic, SDP, word embeddings
2019	Ningthoujam, et al. [67]	i2b2-2010	Bi-LSTM	SDP generated by the dependency parser
2019	Tang, et al. [69]	i2b2-2010	CNN, LSTM	word, position embeddings
2019	He, et al. [62]	i2b2-2010	CNN	concept pair position embeddings
2019	Chauhan, et al. [61]	i2b2-2010	CNN	word, position embeddings
2019	Wei, et al. [83]	i2b2-2010, n2c2-2018	NN	BERT embeddings
2019	Li, et al. [68]	i2b2-2010	CNN, Bi-LSTM	syntactic
2019	Li, et al. [74]	i2b2-2009, MADE1.0, MedEx	MLP, CapNet [75]	orthographic, syntactic
2020	Christopoulou, et al. [66]	n2c2-2018	Bi-LSTM	word, position embeddings
2020	Alimova, et al. [79]	n2c2-2018, MADE-2018	Random Forest, transformers	distance, word, embedding, knowledge
2020	Wei, et al. [70]	n2c2-2018	SVM, CNN-RNN	local context, semantic, word, position embeddings
2020	Ju, et al. [89]	n2c2-2018	CRF, NN	token-based, dictionary, cluster
2020	Hasan, et al. [87]	i2b2-2010	CNN, Bi-LSTM, ResNet, GCN	text embeddings, syntactic, semantic, surface
2020	Li, et al. [84]	n2c2-2018	Bi-LSTM, CRF	word embeddings
2021	Yang, et al. [78]	n2c2-2018, MADE1.0	BERT, RoBERTa, and XLNet	word embeddings, BERT embeddings

3.4 BERT-based Approaches

Pre-trained contextualized language models have been shown to increase the performance of several NLP tasks, especially BERT. As a result, BERT has given state-of-the-art performance for NLP systems in the last few years. In this dissertation, we separate them from deep learning-based approaches to distinguish them. Current approaches primarily utilize BERT-based contextualized embeddings into a single dense feed-forward classification layer with a soft-max output layer for classification.

There are a number of BERT-based contextualized embeddings that have been trained and fine-tuned on various clinical datasets. Alimova, et al. [79] conducted a comparison between three BERT-based models: BERT-uncased [10], BioBERT [80], and Clinical BERT [81] for extracting relations from clinical texts. They utilized the entity texts combined with a context between them as an input into the models. Chauhan, et al. [61] built a unifying framework across multiple domains using datasets from the general, biomedical, and clinical domains to allow for the system to be extendable to new datasets. They studied each dataset by systematically exploring modeling, pre-processing, and training methodologies to determine which of these three was the larger contributor to performance. They found that choices of pre-processing are a large contributor to performance and that omission of such information can further hinder fair comparison. Recent work has looked at combining rule-based and BERT-based approaches. Wei, et al. [83] developed two BERT-based models, Fine-Tuned BERT (FT-BERT) and Feature Combined BERT (FC-BERT), each had a linear classification layer on the top of BERT to predict the relation. For FT-BERT, they represent a candidate relation pair in an input sentence by replacing the entity with its semantic category, and for FC-BERT, they represent the entities

using their corresponding entity labels. Roy, et al. [91] examined different techniques to add medical knowledge from UMLS into a pre-trained BERT model for RE. First, they identified UMLS concepts in clinical notes, and then they used a subset of the Metathesauras and the complete semantic network to create a knowledge graph. Using the knowledge graph, they created UMLS knowledge graph embeddings. Then they combined the knowledge graph embedding with the text embeddings from ClinicalBERT and fed them to the relation classifier.

BERT-based approaches have been explored with chemical datasets recently. Copara, et al [92] used a BERT-based method assessing five variations of the BERT language models, including a domain-specific model called ChemBERTa. The models have a fully connected layer on top of the hidden states of each token and are fine-tuned on the CLEF-2020 dataset. Zhang, et al. [93] proposed a hybrid combination of deep learning models and pattern-based rules. In their work, a new language model, named Patent_BioBERT, was generated by pre-training the patent texts over BioBERT [80]. They built a binary classifier by fine-tuning Patent-BioBERT to recognize relations between the trigger words and entities. They also designed post-processing rules based on patterns observed in the training data and applied them to recover some false-negative relations.

Lai, et al. [94] proposed a novel architecture that combines BERT with Graph Transformer (BERT-GT) by integrating a neighbor–attention mechanism into the BERT architecture. The original transformers architectures utilize the whole sentence to calculate the attention of the current token; however, the BERT-GT architecture calculates its attention utilizing only its neighbor tokens. Thus, each token can pay attention to its neighbor's information with little noise. Existing RE models rely on neural networks to extract the semantic information of sentences and ignore the critical role of important phrase information [95]. Xu, et al. [95] proposed a

BERT gated multi-window attention network (BERT-GMAN) for RE. First, it uses BERT to extract the semantic representation features of the sentence and its constraint information. Then, it constructs the key phrases extraction network to obtain multi-granularity phrase information and uses element-wise max pooling to select key phrases features. Third, it adopts a classification feature perception network to filter further and globally perceive key phrase features to form the overall features of relation classification. Finally, it combines with the softmax classifier to perform RE [95]. Cross-sentence n-ary RE detects relations among n entities across multiple sentences. Yi, et al. [96] combined BERT with bidirectional Gated Recurrent Unit (GRU) [97] (BERT-GRU), which exploits pre-trained deep language representations to obtain the latent linguistic information for relation extraction and without using any high-level linguistic resources extracted by NLP tools.

Table 3. Overview of the works that utilized BERT-based approaches for relation extraction in the clinical domain.

Year	**Paper**	**Dataset**	**BERT models**
2019	Chauhan, et al. [61]	i2b2-2010, SemEval-2010 Task 8 [98]	BERT-uncased [10], BioBERT [80], and Clinical BERT [81]
2019	Wei, et al. [83]	i2b2-2010, n2c2-2018	BERT-uncased [10], BioBERT [80], and Clinical BERT [81]
2020	Alimova, et al. [79]	n2c2-2018, MADE-2018	BERT-uncased [10], BioBERT [80], and Clinical BERT [81]
2020	Copara, et al [92]	CLEF-2020	BERT-uncased, BERT-cased [10], ChemBERTa [92]
2020	Zhang, et al. [93]	CLEF-2020	BioBERT [80], Patent-BioBERT [93]
2020	Lai, et al. [94]	n-ary dataset [99], CDR [100, 101]	BERT [10], BlueBERT [102]
2020	Yi, et al [96]	Semeval-2010 Task 8 [98]	BERT-cased [10], BlueBERT [102]
2021	Roy, et al. [91]	i2b2-2010	Clinical BERT [81]
2021	Xu, et al. [95]	Semeval-2010 Task 8 [98]	BERT-uncased [10]

3.5 GCN-based Approaches

Neural networks gained popularity in the past decade, and different variants of simple neural networks achieved success in many research fields. However, most

of these variants deal with euclidean data, while many real-world data are non-euclidean. GNNs deal with non-euclidean graph data that contains rich relational information between elements. GCN-based approaches have been gaining attention recently among the NLP community. Here, we discuss works related to RE and works that inspired us to propose our approaches using GNN/GCN.

Relational reasoning tries to reason about entities and their relations, which are of great importance in many NLP tasks, including RE [103]. Zhu, et al. [103] proposed to generate the parameters of GNNs (GP-GNNs) according to natural language sentences, which enabled GNNs to process relational reasoning on unstructured text inputs. GP-GNN is constructed with entities in the sequence of the text followed by three modules that encode rich information from natural languages, propagate relational information among various nodes, and classify. GCN update the current node features according to the features of its first-order adjacent nodes and edges but not all important nodes are first-order reachable, which leads to multi-layer GCNs for indirect relevance capturing [104]. Zhou, et al. [104] proposed a novel weighted GCN (WGCN) by constructing a Logical Adjacency Matrix (LAM) which effectively solves the feature fusion of multi-hop relation without additional layers and parameters. They added virtual edges to the dependency tree to construct a LAM which can directly figure out k-order neighborhood dependence with only 1-layer WGCN. They also applied an Entity-Attention mechanism to enrich the entity pairs with more focused semantic information for RE.

Chemical-disease relation (CDR) extraction plays an important role in biomedical text mining [105]. Wang, et al. [105] proposed a novel end-to-end neural network based on the GCN and multi-head attention. They constructed a document-level dependency graph for the inter-sentence RE to capture the syntactic dependency information across sentences. The multi-head attention mechanism learns the rel-

atively important context features from different semantic subspaces. Most of the state-of-the-art GCN models are shallow due to the over-smoothing problem [106]. After multi-layer graph convolution, Laplacian smoothing causes node representation toward a space that contains limited distinguished information, which is called an over-smoothing problem. This makes it difficult for the GCN to model the relation between long-distance nodes [106]. Zeng, et al [106] proposed CID-GCN, an effective GCN architecture with a gating mechanism. They constructed a heterogeneous graph that contains mention, sentence, and entity nodes and combined the gating mechanism with the graph convolution operation to address the over-smoothing problem. Finally, the classifier module gives the relation prediction from two graph representations of entity nodes.

Sahu, et al. [107] presented a novel inter-sentence RE model that builds a labeled edge GCN model on a document-level graph. The graph was constructed using various inter and intra-sentence dependencies, and they utilized multi-instance learning with bi-affine pairwise scoring to predict the relation of an entity pair. Zeng, et al. [108] proposed Graph Aggregation-and-Inference Network (GAIN) featuring double graphs. First, they built a heterogeneous mention-level graph (hMG) to model complex interactions among different mentions across the document. After applying GCN, the graph is transformed into an entity-level graph (EG), based on which they proposed a novel path reasoning mechanism to infer relations between entities. Joint entity and relation extraction is an essential task in information extraction, which aims to extract all relational triples from unstructured text [109]. Zhao, et al. [109] proposed a representation, iterative fusion based on heterogeneous GNN for RE (RIFRE). They modeled relations and words as nodes on the graph and updated them through a message-passing mechanism to perform RE. This fuses the semantic information of the relations nodes to the word nodes associated with them, which

helps extract the entities that form valid relations. Document-level RE requires reasoning over multiple sentences across a document, unlike the sentence-level RE [108]. Inter-sentence RE deals with complex semantic relations in documents [107].

Dependency trees capture long-range relations between words, but they may neglect crucial information by pruning the dependency trees too aggressively or are computationally inefficient because it is difficult to parallelize over different tree structures [110]. Therefore, Zhang, et al. [110] proposed a novel extension of GCN that pools information over arbitrary dependency structures efficiently in parallel. They further applied a novel pruning strategy to the input trees to incorporate relevant information while removing irrelevant content. Huang, et al. [111] proposed a knowledge-aware framework to enhance word representations for distantly supervised relation extraction; a piece-wise attention method is used to distinguish the local and global information. They designed a heterogeneous graph structure for each sentence and introduced a heterogeneous GCN with knowledge attention to aggregate the implicit interaction among sentences from a local-to-global perspective. Tian, et al. [112] proposed a dependency-driven approach for RE where an attention mechanism upon GCNs (A-GCN) is applied to different contextual words in the dependency tree obtained from an off-the-shelf dependency parser to distinguish the importance of different word dependencies. They assume that the dependency types among words contain necessary contextual guidance, potentially helpful for RE. Yu, et al. [113] presented a novel architecture named Dynamically Pruned GCN (DP-GCN), which prunes the dependency tree with rethinking in an end-to-end scheme. In each layer of DP-GCN, they employed a selection module to concentrate on nodes expressing the target relation by a set of binary gates and then augmented the pruned tree with a pruned semantic graph to ensure connectivity. Next, they introduced a rethinking mechanism to guide and refine the pruning operation by repeatedly feeding back the

high-level learned features.

One of the major applications of the GNN is node classification, where we train the graph nodes with labels and try and predict the label for a node without ground truth. This has been adapted to perform text classification using graph structures. Yao, et al. [14] utilized a GCN for text classification. First, they built a single text graph based on word co-occurrence and document word relations, then learned a Text GCN for the corpus. Text GCN jointly learns the embeddings for both words and documents, as supervised by the known classes for documents. Huang, et al. [114] proposed a different GNN based method. Instead of building a single corpus level graph, they built a graph for each input text. They connected the word nodes within a relatively small text window rather than all. The representations of the same nodes and weights of edges are shared globally and updated at the text level through a message passing mechanism, where a node takes in the information from neighboring nodes to update its representation. They claimed this removed the dependency burden between a single input text and the entire corpus. Zhang, et al. [115] proposed a novel method for *INductive word representations* via GNN, termed TextING. They built individual graphs for each document first, then used GNN to learn the fine-grained word representations based on their local structures, effectively producing embeddings for unseen words in the new document. Finally, the word nodes are incorporated as the document embedding. Lu, et al. [116] proposed a model which combines the strengths of BERT with a Vocabulary VGCN in the same model. While learning the classifier, the word embedding and graph embedding interacted through the self-attention mechanism.

Table 4. Overview of the works that utilized graph convolutional network (GCN)-based approaches for relation extraction.

Year	**Paper**	**Dataset**	**NLP problem**	**Algorithms**
2018	Zhang, et al. [110]	TACRED [a]	long-range RE	GCN, pruned dependency trees
2018	Huang, et al. [111]	Riedel-2010 [117], GIDS [118]	distantly supervised RE	GCN,piecewise self-attention mechanism
2019	Sahu, et al. [107]	CDR [100], CHR [107]	relational reasoning	GCN
2019	Zhu, et al. [103]	Wikipedia corpora [119]	relational reasoning	GCN
2020	Zhou, et al. [104]	TACRED, SemEval-2010	long-range RE	GCN, pruned dependency trees
2020	Yu, et al. [113]	TACRED, SemEval-2010	long-range RE	GCN, pruned dependency trees
2020	Wang, et al. [105]	CDR corpus [100]	document-level RE	GCN, Bi-LSTM, attention mechanism
2020	Zeng, et al. [108]	DocRED [120]	document-level RE	GCN
2021	Zhao, et al. [109]	WebNLG [b]	Joint entity and RE	GCN
2021	Zeng, et al [106]	CDR corpus [100]	document-level RE	GCN, BiLSTM
2021	Tian, et al. [112]	ACE05 [c], SemEval-2010 [d]	dependency-driven RE	GCN, attention mechanism

[a] https://catalog.ldc.upenn.edu/LDC2018T24

[b] https://gitlab.com/shimorina/webnlg-dataset

[c] https://catalog.ldc.upenn.edu/LDC2006T06

[d] https://www.kaggle.com/datasets/drtoshi/semeval2010-task-8-dataset

3.6 Discussion

Table 5 provides an overview of the different feature representations and algorithms that have been used for RE across the years. For feature representations, *Static* refers to the static word representations, *Embeddings* refers to the word embeddings representations, and *Contextualized* refers to the contextualized embedding representations. For the approach, *Rule, ML, DL, BERT* and *GCN* refer to rule-based, machine learning-based, deep learning-based, BERT-based approaches, and GCN-based approaches respectively.

Also we have mentioned our works related to this dissertation at the end of the table.

Table 5. Overview of related work

		Feature Representation				Approaches				
Year	**Paper**	**Dictionary**	**Traditional**	**Static**	**Contextual**	**Rule**	**ML**	**DL**	**BERT**	**GCN**
2015	Li, et al. [34]	X				X				
2020	He, et al. [35]	X				X				
2020	Lowe, et al. [36]		X			X				
2019	Miller, et al. [45]		X				X			
2010	Patrick, et al. [41]		X				X			
2010	Fushman, et al. [43]		X				X			
2011	Rink, et al. [46]		X				X			
2013	Zhu, et al. [47]		X				X			
2010	De Bruijn, et al. [48]		X				X			
2010	Grouin, et al. [42]		X			X	X			
2015	Porumb, et al. [53]		X			X	X			
2020	Yang, et al. [55]	X				X	X			
2017	Luo, et al. [60]			X				X		
2019	Sahu, et al. [107]			X				X		
2020	Christopoulou, et al. [66]			X				X		
2019	Ningthoujam, et al. [67]			X				X		
2018	Chika, et al. [72]		X	X		X		X		
2019	Li, et al. [121]		X	X				X		
2019	Tang, et al. [69]			X				X		
2020	Alimova, et al. [79]				X				X	
2019	Chauhan, et al. [61]				X				X	
2019	Wei, et al. [83]				X				X	
2018	Zhang, et al. [110]			X						
2018	Huang, et al. [111]				X					X
2019	Sahu, et al. [107]			X						X
2019	Zhu, et al. [103]			X						X
2020	Zhou, et al. [104]				X					X
2020	Yu, et al. [113]			X						X
2020	Wang, et al. [105]				X					X
2020	Zeng, et al. [108]			X						X
2021	Zhao, et al. [109]			X						X
2021	Zeng, et al [106]			X						X
2021	Tian, et al. [112]				X					X
2018	Mahendran, et al. [122]		X				X			
2020	Mahendran, et al. [123]			X				X		
2020	Mahendran, et al. [124]			X				X		
2021	Mahendran, et al. [125]			X	X	X		X	X	
2021	Mahendran, et al. [126]			X	X	X		X	X	
2021	Mahendran, et al. [127]				X				X	
2022	Mahendran, et al. [128]			X	X					X

CHAPTER 4

DATA

In this chapter, we discuss all the datasets used in our work. Each dataset was introduced as a part of a RE challenge, therefore, we discuss the challenge and statistics of each data below.

4.1 Informatics for Integrating Biology & the Bedside (i2b2-2010)

4.1.1 Challenge

In 2010, Informatics for Integrating Biology & the Bedside (i2b2) [129] workshop on NLP partnered with VA Salt Lake City Health Care System to manually annotating patient reports from three institutions. They presented three tasks: a concept extraction task, an assertion classification task, and a relation classification task. The relation classification task focused on assigning relations that hold between the medical entities: *Problems*, *Tests*, and *Treatments*. A total of 394 training reports, 477 test reports, and 877 unannotated reports were de-identified and released to challenge participants with data use agreements [129]. Sixteen teams participated in this task and their developed systems showed that machine learning approaches combined with rule-based approaches performed well with the relation classification.

4.1.2 Data

The i2b2-2010 corpus includes problem-related attributes and relations from 426 patient discharge summaries. It was manually annotated by medical practitioners to identify three types of entities and eight relations among them. Relations build

on the entities: *Problem*, *Treatment*, and *Test*. The relations can be mainly divided into three categories: Problem-Treatment (Tr-P) relations, Problem-Test (Te-P) relations, and Problem-Problem (P-P) relations. Each category is further divided into relations, as shown in Table 7. The relations are treatment caused medical problems (TrCP), treatment administered medical problem (TrAP), treatment worsens medical problem (TrWP), treatment improve or cure medical problem (TrIP), treatment was not administered because of medical problem (TrNAP), test reveal medical problem (TeRP), test conducted to investigate medical problem (TeCP), medical problem indicates medical problems (PIP). Fig. 5 shows an example of sentence from the dataset containing three entities and three possible relations between them.

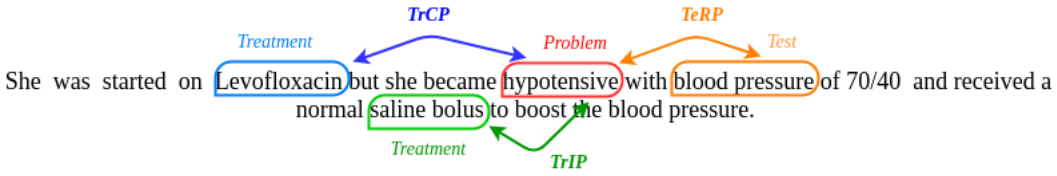

Fig. 5. An example of a sentence from the i2b2-2010 dataset.

The relation between entities are bounded by sentences; the text within the same sentence determines the type of relationship that exists between two concepts. However, two entities can exist in the same sentence and may not have a relation between them. These relations are considered as no-relation pairs, and each relation category includes a no-relation class as follows: no relation between treatment and problem (NTrP), no relation between test and problem (NTeP), and no relation between the two problems (NPP). Table 7 shows the relation statistics of the dataset. i2b2-2010 relation corpus has multiple pairs for a relation.

Table 6. Relation type statistics of i2b2-2010 dataset

Category	Relation	# Train	# Test
Treatment-Problem (Tr-P)	Treatment improves problem (TrIP)	51	152
	Treatment worsens problem (TrWP)	24	109
	Treatment causes problem (TrCP)	184	342
	Treatment is administered for problem (TrAP)	885	1732
	Treatment is not administered because of problem (TrNAP)	62	112
	No relation between treatment and problem (NTrP)	1702	2759
Test-Problem (Te-P)	Test reveals problem (TeRP)	993	2060
	Test conducted to investigate problem (TeCP)	166	338
	No relation between test and problem (NTeP)	993	1974
Problem-Problem (P-P)	Medical problem indicates problem (PIP)	775	1448
	No relation between the two problems (NPP)	4418	8089

4.2 National NLP Clinical Challenges (n2c2-2018)

4.2.1 Challenge

In 2018, National NLP Clinical Challenges (n2c2) [130] was organized to continue the legacy of i2b2. They presented two shared tasks: Cohort selection for clinical trials (Track 1) and Adverse Drug Events (ADE) and Medication Extraction (Track 2). Track 2 required the extraction of medications and associate them with their prescription information and any ADEs from clinical narratives and was evaluated over three tasks: concept extraction, relation classification, and end-to-end systems. Relation classification task focuses on linking the previously mentioned concepts to their medication by identifying relations on gold standard concepts. Ten teams participated in the relation classification task with systems built using deep learning-based, traditional machine learning-based, and rule-based methods.

4.2.2 Data

The n2c2-2018 Adverse Drug Event (ADE) Dataset [130] contains ADE mentions, drug-related attributes, and drug-related relations from 505 patient discharge

Table 7. Relation type statistics of n2c2-2018 dataset.

Relation	**# Train**	**# Test**
Strength-Drug	6702	4244
Duration-Drug	643	426
Route-Drug	5538	3546
Form-Drug	6654	4374
ADE-Drug	1107	733
Dosage-Drug	4225	2695
Reason-Drug	5169	3410
Frequency-Drug	6310	4034

summaries drawn from the MIMIC-III database [131]. It consists of nine entity classes (Drug, Strength, Route, Form, ADE, Dosage, Reason, Frequency) and eight relations between the drug entity and other non-drug entity classes. Fig. 6 shows an example of sentence from the dataset containing a drug entity and two non-drug entities and possible relations between them. The data were split into training and test sets; training set includes 303 annotated files and test set includes 202 annotated files. The class distributions for both concepts and relations are very similar for the test, training, and full datasets. The class distributions for both concepts and relations are very similar for the test and training sets. Table 7 shows the number of relations in the training and test data.

4.3 Cheminformatics Elsevier Melbourne University - Conference and Labs of the Evaluation Forum (ChEMU-CLEF 2020)

4.3.1 Challenge

In 2020, the Cheminformatics Elsevier Melbourne University (ChEMU) evaluation lab which is part of the Conference and Labs of the Evaluation Forum (CLEF-

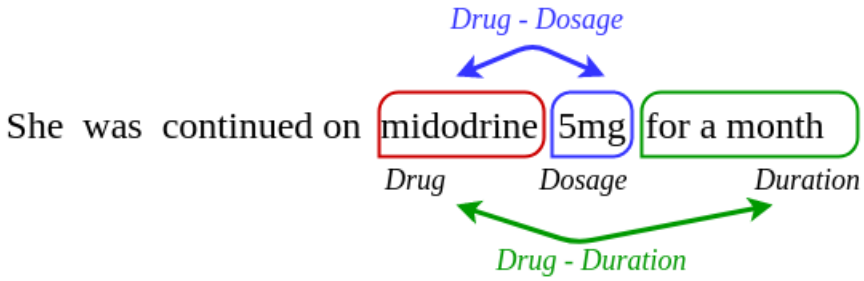

Fig. 6. An example of a sentence from the N2C2-2018 dataset

2020), provided a platform to develop automated information extraction methods over chemical patents [35]. The CLEF-2020 ChEMU [132] challenge was to identify chemical entities and events that explain the sequence of steps that lead from a chemical reaction to an end product. They presented three tasks: chemical named entity recognition (NER), event extraction (EE). The challenge attracted 37 registrants from 13 countries.

4.3.2 Data

The CLEF-2020 data corpus [132] contains 1500 chemical snippets sampled from 170 English document patents from the European Patent Office and the United States Patent and Trademark Office [132]. Each snippet holds a detailed description of chemical reactions.

Entities of this dataset are divided into four categories [35]: (1) chemical compounds that are involved in a chemical reaction; (2) conditions under which a chemical reaction is carried out; (3) yields obtained for the final chemical product; and (4) example labels that are associated with reaction specifications. The four categories are further divided into a total of ten entity types. The compound category defines five roles a chemical compound can play within a chemical reaction. Conditions category and yield category each include two entity types. Fig. 7 summarizes the labels in

each category.

A chemical reaction step involves action and one or more chemical compounds on which the action takes effect [35]. The action is also linked to the conditions under which the action is carried out and the resultant yields from the action. Relations form between actions (trigger words) and all arguments involved in the reaction steps, such as chemical compounds, conditions, and yields. The ARG1 event label corresponds to relations between a trigger word and chemical compound entities. The ARGM event label corresponds to the relations between a trigger word and temperature, time, or yield entities. Table 8 shows the definitions of the entity types, trigger words, and relation types.

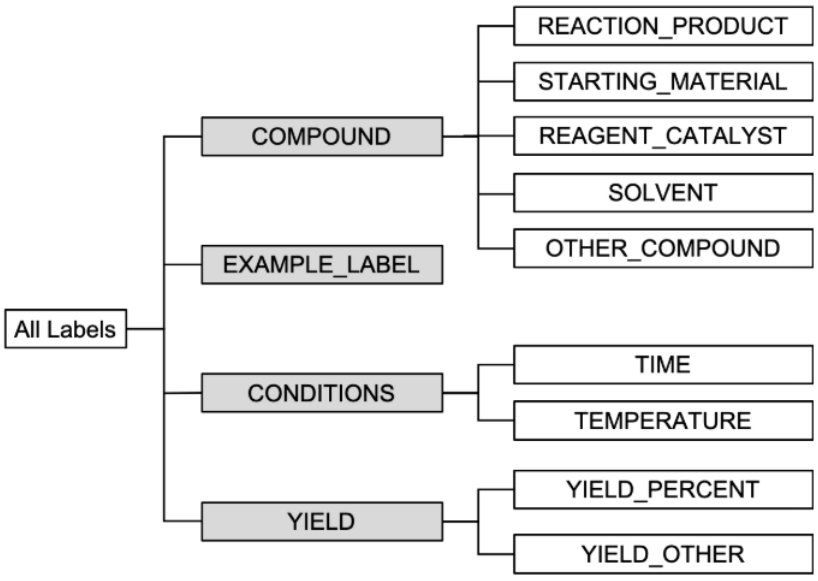

Fig. 7. An illustration of the hierarchical structure of the entity labels of the CLEF-2020 dataset [35]

The corpus includes ten entity types under four categories, two classes of trigger words (REACTION_STEP, WORKUP) and two relation classes (ARG1, ARGM) and Table 9 shows the statistics of the dataset. Fig. 8 shows an example of a Brat Rapid Annotation Tool (BRAT) annotated sentence from the dataset containing chemical

Table 8. Definitions of entity types, trigger words, and relation types of CLEF-2020 dataset [35]

Entity Type	**Definition**
REACTION_PRODUCT (R.P.)	A product is a substance that is formed during a chemical reaction.
STARTING_MATERIAL (S.M.)	A substance that is consumed in the course of a chemical reaction providing atoms to products is considered as starting material.
REAGENT_CATALYST (R.C.)	A reagent is a compound added to a system to cause or help with a chemical reaction. Compounds like catalysts, bases to remove protons or acids to add protons must be also annotated with this tag.
SOLVENT (S)	A solvent is a chemical entity that dissolves a solute resulting in a solution.
OTHER_COMPOUND (O.C.)	Other chemical compounds that are not the products, starting materials, reagents, catalysts and solvents.
TIME	The reaction time of the reaction.
TEMPERATURE (Temp)	The temperature of the reaction.
YIELD_PERCENT (Y.P.)	Yields given in percent values.
YIELD_OTHER (Y.O.)	Yields provided in other units than %.
WORKUP	A manipulation required to isolate and purify the product of a chemical reaction
REACTION STEP	An event that converts starting materials into a product
Arg1	The elation between an event trigger word and a chemical compound
ARGM	The relation between an event trigger word and a temperature, time, or yield entity

Table 9. Entity and relation type statistics of the training set of CLEF-2020 dataset

Events	**Entities**	**Instances**	**REACTION_STEP**	**WORKUP**
ARG1	EXAMPLE_LABEL	886	-	-
	REACTION_PRODUCT	2052	1101	11
	STARTING_MATERIAL	1754	1747	4
	REAGENT_CATALYST	1281	1272	-
	SOLVENT	1140	1134	4
	OTHER_COMPOUND	4640	161	4097
ARGM	YIELD_PERCENT	955	937	1
	YIELD_OTHER	1061	1043	2
	TIME	1059	839	81
	TEMPERATURE	1515	813	242
Triggers	REACTION_STEP	3815		
	WORKUP	3053		

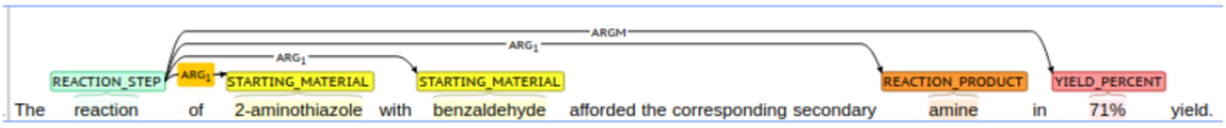

Fig. 8. An example of a BRAT annotated sentence from the CLEF-2020 dataset

entities and relations among the entities.

4.4 DrugProt

4.4.1 Challenge

BioCreative challenges and workshops are a community-wide effort for evaluating text mining and information extraction systems applied to the biological domain. In 2021, they introduced the BioCreative VII challenge which focused on the detection of chemicals, drugs and related substances with five tracks. Track 1 explores recognition of chemical-protein entity relations from abstracts. Aim of this track is to promote the development and evaluation of systems that are able to automatically detect relations between chemical compounds/drug and genes/proteins.

4.4.2 Data

BioCreative VII Track 1 released a manually annotated corpus, the DrugProt corpus which contains chemical and gene mentions, and all binary relationships between them. A range of different types chemical-protein/gene interactions are of key relevance for biology, including metabolic relations (e.g. substrates, products) inhibition, binding or induction associations [133]. The training set contains chemical mentions (46274), gene/protein mentions (43255), and drug/chemical-protein/gene interactions (17288) from 3500 PubMed abstracts. The development and test set includes 750 and 10750 abstracts, respectively. Fig. 9 shows the BRAT annotation of

the entities and relations of a sentence from the dataset. Table 10 shows the number of instances for each relation type in the training and development datasets.

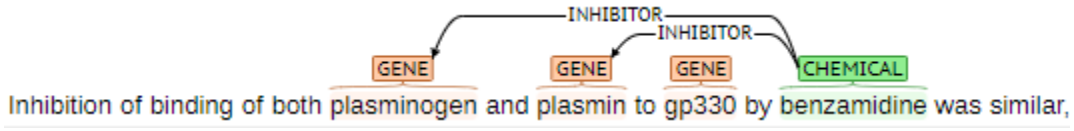

Fig. 9. An example of a BRAT annotated sentence from the DrugProt dataset

Table 10. Relation type statistics of DrugProt dataset

	# Train	**# Development**
INDIRECT-DOWNREGULATOR	1330	332
INDIRECT-UPREGULATOR	1379	302
DIRECT-REGULATOR	2250	458
ACTIVATOR	1429	246
INHIBITOR	5392	1152
AGONIST	659	131
AGONIST-ACTIVATOR	29	10
AGONIST-INHIBITOR	13	2
ANTAGONIST	972	218
PRODUCT-OF	921	158
SUBSTRATE	2003	495
SUBSTRATE_PRODUCT-OF	25	3
PART-OF	886	258
TOTAL	17288	3765

CHAPTER 5

EXPERIMENTAL FRAMEWORK –RelEx

RelEx is an RE framework we developed to identify relations between two entities. The framework is divided into five main approaches: rule-based (co-location), machine learning-based, CNN-based, BERT-based, and GCN-based approaches. The framework is designed to consider two entities in an instance and determine whether a relation exists between the entities. In our case, an instance consists of a sentence, and we use the terms interchangeably. Here, we describe in detail the five approaches we developed for RE: 1) rule-based, 2) machine learning-based, 3) CNN-based, 4) BERT-based, and 5) GCN-based approaches.

RelEx can be found here[1] for reproducibility.

5.1 Rule-based Approach

Our rule-based approach [123, 134] determines whether a relation exists between two entities utilizing the co-location information between them. We use a Breadth-First Search algorithm (BFS) to find the selected entity's closest occurrence on either side of other entities. For each entity, we traverse both sides until the selected entity's closest occurrence is found based on the provided span values of the entities. We explore four traversal mechanisms and report the best traversal mechanism in our results:

1. traverse left-only
2. traverse right-only

[1]https://github.com/NLPatVCU/RelEx

3. traverse left-first-then-right
4. traverse right-first-then-left

This was conducted in two modes:

1. bounded - limiting the traversal to only a single relation per relation type.
2. unbounded - allowing for a drug to be linked to multiple entity classes with the same relation.

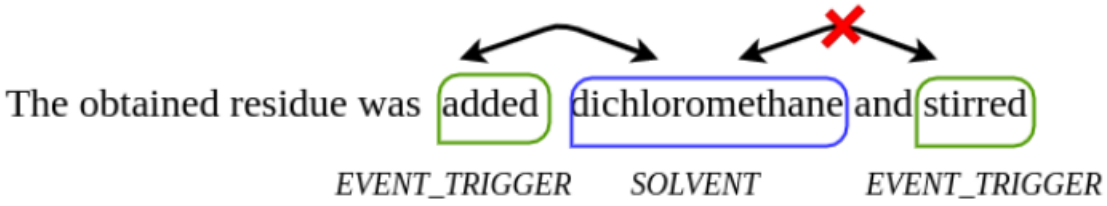

Fig. 10. Example that depicts how left-only traversal works

For example, Fig. 10 shows a sample sentence from CLEF-2020 dataset from the chemical domain. Sentence *The obtained residue was added dichloromethane and stirred.* contains a SOLVENT, *dichloromethane* and two EVENT_TRIGGERS *added* and *stirred*. The SOLVENT has a relation with the closest EVENT_TRIGGER occurrence *added* but not with *stirred* when applying the left-only traversal mechanism.

5.2 Machine Learning-based Approach

Our machine learning-based approach [122] presents a feature-vector based, supervised machine learning approach to extract explicit semantic relations and classify them. Our approach takes sentences as the input, defines a set of features, and combines them into a feature vector to train a machine learning model which is the most crucial part of our approach. The idea is to decrease the size of the effective vocabulary, which would increase the classification accuracy by eliminating the noise in the

feature representation [18]. Relations between entities are extracted and classified through this learned system. Fig. 11 shows the pipeline of our approach, and each step of the approach is discussed in detail in the following sections.

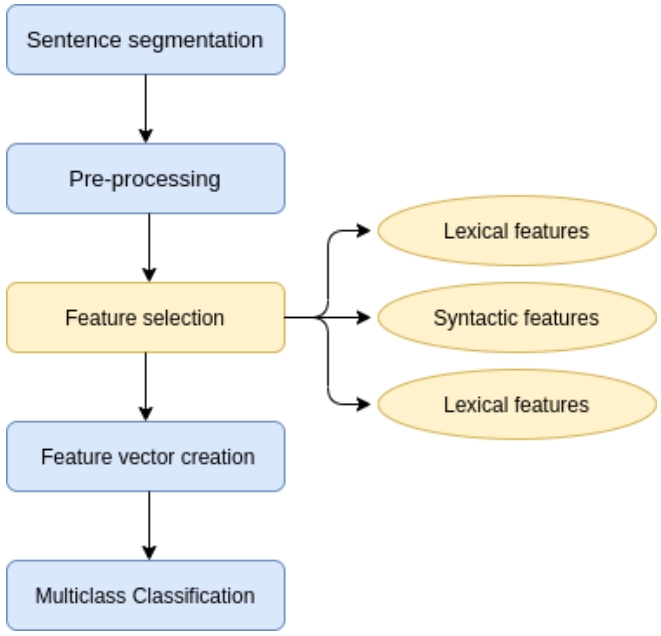

Fig. 11. Pipeline that highlights the main steps of our machine learning-based approach

5.2.1 Preprocessing

All sentences in the abstracts are preprocessed to normalize the text so that the input text is guaranteed to be consistent and feature extraction/classification is simplified. Preprocessing is performed using Natural Language Toolkit's (NLTK)[2] Tokenizers, POS tagger, and Porter Stemmer are used in text preprocessing to extract the following features for each entity:

1. token of the entity

[2]https://www.nltk.org/

2. convert text to lower case
3. removal of special characters
4. stemming

A separate set of steps are followed, where each feature is computed. Some features are extracted in two different scenarios: before removing the stop words and after removing the stop words. Stop words are the most common words of the language that do not contribute to the semantics of the documents or contain any significance but has a high frequency. Filtering out such words reduces the size of the feature space with the idea that the noise occurs with will also be reduced.

5.2.2 Feature selection

After preprocessing the input text, a subset of words that contain the respective entity pair are selected from each sentence, a set of features are computed and a feature vector is created by combining the computed features. Here, we explored variations of the following feature types described in detail in Section 2: 1) bigrams, 2) collocations, 3) BOW, 4) POS, 5) TF-IDF.

The specific features explored in our approach are listed below where E1 refers to the first entity, and E2 refers to the second entity:

1. Number of words before E1 with/without stop words
2. Number of words after E2 with/without stop words
3. Word before E1
4. Word after E2
5. POS of the words before E1 with/without stop words
6. POS of the words after E1 with/without stop words
7. POS of the words before E2 with/without stop words
8. POS of the words after E2 with/without stop words

9. Bigram of the first word before E1 with/without stop words
10. Bigram of the first word after E2 with/without stop words
11. Bigram of E1
12. Bigram of E2
13. Highest bigram value of words in between entities with/without stop words
14. Number of unique POS types in between the entities with/without stop words
15. Number of unique POS types before E1 with/without stop words
16. Number of unique POS types after E2 with/without stop words
17. POS type of the word with the highest TF-IDF score in between the entities
18. POS type of the word with highest TF-IDF score in before E1
19. POS type of the word with highest TF-IDF score in after E2

5.2.3 Classification

In the final step of our approach, a feature vector is generated for each sentence by incorporating the extracted features in the previous step. The generated feature vector is then used to train a classifier that classifies the relation between the entities. The following classifiers, which represent three main classification algorithms, are available to train and evaluate the data set in our approach:[3] Decision Trees, Naive Bayes (NB), and SVMs. The resulting model is then used to classify the relations.

5.3 CNN-based Approach

Our CNN-based approach includes three CNN architectures. Two baselines previously proposed by Luo et al. [60] and our novel extension [123, 134, 135] that utilizes a multi-label architecture. In this section, we describe each of the architectures in detail.

[3]NLTK sci-kit learn library classifiers are used.

5.3.1 Single-label Sentence-CNN

In this architecture, each relation consists of a pair of entities. The Sentence-CNN learns the relation representation for the entire sentence as a whole. First, we identify the sentence where each relation is located and extract the sentence and we feed it into a CNN for learning. Second, we apply the convolution layer on the sentence and learn the local features from the embedding vectors obtained from each word of the sentence. 1D convolutions compute the weighted sum of the features and select a certain combination of features. Next, we apply the max-pooling layer [136] to extract the most important feature from the entire sentence which is called the global feature as it is considered over the entire sentence. The max-pooling layer also helps in reducing the dimensionality of the input and discarding the features that do not contribute to the classification. Then we unstack the volume into a flat vector before feeding it into the fully connected feed-forward layer. Finally, the fixed-length vector is fed into a softmax (fully-connected) layer to perform the classification. Classification error is then back-propagated ,and the model is re-trained until the error is minimized. The weights of the matrix and bias are the parameters that get tuned until the optimized model is obtained. One of the beneficial properties of CNN is that it preserves the spatial orientation; in this case, the sequence of the words in the sentence.

To step through Fig. 12, let us consider a convolutional filter of size 3 (considers three words at a time) with a set of random weights assigned on the sentence vector matrix. First, the filter overlays across the first two word vectors and performs an element-wise product for all its elements, then it sums them up to obtain one value which gives the first element of the output sequence. Then the filter moves down to the second and third vector and performs the same operation. Filter continues to move

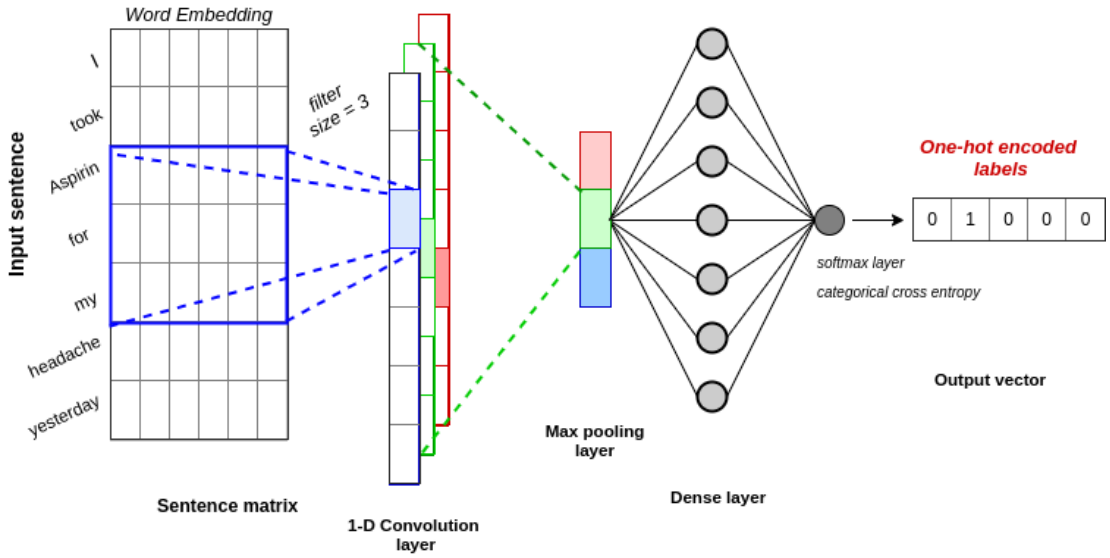

Fig. 12. An illustration of the Sentence-CNN architecture which explains the process of single-label Sentence-CNN

down and perform the operation repeatedly and outputs a vector on which the 1-max pooling function is applied. The max-pooling function extracts the largest element from each vector. This produces a vector of fixed-length that is fed into the softmax layer that performs the classification and backpropagates the error. Convolution units learn the entire sentence at once and predict the classes. A sentence can contain more than one distinct mention of relation (pair of entities) with its context and at that point, this architecture cannot differentiate the sentences that are fed into CNN. To account for this, we propose a new architecture by modifying the existing one to be able to train the model to learn multiple labels for the same sentence.

5.3.2 Multi-label Sentence-CNN

Often in machine learning tasks, there can multiple possible labels for one instance that are not mutually exclusive. This is called a multi-label classification problem. In RE, this happens when an instance can contain more than a single set of

relations between multiple entities. To address this, we modified the Sentence-CNN which predicted a single-label for an instance to predict multiple labels for an instance. As shown in Fig. 13, when the multi-label flag is enabled the system outputs multi-hot encoded labels. The multi-label Sentence-CNN is constructed differently in the following aspects: loss function, choice of the output layer, multi-hot-encoding of labels. Here, we describe each of those aspects in more detail.

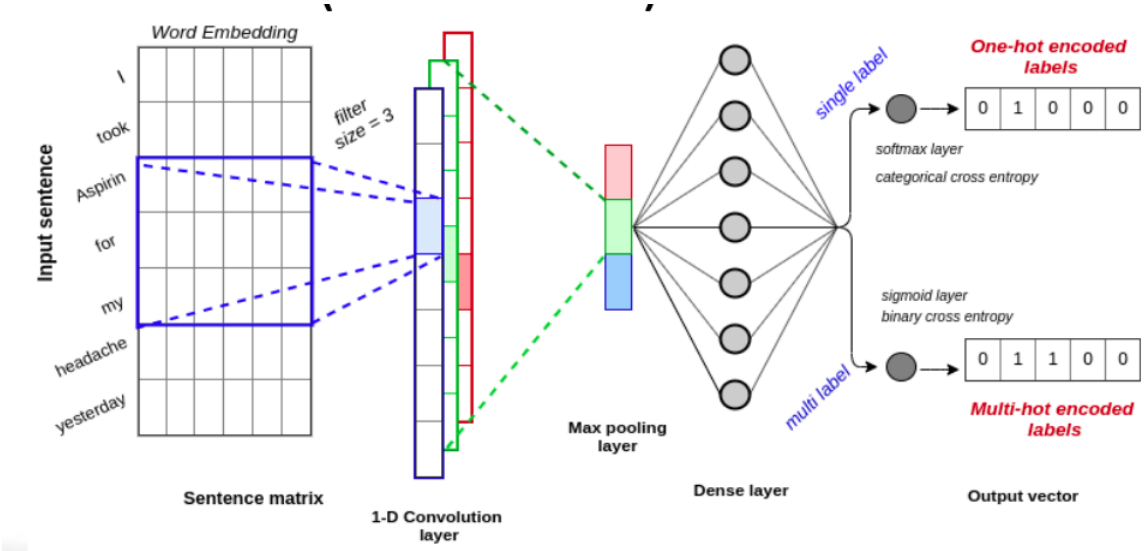

Fig. 13. An illustration of the Sentence-CNN architecture that explains the process of both single-label and multi-label Sentence-CNN

- **Loss function**: Binary cross-entropy is just a special case of categorical cross-entropy. Categorical cross-entropy loss is for multi-class classification where each example belongs to a single class, whereas binary cross-entropy loss is for binary classifications. Since we treat this as the binary classification, we pick binary-cross-entropy and model the output of the network as independent Bernoulli distributions per label.

- **Choice of output layer**: Usually, the softmax layer is chosen for multi-class classification problems and the sigmoid layer for binary classification problems.

The sigmoid activation function at the output layer neural network models the probability of a class as bernoulli distribution. It calculates the conditional probabilities of each target class independent from the other class probabilities and returns the output value, which falls in the range of 0 to 1. The sigmoid function produces the curve, which will be in the Shape S, and a threshold is set to convert the vector into a binary vector. Here, we treat this multi-label classification problem as a binary classification problem for each class in the label. Therefore, we use sigmoid activation instead of softmax with the binary-cross-entropy loss.

- **Multi-hot-encoding of labels**: The system functions as a binary classifier for each class and results in a multi-hot encoded binary vector. The sigmoid activation function in the output layer returns a real-valued output vector with the probabilities of each class. We use the threshold of 0.5 to determine whether that class label is present or not for a particular instance of the inputs.

5.3.3 Segment-CNN

In the Sentence-CNN architecture, each relation is represented by an entire sentence and does not capture the positional information of the entity pairs. Therefore, Luo et al. [60] proposed the Segment-CNN where the sentence is divided into segments and trained by separate convolutional units. Based on where the entities are located in the sentence, we divide the sentence into segments. Different segments play different roles in determining the relation. We re-implemented the base architecture of the *Seg-CNN* [60] and fine-tuned the parameters based on its performance on the datasets. A sentence is explicitly segmented into five segments:

- preceding - tokenized words before the first concept

- concept 1 - tokenized words in the first concept
- middle - tokenized words between the two concepts
- concept 2 - tokenized words in the second concept
- succeeding - tokenized words after the second concept

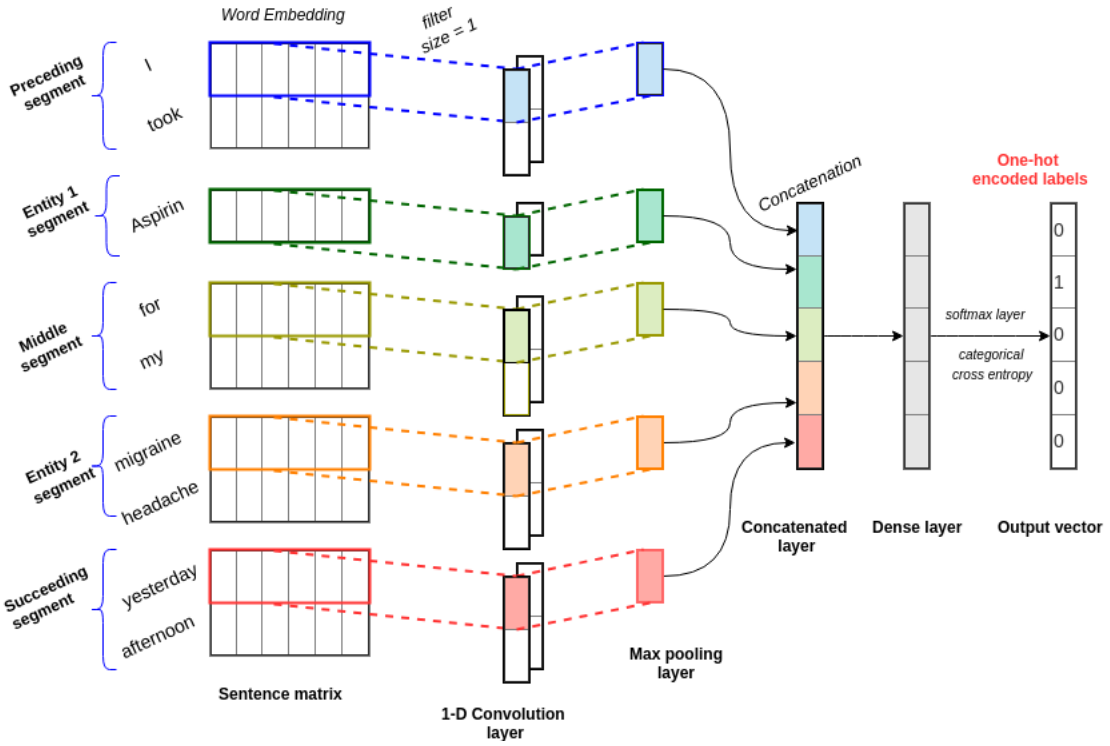

Fig. 14. An illustration of the Segment-CNN architecture when performing multi-class classification.

Fig. 14 shows an abstract view of the construction of the Segment-CNN when performing multi-class classification. If k is the dimension of the word embeddings and N is the number of words in a segment, the segment is represented by a $k * N$ matrix where word embeddings are the columns. We construct separate convolution units for each segment and concatenate before the fixed-length vector is fed to the dense layer that performs the classification. Each convolution unit applies a sliding window that processes the segment and feeds the output to the max-pooling layer to

extract important features independent of their location. The output features of the max-pooling layer of each segment are then flattened and concatenated into a vector before feeding it into the fully connected feed-forward layer. In Segment-CNN, the vector is finally fed into a softmax layer to perform multi-class classification.

5.4 BERT-based Approach

Our BERT-based approach [134] explores using BERT contextualized embeddings into a simple feed-forward neural network. We first extract the sentence containing the relation and pass it through a pre-trained BERT model. The output is then fed into a dropout layer and then into a fully-connected dense layer for classification. As with our CNN-based approaches, we treat the RE as a binary classification task building a separate model for each drug-entity type. Fig. 15 shows the architecture of our BERT-based approach. Our approach contains the following BERT-based language models:

- *BERT [10] (cased and uncased).* The original BERT models are trained on a large corpus of English data: BookCorpus (800M words) and Wikipedia (2,500M words) in a self-supervised manner (without human annotation). BERT-based models are smaller BERT models intended for environments with limited computational resources. BERT-uncased and BERT-cased have 2-heads, 12-layers, 768-hidden units/layer, and a total of 110 M parameters.

- *BioBERT [80].* This model is initialized with the general BERT and further trained over a corpus of biomedical research articles from PubMed[4] abstracts and PubMed Central[5] article full texts.

[4]https://www.ncbi.nlm.nih.gov/pubmed/

[5]https://www.ncbi.nlm.nih.gov/pmc/

- *Clinical BERT [81].* This model is initialized with BioBERT and further fine-tuned over the Medical Information Mart for Intensive Care-III [131] (MIMIC-III) clinical note corpus.

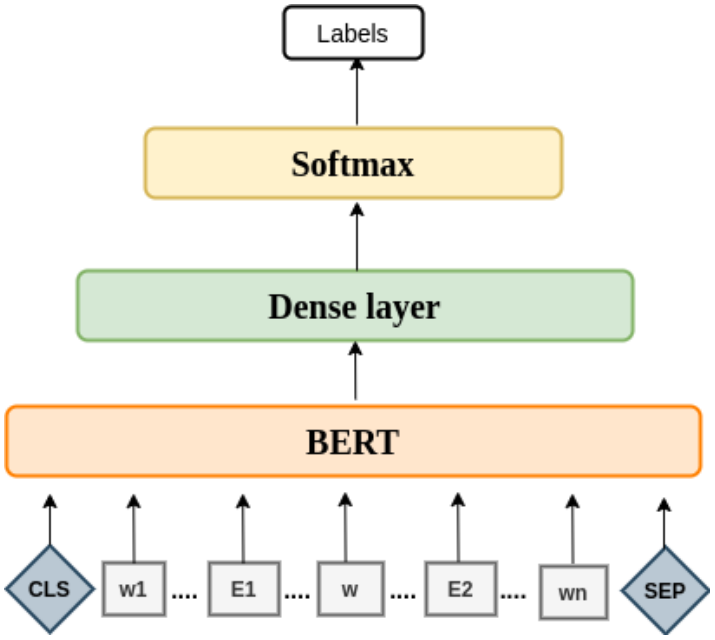

Fig. 15. An illustration of our architecture for the BERT-based approach

5.5 GCN-based Approach

Our GCN-based approach includes two GCN architectures. We explore how to effectively capture the global dependencies between terms within a corpus and how to combine the capability of BERT with a GCN and benefit from the combination. We propose two novel approaches to extract relations between chemical entities: GCN-Vanilla and GCN-BERT.

5.5.1 GCN-Vanilla Approach

In this approach, we first build one single graph with word and sentence nodes over the entire corpus. The number of nodes (V) in the graph equals the number of sentences and the number of unique words in the corpus.

Second, we measure the weight of the edge between two word nodes (word-word nodes) using PMI. The occurrence of two words together can be just by chance or because there is an above-chance frequency of occurrence of two words in that particular order. For example, the term *disturbed sleep* has different independent meanings, but together, they express a precise, unique concept. PMI is a measure that quantifies the likelihood of the co-occurrence of two words. Equation 5.1 shows how PMI is computed between two word nodes. If x and y are independent, their joint probability equals the product of their marginal probabilities that result in a log equal to 0, which means the words occurred by chance. A positive PMI value indicates that the semantic correlation of words in a sentence is high, whereas a negative value indicates no correlation. Only the edges between word pairs with positive PMI values are considered when the graph is generated.

$$PMI(x, y) = \log\left(\frac{P(x, y)}{P(x)P(y)}\right) \tag{5.1}$$

Third, we measure the weight of the edge between the node and sentence (word-sentence nodes) using TF-IDF [17]. With multiple documents inside a corpus, TF-IDF takes into account how frequently tokens appear across multiple documents [18]. Fourth, we utilize pre-trained word embeddings to generate the initial word embeddings for the word nodes. We average the word vectors of the word nodes connected to a sentence node to create an embedding representation for the sentence. Fig. 16 shows the structure of the graph we build for this approach. Nodes that begin with

S are sentence nodes, and the rest are unique word nodes. Black bold edges between sentence nodes and word nodes are sentence-word edges, and the thin black edges between word nodes are word-word edges. Different color sentence nodes depict different classes. We consider this approach as our baseline.

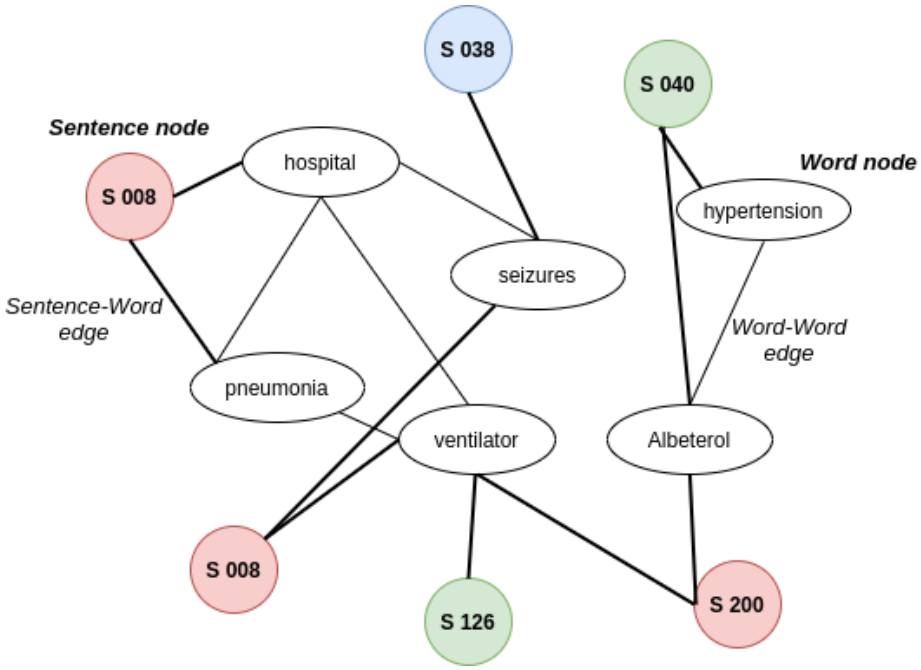

Fig. 16. Structure of the graph for the corpus-level approach.

Fifth, we model the graph with a multi-layer GCN to capture the high-order neighborhoods information. A multi-layer GCN allows message passing between nodes that are not connected directly but a few levels away. A two-layer GCN passes messages from the nodes that are at a maximum of two steps away [14]. Our graph has no direct sentence-sentence nodes, but they are connected through the word nodes; therefore, a two-layer GCN allows information passing from one sentence node to another. Initially, the weight vectors of the nodes are randomly initialized, and then the embeddings are jointly learned for both the words and sentences through the GCN. Fig. 17 demonstrates how the message passing mechanism works on a two-layer GCN

and from which nodes the current node obtains information in the first and second layer of GCN. We consider the sentence node $S1$ as our target node for this example. In the first layer, the sentence node $S1$ gets information from the 1-hop neighbor nodes $W3$ and $W11$, and in the second layer, it gets from the 2-hop neighbor node $S5$. The first segment in the figure shows the sample graph of two sentence nodes and five word nodes with randomly initialized vectors for each node. The middle segment shows sentence node $S1$ and the nodes connected to it get information in the first layer of GCN: $S1$ from $W11$ and $W3$; $W3$ from $S1$ and $W11$; $W11$ from $S1$ and $W3$; $S5$ from $W2$ and $W18$. The last segment shows how the sentence node $S1$ gets the information from the 2-hop neighbor node $S5$ through the 1-hop neighbor node $W3$ in the second layer of GCN.

Finally, the output of the second-layer nodes is fed into a softmax layer for classification. This turns the relation classification problem into a node classification problem. Softmax is calculated as shown in Equation 5.2.

$$Z = \text{softmax}(\tilde{A}\,\text{ReLU}(\tilde{A}\,X\,W_0)W_1), \tag{5.2}$$

where $\tilde{A} = D^{-1/2}AD^{-1/2}$ [14]. The cross entropy error is calculated over all sentences.

5.5.2 GCN-BERT Approach

From our previous works [134] we found our BERT-based approach outperformed other supervised learning approaches. BERT captures the contextual information within a sentence or document locally however it fails to capture the global information. Many real-world data obtained are in the form of graphs and do not have a natural order of nodes. The graph data can be traversed in varying orders to capture the global information. GCN captures the global information between the nodes

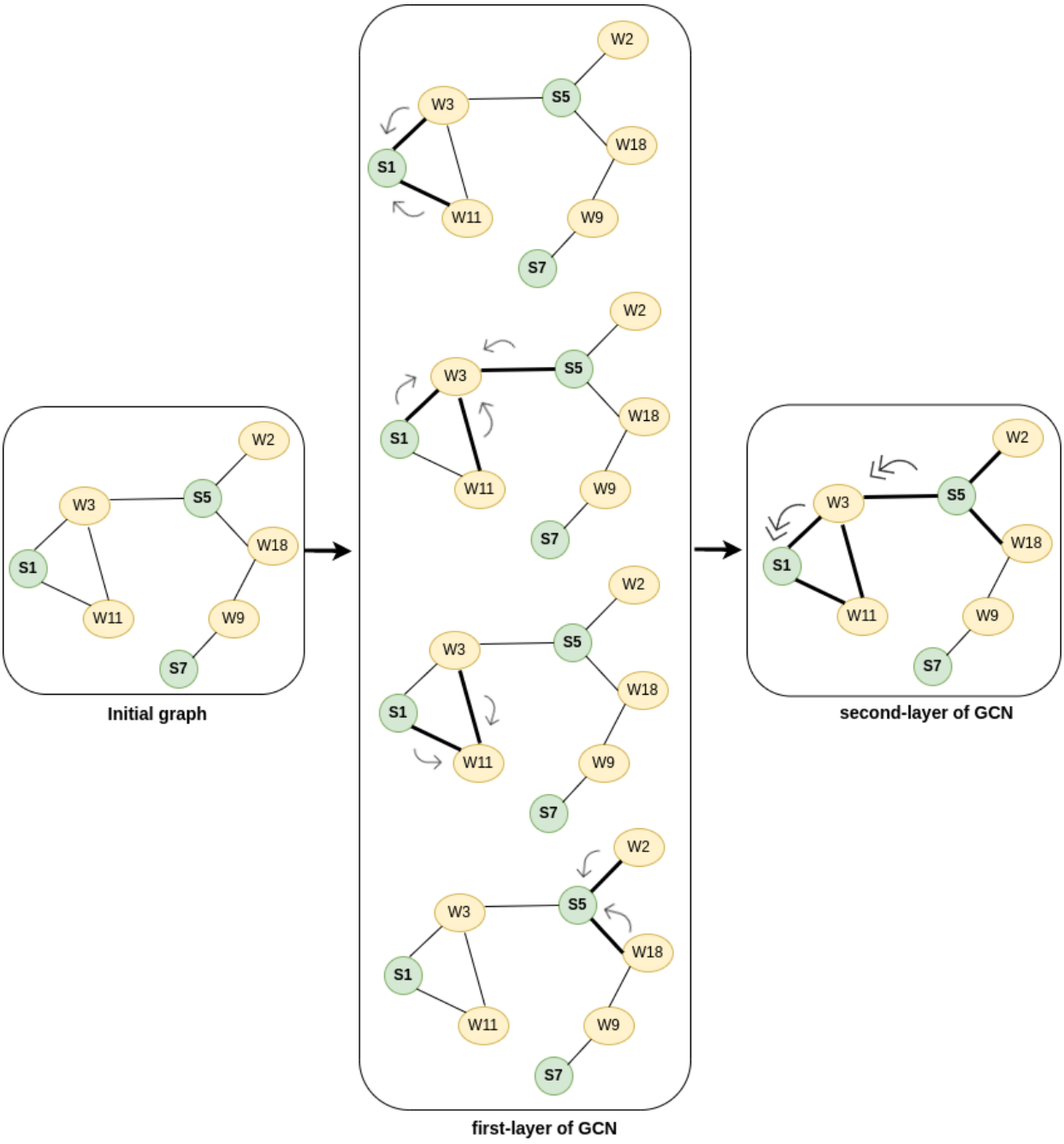

Fig. 17. Illustration of how message passing mechanism works in a two-layer GCN architecture

well but may fail to capture local information. Therefore, we proposed a novel architecture that combines BERT with GCN to benefit from capturing both local and

global information and allowing them to influence mutually and build together a final representation for classification.

First, we extract the sentence where the entity pair is located. We use the BERT tokenizer to tokenize the sentence into words. Since BERT is a pre-trained model, input data needs to be in a specific format, and the BERT tokenizer carries out specific operations to generate the format. First, the words are split into subwords and characters. BERT handles the Out-of-vocabulary (OOV) words by tokenizing them to the character level. They utilize the ## sign to indicate they are part of a larger word, distinguishing a subword token from a word token when generating word embeddings. For example, the word *embeddings* is tokenized with BERT as [*em*, *##bed*, *##ding*, *##s*], here the *##bed* token is differentiated from the token *bed* token. Subword vectors are averaged to generate an approximate vector for the original word. After splitting the sentence into the tokens, we build a vocabulary map mapping the unique tokens to integers.

Second, we generate a vocabulary graph G =(V, E) similar to the graph we built in our GCN-Vanilla approach, but we consider only the word nodes and not the sentence nodes. We denote the word nodes in the graph by the mapped integers, and we measure the weight of the edge between two word nodes (word-word nodes) using PMI, which is calculated as shown in the Equation5.1. The PMI values are normalized (NPMI) between the range of [-1, 1]. A positive NPMI value indicates a high semantic correlation between words, whereas a negative NPMI value indicates little or no semantic correlation. Edges exist between word nodes from the training set when PMI > 0. Next, we pass the graph through a two-layer GCN to generate the graph embeddings. GCN performs multiple levels of convolution to capture the global information between the nodes that are not connected directly. We use ReLU activation function in the GCN [116] described in Equation 5.3.

$$VGCN = ReLU(X_{mv}\tilde{A}_{vv}W_{vh})W_{hc} \tag{5.3}$$

where m is the batch size, v is the vocabulary size, h is the size of the hidden layer, c is the size of the sentence embedding.

Third, we combine the mapped word indices with the generated graph embeddings before passing them into BERT, which helps capture the order of the words in the sentence and the global information captured by the graph. BERT uses a transformer, an attention mechanism that learns contextual relations between words. BERT applies multi-layer multi-head self-attention on the concatenated input. The input text sequence travels through a stack of 12 encoders at each level and a feed-forward neural network and outputs a sentence embedding for classification. Usually, BERT architecture takes a token, segment, and position embeddings of the input text and a [CLS] token. However, here we combine word token embeddings and placeholders for the graph embeddings before passing them into the BERT. When the token embeddings layer converts each word piece token into a vector representation, we combine the graph embeddings vector. Next, BERT applies the bidirectional training, which simultaneously takes the previous and next tokens into account and produces a representation for the input sequence. Finally, the final embedding representation is fed into a fully connected layer for classification.

Fig. 18 shows the overall structure of this combined GCN-BERT approach and illustrates how an input sentence is passed through this architecture. The input sentence $S1$ is tokenized and the word nodes are denoted in blue. In the vocabulary graph, words nodes from the input sentence $S1$ are shown in blue whereas the word nodes from other sentences are shown in yellow. This approach captures and combines the input text's local and global information.

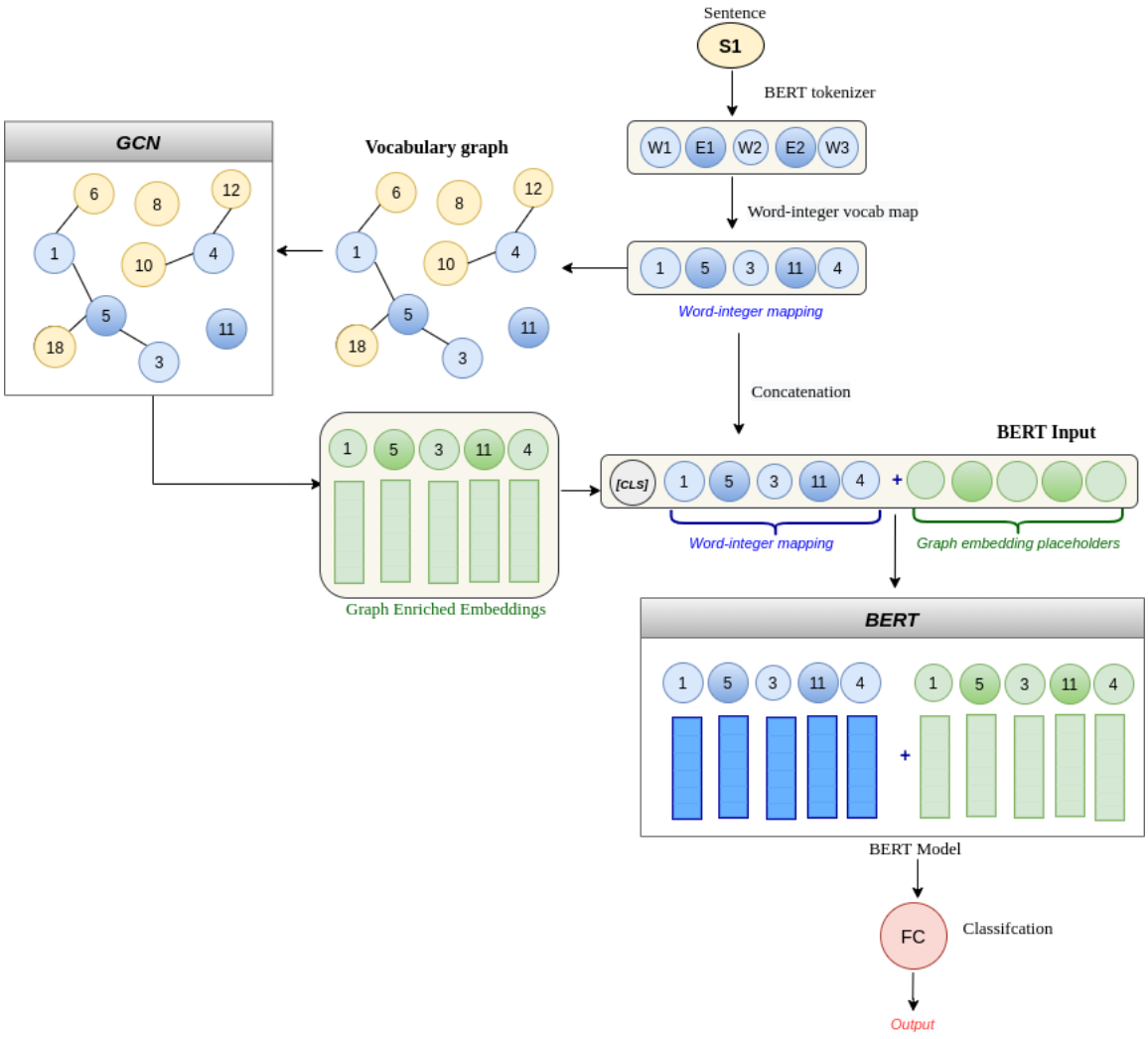

Fig. 18. Structure for the GCN-BERT combined approach.

5.6 Evaluation Criteria

Several evaluation metrics are used to evaluate and compare the performance of RE: Precision (P), Recall (R), and F_1 score (F). Precision calculates how many instances are predicted correctly, and Recall calculates out of all the correct instances that should have been predicted how many instances are correctly predicted. The F_1 score is the harmonic mean of the Precision and Recall, where an F_1 score reaches its

best value at 1 and worst at 0. Accuracy metrics alone can be misleading when there is a great difference in the number of instances in each relation. Therefore calculating a confusion matrix can give a better picture of what the classifier is getting right and what types of errors it is making.

The micro, macro, and weighted averages of the system report the performance of the system. Micro average calculates metrics globally by counting the total true positives, false negatives, and false positives. In contrast, macro average calculates metrics for each label and the unweighted mean as it does not take class imbalance into account. But weighted average calculates metrics for each label, and the average weighted by support (the number of true instances for each label).

Confusion matrices help visualize the classifier's performance over a dataset, compared the predicted and actual instances, and analyze the correctly classified and misclassified instances. It is mainly used for predictive analysis in machine learning. Rows represent the predicted relation instances, whereas the columns represent the instances of the actual relation. Confusion matrices report the number of false positives (FP), false negatives (FN), true positives (TP), and true negatives (TN).

K-fold CV is a statistical method commonly used to evaluate machine learning models, and it is performed on the training dataset. Data is divided into k size partitions; out of k partitions, a single partition is retained as testing data. The remaining k-1 partitions are considered as training data to train the model.

CHAPTER 6

EXPERIMENT: MULTI-LABEL HIERARCHICAL RELATION EXTRACTION

Relation extraction can be challenging as instances may contain multiple medical entities with more than one type of relation in the same sentence. For example, Fig. 5 in Section 4.1 shows a sentence from the i2b2-2010 clinical dataset with three entity types (Treatment, Test, and Problem) and the relations between them. We can see there is more than one relation in the same sentence between the multiple entity pairs: TrCP (Treatment causes problem), TrIP (Treatment improves problem), and TeRP (Test reveals problem). We can see two types of relations (TrCP, TrIP) exist between the same entity types, Problem and Treatment. This is an example of a multi-label classification problem. In this work, we drift from the traditional approaches with feature engineering and utilize deep learning approaches for relation extraction that automatically learn features from sentences [27]. Here, we propose a novel multi-label hierarchical architecture that utilizes CNNs.

A sentence can contain two entities, and there may or may not be a relation between them. When a relation exists, it is considered a positive instance and is further divided into many relation classes that we consider as positive relations. At the same time, when a relation does not exist between the entities it is considered as a negative instance and included in the negative relations. The unbalanced distribution of relation instances between positive and negative relations may decrease the performance of a classifier. Therefore, we propose a hierarchical schema to minimize the impact of the heavy imbalanced classes.

6.1 Methods

In this work, we use the three CNN architectures for clinical RE and propose a hierarchical classification schema that alleviates the difficulty of the class imbalance of negative relations.

6.1.1 Convolutional Neural Network Architectures

Here, we use Single-label Sentence-CNN, Multi-label Sentence-CNN, and Segment-CNN discussed in Section 5.3 of Chapter 5.

Single-label Sentence-CNN. Single-label Sentence-CNN represents each entity pair by a sentence. Each relation consists of two entities for example, in i2b2 the relation Tr-P consists of *Treatment* and *Problem*. In this architecture, the Sentence-CNN learns the relation representation for the entire sentence as a whole. Fig. 12 in Section 5.3 steps through the architecture in detail. However, a sentence can contain more than one distinct mention of relation, and at that point, this architecture cannot differentiate the sentences fed into CNN.

Multi-label Sentence-CNN. Multi-label Sentence-CNN is a modification to single-label Sentence-CNN. This architecture considers all possible labels for a sentence. Fig. 13 in Section 5.3 steps through the architecture in detail.

Segment-CNN. Segment-CNN divides a sentence into five segments and train using separate convolutional units. This architecture is used as our base model to compare the performance of our multi-label Sentence-CNN. Fig. 14 in Section 5.3 steps through the architecture in detail.

6.1.2 Hierarchical Classification

Relations are bounded by sentences in our dataset. The unbalanced distribution of relation instances between positive and negative relations may decrease the performance of a classifier. In hierarchical classification, first, we perform binary classification to classify the positive and negative relations, then we perform multi-class classification on the positive predictions:

1. Re-label the positive and negative instances as 'yes' and 'no' respectively.
2. Train the models on the training set and predict on the test set.
3. Remove all 'no' instances from the test set according to the predicted labels and from the train set according to the original labels.
4. Re-train the model with the original labels and predict on the test set again and evaluate the predictions.

Fig. 19 demonstrates hierarchical classification using an example from i2b2-2010 relation corpus. Tr-P category consists of six relations: one negative and five positive. As described above, first we perform a binary classification to classify the negative instances (NTrP), and then based on the predictions (TrP), we further classify the positive instances into five relations.

6.2 Experimental Design

Word representation. For our CNN-based approaches we use the word2vec, a non-contextualized word embeddings [23].

Text tokenization and vectorization. We use multiple methods to vectorize the text. Text in each segment is converted into vector sequences using the Keras tokenizer by assigning a unique index to each unique word. The Keras tokenizer takes the input data and creates a customized tokenizer that considers only the top,

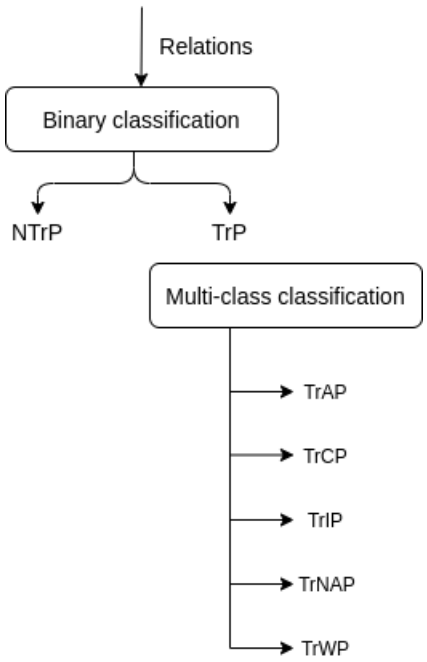

Fig. 19. A diagram depicts how the hierarchical classification for a relation category works on the i2b2-2010 dataset

given the most common words in the input data and creates a word index. Using the tokenizer, data is tokenized, and it returns a vector sequence that is padded according to the maximum length of a sequence. The arguments control the position of the padding. Also, we use one-hot-encoding to convert the text into vectors of 0s and 1s. Given a batch of vector sequences as input, an embedding layer converts the sequence into real-valued embedding vectors. Word embeddings encode similarities between words into close embedding vectors. Initial weights are assigned randomly and adjusted gradually through backpropagation.

Label Binarization. Labels are expressed in terms of a boolean vector that represents each relation. Scikit-learn binarizer converts a label into a boolean vector in a one-vs-all fashion and outputs the label's one-hot encoding. The multi-class label is converted to a boolean vector by assigning a unique value or number to each label in a

categorical feature. For example, if a sentence has three labels (*TrCP, TrIP, TeRP*), the binarized labels for a single-label Sentence-CNN and Segment-CNN would be the following: 1 0 0 0 0 (*TrCP*), 0 1 0 0 0 (*TrIP*) and 0 0 1 0 0 (*TeRP*). If the example is considered for multi-label Sentence-CNN, labels are binarized in the following manner: 1 1 1 0 0

Hyperparameters and Fine-tuning. We experiment with different sliding window sizes, filter sizes, loss functions to fine-tune our approach. Single-label Sentence-CNN and Segment-CNN perform well with small filter sizes, while multi-label Sentence-CNN performs well with large filter sizes. To regularize the model, we apply the dropout technique on the output of the convolution layer. The dropout technique randomly drops a few nodes to prevent co-adaptation of hidden units, and we set this value to 0.5 while training. We use *Adam* and *rmsprop* techniques to optimize our loss function.

6.3 Results and Discussion

In this section, we discuss the results of our three CNN models on the i2b2-2010 *Problem-Treatment-Test* dataset described in Section 4.1 of Chapter 4, followed by comprehensive error analysis.

6.3.1 Results of Individual CNN Architectures

Table 11 shows the Precision, Recall and F_1 score obtained over 5-fold CV on the training set of i2b2-2010 dataset. The number of relations in each relation category is indicated in the left-most column along with the name of each category.

The results show that the Segment-CNN obtains a higher overall F_1 score than both the single and multi-label Sentence-CNN except for the Problem-Problem (PP). To understand these results we conducted an extensive error analysis.

Table 11. Precision (P), Recall (R), and F_1 (F) scores of the individual CNN models over 5-fold CV on the training set of i2b2-2010 dataset.

	Single-label Sentence-CNN			**Multi-label Sentence-CNN**			**Segment-CNN**		
Relation Category	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Tr-P (6)	0.69	0.69	0.69	0.71	0.62	0.66	0.70	0.71	**0.71**
Te-P (3)	0.73	0.73	0.73	0.75	0.7	0.72	0.78	0.78	**0.78**
P-P (2)	0.87	0.87	0.87	0.93	0.89	**0.92**	0.85	0.85	0.85

6.3.1.1 Analysis of the multi-labels

Table 12. Precision (P), Recall (R), and F_1 (F) scores of the multi-label Sentence-CNN model on partitioned i2b2-2010 dataset

	Single labels only				**Multi-labels only**				**All labels**		
Relation	**# instances**	**P**	**R**	**F**	**# instances**	**P**	**R**	**F**	**P**	**R**	**F**
Tr-P	644	0.65	0.64	0.65	240	0.96	0.59	**0.73**	0.71	0.62	0.66
Te-P	738	0.61	0.82	0.70	209	0.88	1.00	**0.93**	0.75	0.7	0.72
P-P	1039	0.93	0.68	0.78	469	1.00	0.78	0.88	0.93	0.89	**0.92**

To analyze the multi-label CNN model, we separated the instances with only one label per instance and multiple labels per instance and re-evaluated the multi-label Sentence-CNN. Table 12 shows the Precision, Recall, and F_1 score when running the model over just the single-label instances, just the multi-label instances, and then overall the instances for comparability. The table also contains the number of instances that contain either single or multiple labels. The results show that the multi-label Sentence-CNN obtains a higher F_1 score when classifying instances with multiple labels than those that only have a single relation. This indicates that the multi-label classification is performing well when there are multiple labels per sentence.

6.3.1.2 Analysis over each category of the i2b2-2010 training dataset

Here, we analyze the individual categories of each of the relations. Tables 15, 13, and 14 show a class-wise performance comparison with all three CNN models for each

relation category of the i2b2-2010 relation corpus. The tables contain the number of instances of each relation type indicated in the left-most column along with the name of each relation of the category.

Table 13. Precision (P), Recall (R), and F_1 (F) scores of the CNN models for the Treatment-Problem (Tr-P) category of the i2b2-2010 dataset training set

	Single-label Sentence-CNN			**Multi-label Sentence-CNN**			**Segment-CNN**		
Relation	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
NTrP (1702)	0.76	0.82	0.79	0.64	0.57	0.6	0.77	0.87	**0.82**
TrAP (885)	0.56	0.65	0.60	0.76	0.82	0.79	0.62	0.66	**0.64**
TrCP (184)	0.51	0.11	0.18	0.73	0.21	**0.33**	0.76	0.21	**0.33**
TrNAP (62)	0.86	0.1	0.17	1.00	0.09	0.17	0.93	0.21	**0.34**
TrIP (51)	1.00	0.04	**0.08**	0.5	0.03	0.05	0.00	0.00	0.00
TrWP (24)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Micro avg	0.69	0.69	0.69	0.71	0.62	0.66	0.73	0.73	**0.73**

Table 14. Precision (P), Recall (R), and F_1 (F) scores of the CNN models for Test-Problem (Te-P) category of the i2b2-2010 dataset training set

	Sentence-CNN (single-label)			**Multi-label Sentence-CNN**			**Segment-CNN**		
Relation	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
NTeP (993)	0.73	0.83	0.78	0.68	0.62	0.65	0.78	0.86	**0.82**
TeCP (166)	0.60	0.34	0.43	0.81	0.32	0.46	0.68	0.39	**0.50**
TeRP (993)	0.75	0.66	0.70	0.78	0.84	**0.81**	0.79	0.75	0.77
Micro avg	0.73	0.73	0.73	0.75	0.7	0.72	0.78	0.78	**0.78**

Table 14 shows the performance of the models on the relations of the category Problem-Test (Te-P). Again, we can see an imbalance between the relation TeCP and other relations. Since this relation does not have enough instances to distinguish from other relations we can see the performance is comparatively lower.

Table 15 shows the performance of the models on the relations of the category Problem-Problem (P-P). This category has two relations only, and hence it is a binary classification. Here, the multi-label Sentence-CNN model obtains a higher F_1 score

Table 15. Precision (P), Recall (R), and F_1 (F) scores of the CNN models for Problem-Problem (P-P) category of the i2b2-2010 dataset training set

	Single-label Sentence-CNN			Multi-label Sentence-CNN			Segment-CNN		
Relation	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
NPP (4418)	0.88	0.98	0.93	1.00	1.00	**1.00**	0.88	0.96	0.92
PIP (755)	0.17	0.03	0.05	0.65	0.56	**0.6**	0.12	0.04	0.05
Micro avg	0.87	0.87	0.87	0.93	0.89	**0.91**	0.85	0.85	0.85

than the other two models. We believe that this is due to the possibility of multiple *Problems* described in an instance where not all of the *Problems* are related, which won't be able to be captured by the single-label Sentence-CNN.

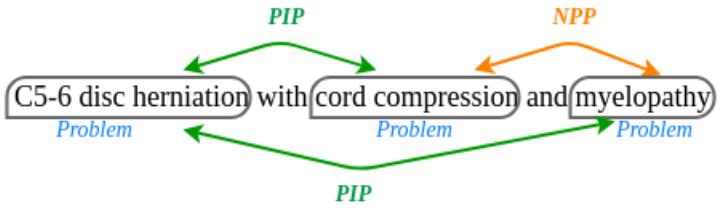

Fig. 20. A sentence from i2b2-2010 relation corpus depicting multiple pairs of medical entities in the same sentence.

For example, Fig. 20 shows a sentence from the P-P relation category of i2b2-2010 where multiple *Problems* are located in the same sentence, but not all the *Problems* are related to each other (NPP). Table 13 shows the performance of the models on the relations of the category Problem-Treatment (Tr-P). From the results, we can see a decrease in performance when the number of instances of the relation decreases. The last two relations (TrIP, TrWP) do not have enough instances to differentiate themselves from other instances, which results in poor performance compared to other relations.

The confusion matrices for the single-label Sentence-CNN and Segment-CNN are shown in Tables 16 and 17 respectively. The rows indicate the gold standard, and the

Table 16. Confusion matrix of the system outputs of the single-label Sentence-CNN model on i2b2-2010 training set

	Treatment-Problem (Tr-P)						**Test-Problem (Te-P)**			**Problem-Problem (P-P)**	
	NTrP	TrAP	TrCP	TrIP	TrNAP	TrWP	NTeP	TeCP	TeRP	NPP	PP
NTrP	***1405***	297	12								
TrAP	299	***565***	2								
TrCP	81	76	***19***		1						
TrIP	24	23		***2***							
TrNAP	20	33	3		***6***						
TrWP	13	10	1								
NTeP							***1027***	22	181		
TeCP							69	***55***	40		
TeRP							314	14	***647***		
NPP										***5085***	105
PP										718	***22***

Table 17. Confusion matrix of the system output of the Segment-CNN model on i2b2-2010 training set

	Treatment-Problem (Tr-P)						**Test-Problem (Te-P)**			**Problem-Problem (P-P)**	
	NTrP	TrAP	TrCP	TrIP	TrNAP	TrWP	NTeP	TeCP	TeRP	NPP	PP
NTrP	***1485***	225	4								
TrAP	295	***568***	2		1						
TrCP	82	57	***38***								
TrIP	22	24	3								
TrNAP	18	28	3		***13***						
TrWP	17	7									
NTeP							***1052***	17	161		
TeCP							68	***64***	32		
TeRP							230	13	***732***		
NPP										***5004***	186
PP										714	***26***

columns indicate the predicted labels. The bold terms indicate the correctly classified instances of each relation. Zero instances are removed for better visualization. The results are consistent with our previous observation. For all cases, the majority label is where most of the errors are occurring, indicating that the class imbalance plays a significant role in the miss-annotation.

6.3.1.3 Performance analysis removing the negative instances

Since the number of instances of no-relation pair relations in this dataset is higher than the number of instances of positive relations, they impact the performance of each relation category. Therefore, we investigated removing the instances of negative relations and re-evaluate the models.

Tables 18 and 19 show the results for the Problem-Treatment (Tr-P), and Problem-Test (Te-P) relations. The number of instances of each relation type is indicated in the left-most column along with the name of each relation of the category. Bold terms indicate the best performing method for a dataset. The results show an improvement in the number of instances per relation and the overall F_1 score for both Problem-Treatment (Tr-P) and Problem-Test (Te-P) relations. The multi-label Sentence-CNN results show the noticable increase in results obtaining a higher F_1 score than the other models.

Table 18. Precision (P), Recall (R), and F_1 (F) scores of the CNN models for Treatment-Problem (Tr-P) in the i2b2-2010 dataset after removal of the negative relations

	Single-label Sentence-CNN			**Multi-label Sentence-CNN**			**Segment-CNN**		
Relation	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
TrAP (885)	0.82	0.98	0.89	0.92	0.88	**0.90**	0.77	0.99	0.87
TrCP (184)	0.74	0.46	0.57	0.70	0.58	**0.64**	0.72	0.20	0.32
TrNAP (62)	0.80	0.26	0.39	0.67	0.40	**0.50**	0.80	0.13	0.22
TrIP (51)	0.92	0.22	**0.36**	0.00	0.00	0.00	1.00	0.14	0.25
TrWP (24)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Micro avg	0.81	0.81	0.81	0.88	0.76	**0.82**	0.77	0.77	0.77

We can see several important observations from the above results.: First, the overall performance of each relation category has increased for all three CNN models. Second, multi-label Sentence-CNN outperforms Segment-CNN, and there is an in-

Table 19. Precision (P), Recall (R), and F_1 (F) scores of the CNN models for Test-Problem (P-Te) in the i2b2-2010 dataset after removal of the negative relations

	Single-label Sentence-CNN			**Multi-label Sentence-CNN**			**Segment-CNN**		
Relation	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
TeCP (166)	0.72	0.35	0.47	0.79	0.48	**0.59**	0.83	0.41	0.55
TeRP (993)	0.90	0.98	0.94	0.90	0.98	0.94	0.91	0.99	**0.95**
Micro avg	0.89	0.89	0.89	0.89	0.90	**0.90**	0.90	0.90	**0.90**

crease in Sentence-CNN results than before. Third, the performance of each relation increased than before when running with a negative relation. The classification accuracy of each relation improved when the negative relation is removed as the relation distribution is more balanced. Since the Problem-Problem (P-P) category had two relations, including the negative relation it is not considered for evaluation.

6.3.1.4 Performance Analysis for binary classification

Here, we evaluate the binary classification of each of the entity classes. We separated the negative and positive instances and relabeled them as *yes* and *no* and ran the models on the i2b2-2010 training set (5-fold CV), considering the classification as a binary classification. Table 20 shows the results for each of the Problem-Treatment (Tr-P), Problem-Test (Te-P), and Problem-Problem (P-P) instances. The results are interesting with the multi-label Sentence-CNN due to the binary classification task. The model could label an instance as having a positive relation and having a negative relation simultaneously. There could be multiple entity pairs where there is a relationship between some but not others.

Table 20. Precision (P), Recall (R), and F_1 (F) scores of the binary classification on the i2b2-2010 dataset

		Single-label Sentence-CNN			Multi-label Sentence-CNN			Segment-CNN		
	Relation	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Tr-P	Neg (1714)	0.81	0.82	0.81	0.58	0.56	0.57	0.83	0.85	**0.84**
	Pos (1178)	0.73	0.71	0.72	0.76	0.96	**0.85**	0.78	0.74	0.76
	Micro avg	0.77	0.77	0.77	0.71	0.81	0.76	0.81	0.81	**0.81**
Te-P	Neg (1230)	0.81	0.80	**0.81**	0.69	0.66	0.67	0.81	0.80	0.80
	Pos (1139)	0.79	0.80	0.80	0.85	0.94	**0.89**	0.79	0.80	0.79
	Micro avg	0.80	0.80	**0.80**	0.77	0.81	0.79	0.80	0.80	**0.80**
P-P	Neg (1714)	0.88	1.00	0.93	1.00	1.00	**1.00**	0.88	0.99	0.93
	Pos (1178)	0.18	0.00	0.01	0.69	0.49	**0.57**	0.24	0.01	0.03
	Micro avg	0.87	0.87	0.87	0.94	0.87	**0.90**	0.87	0.87	0.87

6.3.1.5 Analysis of each category of i2b2-2010 test dataset

To see if the results were not just a fluke in the training data, we experimented with the performance of our approach on the i2b2-2010 test dataset. Tables 21, 22 and 23 show the Precision, Recall and F_1 score when the CNN models are trained on the i2b2-2010 training set and tested on i2b2-2010 test dataset. The number of instances of each relation type is indicated in the left-most column along with the name of each relation of the category. The difference between the test and the training is the test set contains roughly twice as many instances as the train set, and overall these results are slightly lower than the 5-fold CV results.

Table 23 shows the results when the models are run on the test set of the P-P category. There is a heavy imbalance between the relations, and its negative affect the performance. From the results, we can see the instances of the minority class are mostly predicted as the majority class as the models are trained more on instances from the majority class. Multi-label Sentence-CNN obtained a higher Recall than the other two models but slightly lower Precision than the Segment-CNN.

Table 21. Precision (P), Recall (R), and F_1 (F) scores of the CNN models for the Treatment-Problem (Tr-P) category of the i2b2-2010 test dataset

	Single-label Sentence-CNN			**Multi-label Sentence-CNN**			**Segment-CNN**		
Relation	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
NTrP (2376)	0.67	0.67	0.67	0.55	0.56	0.55	0.60	0.79	**0.68**
TrAP (1726)	0.50	0.65	0.57	0.74	0.79	**0.76**	0.47	0.45	0.46
TrCP (342)	0.46	0.27	**0.34**	0.85	0.12	0.22	0.06	0.00	0.01
TrNAP (112)	1.00	0.03	**0.05**	0.50	0.01	0.02	0.00	0.00	0.00
TrIP (152)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TrWP (108)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Micro avg	0.58	0.58	0.58	0.67	0.56	**0.61**	0.55	0.55	0.55

Table 22. Precision (P), Recall (R), and F_1 (F) scores of the CNN models for the Test-Problem (Te-P) category of the i2b2-2010 test dataset.

	Single-label Sentence-CNN			**Multi-label Sentence-CNN**			**Segment-CNN**		
Relation	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
NTeP (1881)	0.62	0.66	0.64	0.61	0.48	0.54	0.50	0.92	**0.65**
TeCP (355)	0.00	0.00	0.00	0.74	0.11	**0.19**	0.18	0.01	0.01
TeRP (2054)	0.65	0.72	0.68	0.79	0.90	**0.84**	0.77	0.29	0.42
Micro avg	0.64	0.64	0.64	0.74	0.68	**0.71**	0.55	0.55	0.55

Table 23. Precision (P), Recall (R), and F_1 (F) scores of the CNN models for the Problem-Problem (P-P) category of the i2b2-2010 test dataset.

	Single-label Sentence-CNN			**Multi-label Sentence-CNN**			**Segment-CNN**		
Relation	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
NPP (22087)	0.94	1.00	0.97	1.00	1.00	**1.00**	0.94	1.00	0.97
PIP (1447)	0.00	0.00	0.00	0.34	0.42	**0.37**	0.39	0.01	0.02
Micro avg	0.94	0.94	**0.94**	0.90	0.93	0.92	0.94	0.94	**0.94**

Table 21 shows the Problem-Treatment (Tr-P) results. The results show that no one model is performing better than the other across the categories. However, for the Problem-Test (Te-P) results shown in Table 22 the multi-label Sentence-CNN obtains a higher F_1 score overall.

6.3.2 Results of Hierarchical Classification

Here, we evaluate our hierarchical classification methodology by building the models on the i2b2-2010 training set and evaluating on the test set. First, we re-labeled the instances as *yes* and *no* for positive and negative instances. Then we trained the models on the training set and predicted on the test set. Next, we removed instances with *no* labels from the test set according to the predicted relations and from the train set according to the original relations. Then we re-trained the model with the original relations and predicted on the test set again. Finally, we evaluated the prediction with the given original relations.

Table 24. Precision (P), Recall (R), and F_1 (F) scores for the hierarchical classification of Treatment-Problem (Tr-P) category on i2b2-2010 data set

		Single-label Sentence-CNN			**Multi-label Sentence-CNN**			**Segment-CNN**		
		P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Binary	NTrP	0.69	0.58	0.63	0.71	0.44	0.54	0.55	0.92	**0.69**
	Tr-P	0.65	0.74	0.69	0.60	0.82	**0.70**	0.78	0.27	0.40
	Micro avg	0.58	0.58	0.58	0.67	0.56	**0.61**	0.55	0.55	0.55
Tr-P	TrAP	0.68	1.00	0.81	0.77	0.96	**0.85**	0.69	1.00	0.82
	TrCP	0.74	0.06	0.11	0.95	0.11	**0.20**	0.00	0.00	0.00
	TrNAP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	TrIP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	TrWP	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Micro avg	0.69	0.69	0.69	0.77	0.68	**0.72**	0.69	0.69	0.69
Previous	***Micro avg***	0.58	0.58	0.58	0.67	0.56	**0.61**	0.55	0.55	0.55

Tables 24 and 25 show the results of hierarchical classification for categories Problem-Treatment (Tr-P) and Problem-Test (Te-P) respectively. First, we report the performance of binary classification; second, we report the performance of the multi-class classification of the positive relations for each relation individually; and third, we report the previous results from Tables 14 and 13 (Previous) for ease of comparison. The results show an overall a improvement in the performance across

Table 25. Precision (P), Recall (R), and F_1 (F) scores for the hierarchical classification of Test-Problem (Te-P) category on i2b2-2010 data set

		Single-label Sentence-CNN			**Multi-label Sentence-CNN**			**Segment-CNN**		
		P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Binary	NTeP	0.67	0.59	0.63	0.66	0.46	0.54	0.54	0.85	**0.67**
	Te-P	0.70	0.77	**0.73**	0.66	0.82	**0.73**	0.79	0.44	0.56
Te-P	TeCP	0.66	0.18	0.28	0.45	0.47	**0.46**	0.00	0.00	0.00
	TeRP	0.88	0.99	0.93	0.90	0.98	0.94	0.92	1.00	**0.96**
	Micro avg	0.88	0.88	0.88	0.84	0.91	0.88	0.92	0.92	**0.92**
Previous	***Micro avg***	0.64	0.64	0.64	0.74	0.68	**0.71**	0.55	0.55	0.55

Table 26. Indirect comparison with current related works

	Tr-P			**Te-P**			**P-P**			**System**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Multi-label hierarchical CNN	***0.77***	0.68	***0.72***	***0.84***	***0.91***	***0.88***	***0.9***	***0.93***	***0.92***	***0.84***	***0.84***	***0.84***
Luo, et al. [60]	0.69	**0.69**	0.69	**0.84**	0.80	0.82	0.70	0.70	0.70	0.74	0.73	0.74
Sahu, et al. [107]	-	-	-	-	-	-	-	-	-	0.76	0.67	0.71
Li, et al. [121]	-	-	-	-	-	-	-	-	-	0.76	0.73	0.74
Tang, et al. [69]	-	-	-	-	-	-	-	-	-	0.72	0.70	0.71
Chika, et al. [72]	0.50	0.52	0.51	-	-	-	-	-	-	-	-	-

all three models indicates the effectiveness of hierarchical classification.

6.3.3 Comparison with Previous Work

Table 26 shows an indirect comparison with five state-of-the-art deep learning-based RE approaches evaluated over the i2b2-2010 dataset. Li, et al. [121], Chika, et al. [72] and Tang, et al. [69] report overall system results over the test data; while Sahu, et al. [107] report the 5-fold CV results over the training data. Chika, et al. [72] only evaluate over the Problem-Treatment (Tr-P) category. The results show that our multi-label Sentence-CNN obtained on par or higher results.

6.4 Conclusions and Contributions

Here, we evaluate a novel multi-label hierarchical architecture that utilizes CNNs. We compare our model with two additional CNN approaches based on Luo. et, al [60]. Our results show that multi-label Sentence-CNN obtains a higher F_1 score overall and outperforms the other approaches. We perform hierarchical classification to remove the influence of the negative instances during the multi-class classification. The results show an overall significant improvement in the performance across all three approaches, which indicates the effectiveness of hierarchical classification. The contributions of this work are:

1. A sentence often has more than one distinguish entities, resulting in multiple possible relations within the sentence. Here, we proposed a multi-label Sentence-CNN to predict multiple relations labels for a single sentence.
2. The number of relations for an entity pair is often highly imbalanced since most of the possible entity pairs do not result in a relation (Negative relation). The unbalanced distribution of relation instances between positive and negative relations may decrease the performance of a classifier. Therefore, we proposed a hierarchical classification approach that utilizes CNNs for datasets that include no-relation pairs to alleviate the class imbalance.

CHAPTER 7

EXPERIMENT: EVALUATION OF RELATION EXTRACTION APPROACHES FOR SINGLE LABEL RELATIONS

Clinical datasets which explore the relationship between a drug entity and its associated attribute entities can have a single relation per entity pair. One such dataset is the n2c2-2018 dataset. For example, Fig. 21 shows a sentence containing the drug *Naoloxone* in which the patient was given *0.4 mg* causing the ADE *quite agitated*. These relations differ from the previous relations 4.1: 1) as there is only a single relation type between an entity pair; 2) all entities relate to a single anchor entity (in this case, the medication), and 3) there may be more than one anchor entity in a sentence. In this work, we evaluate the following three state-of-the-art RE approaches: a rule-based approach utilizing co-location information, a deep learning-based approach utilizing CNNs, and a BERT-based approach.

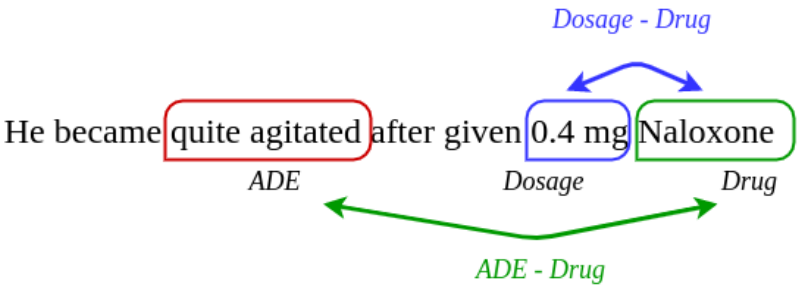

Fig. 21. Sentence depicting a drug and its associated attributes.

7.1 Methods

In this work, we use our rule-based approaches discussed in Section 5.1 in Chapter 5, two CNN-based approaches discussed in Section 5.3 in Chapter 5, and a BERT-based approach discussed in Section 5.4 in Chapter 5.

7.1.1 Rule-based Approach

Here, we use three traversal mechanisms: *left-only traversal*, *first-left-then-right traversal*, and *sentence-bound traversal*. In *left-only traversal*, we traverse to the left side of the entity finding the closest occurrence of the drug. In *first-left-then-right traversal*, we traverse to the left side of the entity first to find the closest occurrence of the drug, if not we traverse the right side. In *sentence-bound traversal* where we find all the closest occurrences of drugs for the entity within the sentence. All traversal techniques are within the sentence boundary.

7.1.2 CNN-based Approach

Here, we use two CNN architectures. We treat the RE task as a binary classification task, building a separate model for each drug-entity type to determine whether a relation exists between two entities. Also, we explore two non-contextualized word embeddings types, word2vec [23] and GloVe [24], described in Section 2.1.2.

Sentence-CNN. For each drug-entity pair, we extract the sentence containing the relation and feed it into a CNN where each word in the sentence is represented as a vector embedding.

Segment-CNN. For each drug-entity pair, we extract the sentence containing the relation, divide it into five segments, and train each segment using separate convolutional units.

7.1.3 BERT-based Approach

Here, we explore the following BERT-based language models: BERT-cased [10], BERT-uncased [10], BioBERT [80], and Clinical BERT [81].

7.2 Experimental Design

Word representation. We use word2vec and GloVe representations for our CNN-based approaches and BERT word embeddings for our BERT-based approaches.

Text tokenization and vectorization. For the rule-based and the CNN-based approaches, we use SpaCy tokenizer [1] and Keras tokenizer [137]. For the BERT-based approaches, we use BertTokenizer and AutoTokenizer [81].

Hyperparameters and Fine-tuning. We define our model training hyper-parameters by adjusting the batch size, learning rate, and the number of epochs. We use the batch size of 512, rmsprop optimizer with the learning rate of 0.001, and train for 10-20 epochs for our CNN-based approach. For our BERT-based approach, we used the HuggingFaceTransformers [2] to build the BERT model on RE task with Tensorflow 2.0. We use TFRecord to read data into a Dataset object efficiently. We use SparseCategoricalCrossentropy as the loss function and Adam as the optimizer to minimize the loss function.

7.3 Results and Discussion

In this section, we discuss the results of our three approaches on the n2c2-2018 dataset described in Section 4.2 of Chapter 4 and compare our results across with previous works.

---

[1] https://spacy.io/api/tokenizer

[2] https://huggingface.co/transformers/

7.3.1 Results of Individual Approaches

7.3.1.1 Rule-based Results

Table 27. Precision (P), Recall (R), and F_1 (F) scores on the test set of n2c2-2018 dataset for our rule-based approaches

	Left-only			**Left-Right (unbounded)**			**Left-Right (bounded)**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Strength-Drug	0.96	0.95	**0.95**	0.46	0.90	0.61	0.94	0.94	0.94
Duration-Drug	0.78	0.69	**0.73**	0.58	0.74	0.65	0.46	0.41	0.43
Route-Drug	0.90	0.89	**0.89**	0.45	0.64	0.53	0.37	0.36	0.37
Form-Drug	0.98	0.98	**0.98**	0.62	0.63	0.63	0.67	0.66	0.67
ADE-Drug	0.46	0.39	0.43	0.55	0.75	**0.64**	0.60	0.51	0.55
Dosage-Drug	0.89	0.89	**0.89**	0.61	0.57	0.59	0.89	0.88	0.89
Reason-Drug	0.48	0.35	0.41	0.61	0.57	**0.59**	0.39	0.28	0.33
Frequency-Drug	0.98	0.98	**0.98**	0.39	0.62	0.48	0.10	0.10	0.10
System (Micro)	0.88	0.83	**0.86**	0.50	0.67	0.57	0.56	0.53	0.55
System (Macro)	0.85	0.80	**0.83**	0.61	0.70	0.63	0.58	0.53	0.55

Table 27 shows the Precision, Recall, and F_1 scores for our rule-based approach on the test set of the n2c2-2018 dataset for the top three traversal mechanisms. Analysis of the various traversal mechanisms over all non-drug entities showed that the *Left-only* traversal mechanism obtained the best results except for the entity-drug pair Duration-Drug, ADE-Drug, and Reason-Drug. For these three entities, using the *Left-Right (unbounded)* traversal mechanism obtained the highest F_1 score. This is mainly because all other drug attributes are usually mentioned before the mention of the drug entity. However, the Duration, Reason, and ADE usually are mentioned after the mention of the drug. Overall, This approach achieved an overall Precision of 0.88, Recall of 0.83, and F_1 score of 0.86.

The results indicate that for most entity-drug pairs, co-location information is sufficient to identify most relations. However, the performance of the entity-drug

pair ADE-Drug and Reason-Drug are lower compared to the other relations. Our supposition for this is that the co-location information was insufficient to identify the correct ADE or Reason when multiple drugs were in the same sentence. For example, in the sentence *Since no new infection was found this was presumed steroids and the leukocytosis improved with prednisone taper.* the non-drug entity *leukocytosis* (ADE) is associated with both *steroids* (Drug) and *prednisone* (Drug).

7.3.1.2 CNN-based Results

Table 28. Precision (P), Recall (R), and F_1 (F) scores on the test set of n2c2-2018 dataset for our CNN-based architectures

	Segment-CNN			**Sentence-CNN**		
	P	**R**	**F**	**P**	**R**	**F**
Strength-Drug	0.91	0.88	**0.90**	0.90	0.91	**0.90**
Duration-Drug	0.39	0.90	0.55	0.41	0.90	**0.57**
Route-Drug	0.77	0.89	**0.83**	0.76	0.91	**0.83**
Form-Drug	0.85	0.95	**0.90**	0.85	0.96	**0.90**
ADE-Drug	0.32	0.85	**0.46**	0.32	0.85	**0.46**
Dosage-Drug	0.83	0.92	**0.87**	0.82	0.93	**0.87**
Reason-Drug	0.27	0.88	**0.42**	0.27	0.88	0.41
Frequency-Drug	0.56	0.88	**0.69**	0.56	0.88	**0.69**
System (Micro)	0.69	0.90	**0.78**	0.68	0.92	**0.78**
System (Macro)	0.68	0.90	**0.77**	0.67	0.91	**0.77**

Table 28 shows the Precision (P), Recall (R) and F_1 scores for our Segment-CNN and Sentence-CNN models over the n2c2-2018 test set. The results show that both models performed comparatively similar. In theory, we believed that Segment-CNN should have performed better because the Sentence-CNN cannot differentiate the inputs when multiple drug-entity pairs are located in a sentence, but the results

contradict the assumption. We believe this is because we treat this as a binary classification problem and build a separate model for each relation type.

7.3.1.3 BERT-based results

Table 29 shows the Precision (P), Recall (R) and F_1 scores of the four fine-tuned BERT models over the n2c2-2018 test dataset. Results show that the models obtain a similar performance overall and for each entity-drug pairs. Comparatively, BERT-cased model performs better in some categories than the other models.

Table 29. Precision (P), Recall (R), and F_1 (F) scores on the test set of n2c2-2018 dataset for our BERT-based approaches

	BERT (uncased)			**BERT (cased)**			**BioBERT**			**Clinical BERT**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Strength-Drug	0.86	0.88	0.87	0.86	0.99	**0.92**	0.86	0.90	0.88	0.87	0.82	0.84
Duration-Drug	0.95	0.93	0.94	0.96	0.93	0.94	0.96	0.93	**0.95**	0.96	0.92	0.94
Route-Drug	0.92	0.99	0.95	0.92	0.97	**0.97**	0.92	0.97	0.94	0.92	0.95	0.93
Form-Drug	0.96	0.97	**0.97**	0.96	0.95	0.96	0.96	0.97	0.96	0.96	0.97	**0.97**
ADE-Drug	0.95	0.99	**0.97**	0.95	0.99	**0.97**	0.95	0.99	**0.97**	0.95	0.99	**0.97**
Dosage-Drug	0.93	0.96	0.94	0.93	0.96	**0.95**	0.93	0.96	0.94	0.93	0.89	0.91
Reason-Drug	0.96	0.98	**0.97**	0.96	0.98	**0.97**	0.96	0.99	**0.97**	0.96	0.99	**0.97**
Frequency-Drug	0.93	0.96	**0.94**	0.93	0.92	0.93	0.93	0.95	**0.94**	0.93	0.95	**0.94**
System (Micro)	0.93	0.96	**0.94**	0.93	0.96	**0.94**	0.93	0.95	**0.94**	0.93	0.96	**0.94**
System (Macro)	0.92	0.95	**0.93**	0.92	0.96	**0.93**	0.92	0.95	**0.93**	0.92	0.95	**0.93**

Fig. 22 shows the breakdown of the performance of each relation when the binary classification is performed using the BERT-uncased model. We report Support, Precision, Recall, and F_1 score and *Blue* and *Brown* bars represent the positive and no-relation (negative) relations respectively. The support shows the number of actual occurrences of the relations. When there is no relation between the drug and the entity, we labeled them as no-relation. The Precision, Recall, and F_1 score of the positive relations are higher except Strength-Drug than the negative relations. We believe this explains the higher performance of the BERT-uncased model compared

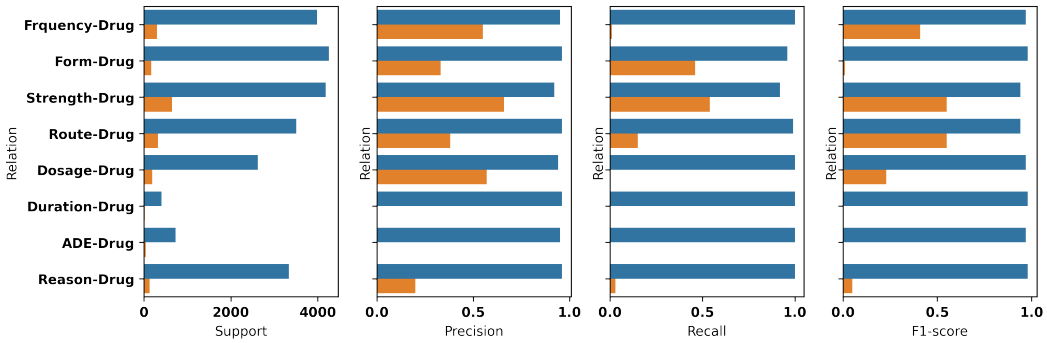

Fig. 22. Error analysis of each relation type during the binary classification using BERT (uncased) model. The *Blue* and *Brown* bars represent the positive and negative relations respectively.

to the other three models. The performance of the no-relation classes is poor, and we believe this is due to the data imbalance of the relations, as shown in the support. Positive relations are significantly larger than the negative relations, and due to this, the poor performance of the negative relations did not affect the performance of the positive relation.

7.3.2 Comparison across Our Approaches

Table 30 shows the Precision (P), Recall (R), and F_1 score for the best results of each of our three approaches: 1) rule-based approach using left-only traversal mechanism; 2) deep learning approach using Segment-CNN, and 3) BERT-based approach using BioBERT. Comparing the rule-based approach with our CNN-based approach shows that the rule-based approach obtained an overall higher Precision, Recall, and F_1 score except for the relations ADE-Drug and Reason-Drug. BERT-based models outperform the other two approaches except for the Strength-Drug, Frequency-Drug, and Form-Drug pairs. The overall Precision and Recall are higher, especially for the entity-drug pairs that performed poorly with the other approaches (ADE-Drug, Reason-Drug, and Duration-Drug). Using pre-trained language representations to

fine-tune models is advantageous as they use minimal task-specific parameters and are trained on the downstream tasks by simply fine-tuning all the pre-trained parameters.

Table 30. Comparison across our approaches over the n2c2-2018 test set

	Train	**Test**	**Rule-based**			**Segment-CNN**			**BioBERT**		
	#	#	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Strength-Drug	6702	4244	0.96	0.95	**0.95**	0.91	0.88	0.90	0.86	0.90	0.88
Duration-Drug	643	426	0.78	0.69	0.73	0.39	0.90	0.55	0.96	0.93	**0.95**
Route-Drug	5538	3546	0.90	0.89	0.89	0.77	0.89	0.83	0.92	0.97	**0.94**
Form-Drug	6654	4373	0.98	0.98	**0.98**	0.85	0.95	0.90	0.96	0.97	0.96
ADE-Drug	1107	733	0.46	0.39	0.43	0.32	0.85	0.46	0.95	0.99	**0.97**
Dosage-Drug	4255	2695	0.89	0.89	0.89	0.83	0.92	0.87	0.93	0.96	**0.94**
Reason-Drug	5169	3410	0.48	0.35	0.41	0.27	0.88	0.42	0.96	0.99	**0.97**
Frequency-Drug	6310	4034	0.98	0.98	**0.98**	0.56	0.88	0.69	0.93	0.95	0.94
System (Micro)			0.88	0.83	0.86	0.69	0.90	0.78	0.93	0.95	**0.94**
System (Macro)			0.85	0.80	0.83	0.68	0.90	0.77	0.92	0.95	**0.93**

7.3.3 Comparison with Previous Work

Here, we compare our results with two previous works utilizing BERT: Wei, et al. [83] and Alimova, et al. [79] To the best of our knowledge, these are the only two works that have applied pre-trained language models of BERT on the n2c2-2018 dataset. Table 31 shows the overall Precision, Recall, and F_1 score of our fine-tuned BERT models with the reported results from the other state-of-the-art BERT-based models on the n2c2-2018 dataset. The F_1 score of all models of Wei et al's and our three models is the same, but the Precision of Wei, et al's models are higher, whereas the Recall of our models is higher. There is a notable difference between Alimova, et al. models and ours. The F_1 scores of all three models of Alimova, et al. are lower than ours, and we believe this is due to the difference in the representation of the inputs of the models.

Table 31. Overall results in comparison with previous work on the n2c2-2018 test data

	Our models				**Wei, et al. [83]**				**Alimova, et al. [79]**		
	Cased	**Uncased**	**Bio**	**Clinical**	**Cased**	**Uncased**	**Bio**	**Clinical**	**Uncased**	**Bio**	**Clinical**
Precision	0.93	0.93	0.93	0.93	**0.98**	**0.98**	**0.98**	**0.98**	**-**	**-**	**-**
Recall	**0.96**	**0.96**	0.95	0.93	0.90	0.90	0.90	0.90	-	-	-
F_1 **score**	**0.94**	**0.94**	**0.94**	0.93	**0.94**	**0.94**	**0.94**	**0.94**	0.56	0.75	0.75

Table 32. Comparison of F_1 score with previous work over each relation of the n2c2-2018 dataset.

	Our models				**Wei, et al. [83]**				**Alimova, et al. [79]**		
	Cased	**Uncased**	**Bio**	**Clinical**	**Cased**	**Uncased**	**Bio**	**Clinical**	**Uncased**	**Bio**	**Clinical**
Strength-Drug	0.87	0.87	0.88	0.84	0.98	**0.99**	0.98	**0.99**	0.58	0.68	0.68
Duration-Drug	**0.94**	**0.94**	**0.94**	**0.94**	0.88	0.89	0.88	0.89	0.41	0.66	0.65
Route-Drug	0.95	0.95	0.94	0.93	**0.97**	**0.97**	**0.97**	**0.97**	0.63	0.74	0.74
Form-Drug	0.97	0.97	0.96	0.97	0.97	**0.98**	**0.98**	**0.98**	0.62	0.81	0.81
ADE-Drug	**0.97**	**0.97**	**0.97**	**0.97**	0.80	0.80	0.81	0.81	0.10	0.62	0.62
Dosage-Drug	0.94	0.94	0.94	0.91	**0.97**	**0.97**	**0.97**	**0.97**	0.67	0.82	0.82
Reason-Drug	**0.97**	**0.97**	**0.97**	**0.97**	0.76	0.76	0.76	0.77	0.22	0.73	0.73
Frequency-Drug	0.94	0.94	0.94	0.94	**0.96**	**0.96**	**0.96**	**0.96**	0.53	0.79	0.78
	0.94	**0.94**	**0.94**	**0.94**	0.911	0.92	0.91	0.92	0.47	0.73	0.73

Table 32 shows a comparison of the F_1 score of our models the results reported by Wei, et al. [83]'s and Alimova, et al. [79] for each relation of the dataset. The results show that Alimova, et al.'s models perform lower. However, when comparing the results with Wei, et al., we found the results are complementary; relations that did not perform well with Wei, et al.'s models performed well with our models. Specifically, the relations *Reason-Drug, ADE-Drug, and Duration-Drug* obtained a higher Precision, Recall, and F_1 score than Wei, et al.'s. Meanwhile, the Precision, Recall, and F_1 score of the relation *Strength-Drug* are higher in Wei, et al. We believe this is due to three differences between our approaches: 1) Wei, et al. represent an entity-drug pair in an input sentence using the semantic type of an entity to replace the entity itself, whereas we do no such replacement; 2) they perform a multi-class classification, whereas we perform binary classification creating a separate model for each entity; and 3) Wei, et al. Clinical BERT representations were fine-tuned with MIMIC-III

over BERT (cased), whereas our representations were fine-tuned over BioBERT.

7.4 Conclusions and Contributions

Here, we evaluate the following three state-of-the-art RE approaches: a rule-based approach utilizing co-location information, a deep learning-based approach utilizing CNNs, and a BERT-based approach. Our experimental results demonstrate that the BERT-based approach outperformed other models overall and obtain state-of-the-art performance in ADE extraction with a Precision of 0.93, Recall of 0.96, and an F_1 score of 0.94; however, the rule-based approach obtained a higher Precision and Recall for certain relations. However from the rule-based results it is safe to say, co-location information is sufficient to identify most relations as the Rule-based approach obtained a higher Precision and Recall for certain relations, for e.g. Strength-Drug, Form-Drug, and Frequency-Drug.

The contributions of this work are:

1. There is a variety of RE approaches: Rule-based, Deep learning-based, and BERT-based approaches. Here we applied these approaches to an ADE dataset and analyzed the performance of each for clinical RE.

2. There are a variety of BERT-based models fine-tuned over different corpora. Here, we conducted an in-depth analysis of BERT-based models for clinical RE.

3. We achieved the state-of-the-art performance with the n2c2-2018 dataset.

4. We demonstrated that rule-based approach is sufficient for extracting relations with consistent positions.

CHAPTER 8

EXPERIMENT: UTILIZING RELATION EXTRACTION TO IDENTIFY EVENTS

A chemical reaction is an ordered sequence of reaction and workup steps that transforms a starting material into an end product [138]. Extracting these steps consists of two key tasks: Chemical NER and EE. To perform EE, we need to identify the trigger word that indicates a chemical reaction step and the relation between a trigger word and chemical compound(s) that is(are) linked to the trigger word. These relations are fundamentally different from the relations in the dataset we discussed in Chapters 6 and 7. For example, Fig. 23 in Section 4.3 shows a sentence from CLEF-2020 dataset containing three gold standard entities and two chemical events [35]. A single type of event forms between a trigger word and a chemical entity similar to the n2c2-2018 dataset; therefore, we evaluate the approaches we developed for RE to perform EE.

8.1 Methods

In this work, we use a rule-based approach discussed in Section 5.1 in Chapter 5, a CNN-based approaches discussed in Section 5.3 in Chapter 5, and a BERT-based approach discussed in Section 5.4 in Chapter 5 for EE. We focus on identifying the relation between a trigger word and a chemical compound. To identify the trigger words, we use our NER system [126] which is outside the scope of this dissertation.

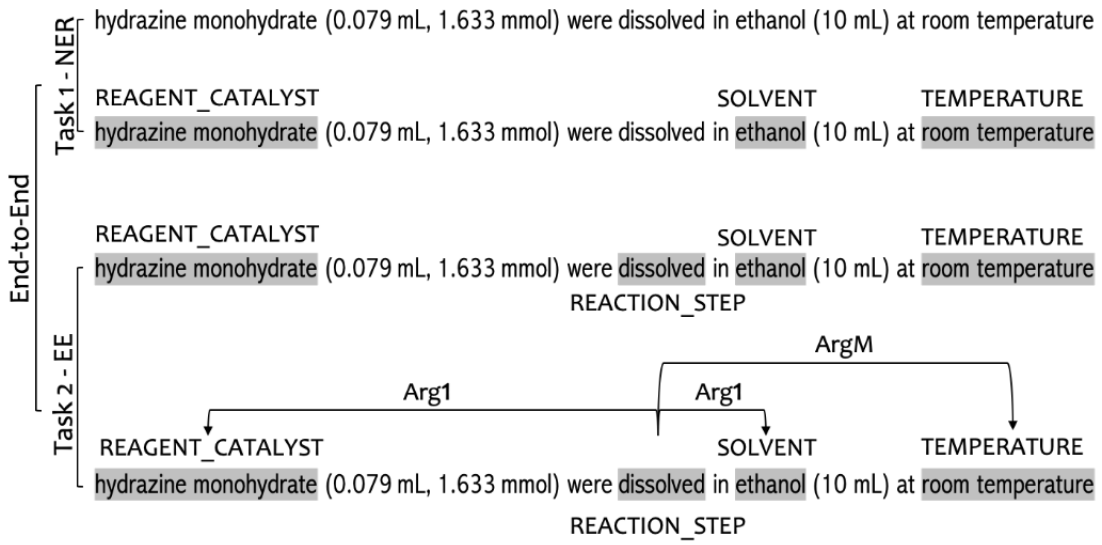

Fig. 23. Illustration of Event Extraction (EE) [35]. Shaded text spans represents annotated entities or trigger words. Arrows represent relations between entities

8.1.1 Rule-based Approach

Here, we use the left-only traversal mechanism where we traverse to the left side of the entity mention finding the closest occurrence of the trigger words.

8.1.2 CNN-based Approach

Here, we use RelEx's Segment-CNN approach, as described in Chapter 5, Section 5.3, to extract and classify the events automatically. Here, for each *Trigger word-Entity pair* we perform a binary classification to identify whether there is a relation between the trigger word and the entity or not as shown in Fig. 14 in the Section 5.3.3 which shows an abstract view of the construction of the CNN-based model for this work.

8.1.3 BERT-based Approach

Here, we explore the following BERT-based language models: BERT-cased [10], and BioBERT [80].

8.2 Experimental Design

Word representation. We explore two pre-trained word embeddings: 1) ChemPatent embeddings [132] trained over a collection of 84,076 full patent documents (1B tokens); and 2) WikiPubmed embeddings [139] in our methods.

Hyperparameters and Fine-tuning. We used Keras [140] for the implementation of the CNN architecture. We experimented with different sliding window sizes, filter sizes, loss functions for fine-tuning, and in this work, small filter sizes generated the best results for small filter sizes. We applied the dropout technique on the output of the convolution layer to regularize the model. We used *Adam* and *rmsprop* optimizers to minimize our loss function. We trained the models for 5-10 epochs to avoid overfitting.

8.3 Results and Discussion

In this section, we discuss the results of our approaches on the CLEF-2020 dataset described in Section 4.3 of Chapter 4 and compare our results across with previous works.

8.3.1 Results of Our Approaches

In this section, we discuss the results reporting the Precision, Recall, and F_1 scores of our approaches. Tables 33, 34, 35, and 36 show the results obtained from the rule-based, CNN-based, BERT-based, and BioBERT-based approaches respec-

tively. The triggers were identified using our BiLSTM+CRF method trained over the ChemPatent embeddings.

Table 33. Precision (P), Recall (R), and F_1 (F) score of the rule-based approach with trigger words identified using our BiLSTM+CRF trained with ChEMU patent embeddings

Argument	**Trigger**	**Entity**	**# Train**	**P**	**R**	**F**
ARG1	REACTION_STEP	OTHER_COMPOUND	161	0.02	0.06	0.04
		REACTION_PRODUCT	1101	0.82	0.78	0.80
		REAGENT_CATALYST	1272	0.52	0.35	0.42
		SOLVENT	1134	0.81	0.55	0.65
		STARTING_MATERIAL	1747	0.63	0.31	0.41
		Average		0.56	0.52	0.46
	WORKUP	OTHER_COMPOUND	4097	0.90	0.86	0.88
		REACTION_PRODUCT	11	0.01	1.00	0.02
		REAGENT_CATALYST	-	0.00	0.00	0.00
		SOLVENT	4	0.07	1.00	0.14
		STARTING_MATERIAL	4	0.04	1.00	0.08
		Average		0.20	0.77	0.22
ARGM	REACTION_STEP	TEMPERATURE	813	0.77	0.89	0.83
		TIME	839	0.85	0.93	0.89
		YIELD_OTHER	1043	0.83	0.80	0.81
		YIELD_PERCENT	937	0.86	0.85	0.85
		Average		0.83	0.87	0.85
	WORKUP	TEMPERATURE	242	0.66	0.81	0.73
		TIME	81	0.36	0.53	0.43
		Average		0.51	0.67	0.58
System				**0.51**	**0.72**	**0.60**

Each trigger word category shows the arithmetic mean for both trigger word classes for each entity argument class. We can see the CNN-based method performs well with the REACTION_STEP classes and poor with WORKUP classes. This is

Table 34. Precision (P), Recall (R), and F_1 (F) score of the CNN-based approach with trigger words identified using our BiLSTM+CRF trained with ChEMU patent embeddings

Argument	**Trigger**	**Entity**	**# Train**	**P**	**R**	**F**
ARG1	REACTION_STEP	OTHER_COMPOUND	161	0.00	0.00	0.00
		REACTION_PRODUCT	1101	0.92	0.96	0.94
		REAGENT_CATALYST	1272	0.78	0.69	0.74
		SOLVENT	1134	0.64	0.74	0.69
		STARTING_MATERIAL	1747	0.82	0.43	0.56
		Average		0.63	0.56	0.59
	WORKUP	OTHER_COMPOUND	4097	0.73	0.29	0.42
		REACTION_PRODUCT	11	0.00	0.00	0.00
		SOLVENT	4	0.00	0.00	0.00
		STARTING_MATERIAL	4	0.00	0.00	0.00
		Average		0.18	0.07	0.11
ARGM	REACTION_STEP	TEMPERATURE	813	0.83	0.30	0.44
		TIME	839	0.78	0.73	0.75
		YIELD_OTHER	1043	0.93	0.96	0.95
		YIELD_PERCENT	937	0.91	0.94	0.92
		Average		0.86	0.73	0.77
	WORKUP	TEMPERATURE	242	0.56	0.08	0.14
		TIME	81	0 .00	0.00	0.00
		Average		0.28	0.04	0.07
System				**0.81**	**0.54**	**0.65**

Table 35. Precision (P), Recall (R), and F_1 (F) score of the BERT-based approach with trigger words identified using our BiLSTM+CRF trained with ChEMU patent embeddings

Argument	**Trigger**	**Entity**	**# Train**	**P**	**R**	**F**
ARG1	REACTION_STEP	OTHER_COMPOUND	161	0.03	0.06	0.04
		REACTION_PRODUCT	1101	0.84	0.82	0.83
		REAGENT_CATALYST	1272	0.51	0.2	0.29
		SOLVENT	1134	0.49	0.62	0.55
		STARTING_MATERIAL	1747	0.55	0.92	0.69
		Average		0.48	0.52	0.59
	WORKUP	OTHER_COMPOUND	4097	0.54	0.48	0.51
		REACTION_PRODUCT	11	0.00	0.00	0.00
		SOLVENT	4	0.00	0.00	0.00
		STARTING_MATERIAL	4	0.00	0.00	0.00
		Average		0.14	0.12	0.13
ARGM	REACTION_STEP	TEMPERATURE	813	0.44	0.20	0.27
		TIME	839	0.51	0.82	0.63
		YIELD_OTHER	1043	0.83	0.83	0.83
		YIELD_PERCENT	937	0.84	0.92	0.88
		Average		0.66	0.69	0.65
	WORKUP	TEMPERATURE	242	0.26	0.17	0.21
		TIME	81	0.23	0.26	0.24
		Average		0.25	0.22	0.23
System				**0.58**	**0.59**	**0.58**

Table 36. Precision (P), Recall (R), and F_1 (F) score of the BioBERT-based approach with trigger words identified using our BiLSTM+CRF trained with ChEMU patent embeddings

Argument	**Trigger**	**Entity**	**# Train**	**P**	**R**	**F**
ARG1	REACTION_STEP	OTHER_COMPOUND	161	0.04	0.02	0.02
		REACTION_PRODUCT	1101	0.84	0.82	0.83
		REAGENT_CATALYST	1272	0.53	0.45	0.49
		SOLVENT	1134	0.51	0.39	0.44
		STARTING_MATERIAL	1747	0.59	0.27	0.37
		Average		0.50	0.39	0.43
	WORKUP	OTHER_COMPOUND	4097	0.52	0.53	0.54
		REACTION_PRODUCT	11	0.00	0.00	0.00
		SOLVENT	4	0.00	0.00	0.00
		STARTING_MATERIAL	4	0.00	0.00	0.00
		Average		0.13	0.13	0.14
ARGM	REACTION_STEP	TEMPERATURE	813	0.43	0.08	0.13
		TIME	839	0.57	0.30	0.40
		YIELD_OTHER	1043	0.84	0.81	0.82
		YIELD_PERCENT	937	0.84	0.88	0.86
		Average		0.67	0.52	0.56
	WORKUP	TEMPERATURE	242	0.27	0.20	0.23
		TIME	81	0.17	0.02	0.04
		Average		0.22	0.11	0.14
System				**0.62**	**0.50**	**0.55**

mainly because of the number of instances in each event class. Comparatively, most of the REACTION_STEP classes have more instances for the CNN to train than most WORKUP classes. This is the same reason the rule-based method performs better with the WORKUP classes. BERT-based methods results are similar to the CNN-based method; they perform well with the REACTION_STEP classes compared to the WORKUP classes. Since both BERT-based and CNN-based methods are supervised learning methods, they need more instances for each class to improve the results.

8.3.1.1 Error Analysis

Tables 37, 38, 39, and 40 show a detailed error analysis of our EE approaches. We report the number of true positives (tp), false positives (fp), and false negatives (fn), and also fpm and fnm, two metrics that represent the number of false positives and false negatives of the trigger words predicted.

The results are consistent with the previous observations from Tables 33, 34, 35, and 36. We can see that REACTION_STEP classes performed better than the WORKUP classes. It is safe to say that class imbalance plays a significant role in the miss-annotation of the instances. The results also show that the rule-based method significantly over annotates given the number of false positives. For example, the rule-based method identified 379 instances of the WORKUP-REACTION_PRODUCT event class, with only four being true positives. Despite having significant training instances in the REACTION_STEP classes, we can see an equally high number of false positives as true positives. This is mainly because extracting events is often trickier regardless of the sentence pattern. For example, the following sentences show a *trigger word-REACTION_PRODUCT* pair in each.

1. After cooling, the solid was collected by filtration and washed with cold dichloromethane to *give N-(4-(2-oxo-1,2,3,4-tetrahydroquinolin-6-yl)thiazol-2-yl)oxazole-5-carboxamide*

Table 37. Error analysis for the Event extraction rule-based approach where trigger words are trained with ChemPatent embeddings

Argument	**Trigger**	**Entity**	**tp**	**fp**	**fn**	**fpm**	**fnm**
ARG1	REACTION_STEP	OTHER_COMPOUND	40	1798	23	18	11
		REACTION_PRODUCT	351	75	101	10	3
		REAGENT_CATALYST	177	162	328	8	8
		SOLVENT	234	54	193	4	7
		STARTING_MATERIAL	217	128	494	15	9
	WORKUP	OTHER_COMPOUND	1501	171	249	54	73
		REACTION_PRODUCT	4	375	0	9	0
		REAGENT_CATALYST	0	40	0	9	0
		SOLVENT	2	25	0	5	0
		STARTING_MATERIAL	1	24	0	2	0
ARGM	REACTION_STEP	TEMPERATURE	450	131	53	29	15
		TIME	386	66	27	21	10
		YIELD_OTHER	350	74	85	11	3
		YIELD_PERCENT	326	55	58	11	3
	WORKUP	TEMPERATURE	89	45	21	13	20
		TIME	23	41	20	16	13
System			2984	1782	2909	204	169

Table 38. Error analysis for the Event extraction CNN-based approach where trigger words are trained with ChemPatent embeddings

Argument	Trigger	Entity	tp	fp	fn	fpm	fnm
ARG1	REACTION_STEP	OTHER_COMPOUND	0	0	63	0	11
		REACTION_PRODUCT	436	36	16	11	3
		REAGENT_CATALYST	350	97	155	17	8
		SOLVENT	316	179	111	16	7
		STARTING_MATERIAL	305	68	406	12	9
	WORKUP	OTHER_COMPOUND	516	192	1234	23	73
		REACTION_PRODUCT	0	0	4	0	0
		REAGENT_CATALYST	-	-	-	-	-
		SOLVENT	0	0	2	0	0
		STARTING_MATERIAL	0	0	1	0	0
ARGM	REACTION_STEP	TEMPERATURE	151	30	352	15	15
		TIME	300	87	113	16	10
		YIELD_OTHER	418	31	17	11	3
		YIELD_PERCENT	361	36	23	13	3
	WORKUP	TEMPERATURE	9	7	101	0	20
		TIME	0	0	43	0	13
System			3162	763	2641	134	175

Table 39. Error analysis for the Event extraction BERT-based approach where trigger words are trained with ChemPatent embeddings

Argument	Trigger	Entity	tp	fp	fn	fpm	fnm
ARG1	REACTION_STEP	OTHER_COMPOUND	4	120	59	15	11
		REACTION_PRODUCT	369	72	83	17	3
		REAGENT_CATALYST	101	97	404	9	8
		SOLVENT	266	273	161	22	7
		STARTING_MATERIAL	654	531	57	54	9
	WORKUP	OTHER_COMPOUND	845	708	905	77	73
		REACTION_PRODUCT	0	0	4	0	0
		REAGENT_CATALYST	-	-	-	-	-
		SOLVENT	0	0	2	0	0
		STARTING_MATERIAL	0	0	1	0	0
ARGM	REACTION_STEP	TEMPERATURE	101	131	402	15	15
		TIME	338	319	75	18	3
		YIELD_OTHER	360	73	75	18	3
		YIELD_PERCENT	353	65	31	17	3
	WORKUP	TEMPERATURE	19	54	91	6	20
		TIME	11	37	32	0	13
System			2984	1782	2909	204	169

Table 40. Error analysis for the Event extraction BioBERT-based approach where trigger words are trained with ChemPatent embeddings

Argument	**Trigger**	**Entity**	**tp**	**fp**	**fn**	**fpm**	**fnm**
ARG1	REACTION_STEP	OTHER_COMPOUND	0	10	63	2	11
		REACTION_PRODUCT	440	88	12	20	3
		REAGENT_CATALYST	156	146	349	13	8
		SOLVENT	256	235	171	20	7
		STARTING_MATERIAL	236	169	475	23	9
	WORKUP	OTHER_COMPOUND	928	790	822	73	68
		REACTION_PRODUCT	0	0	4	0	0
		REAGENT_CATALYST	-	-	-	-	-
		SOLVENT	0	0	2	0	0
		STARTING_MATERIAL	0	0	1	0	0
ARGM	REACTION_STEP	TEMPERATURE	40	53	463	8	15
		TIME	95	95	288	7	10
		YIELD_OTHER	352	67	83	17	3
		YIELD_PERCENT	338	65	46	17	3
	WORKUP	TEMPERATURE	22	59	88	4	19
		TIME	1	5	42	0	13
System			2984	1782	2909	204	169

(0.121 g, 87%) as a beige solid.

2. {*Methyl 4-[(6-bromo-2-phenyl-3-propylquinolin-4-yl)carbonyl]aminobicyclo[2.2.2] octane-1-carboxylate* 150 mg (0.40 mmol) of the compound from Example 38A were *dissolved* in 1.4 ml (19.8 mmol) of thionyl chloride.

In the first sentence, the entity REACTION_PRODUCT *N-(4-(2-oxo-1,2,3,4-tetrahydroquinolin-6-yl)thiazol-2-yl)oxazole-5-carboxamide* is related to the trigger word *give* but in the second sentence the entity REACTION_PRODUCT {*Methyl 4-[(6-bromo-2-phenyl-3-propylquinolin-4-yl)carbonyl]aminobicyclo[2.2.2]octane-1-carboxylate* is not related to the trigger word *dissolved*. Despite the similar sentence structure the results are not similar. These kind of instances makes the EE in this dataset quite hard.

In our EE methods, we utilized the trigger words predicted from our NER methods and the ground truth entities as the trigger words and respectively. From the metrics fpm and fnm for a trigger word-entity pair, we can see that when the number of fpm and fnm of a trigger word increases, the performance of the trigger word-entity pair decreases. We believe this is because the prediction of the trigger word-entity pair depends on the trigger word predicted by the biLSTM+CRF model [126]

8.3.2 Comparison with Previous Work

Table 41 shows a comparison between the top results reported by the CLEF ChEMU-2020 challenge using the CLEF-2020 dataset, the co-occurrence baseline provided by the organizers of the challenge, and the overall results of our EE methods. The overall results show that all three of our systems obtain a higher precision and F_1 score than the baseline but not recall. Since the baseline method is a rule-based

Table 41. Our best results in comparison with the top results of the ChEMU-2020 competition for Event extraction (EE). Baseline is provided by the organizers of the ChEMU-2020 challenge

		P	**R**	**F**
Our methods	Rule-based	0.51	0.72	0.60
	CNN-based	0.81	0.54	0.65
	BERT-based	0.58	0.59	0.58
	BioBERT-based	0.62	0.50	0.55
ChEMU_2020 teams	Melaxtech [93]	**0.96**	**0.95**	**0.95**
	NextMove/Minesoft [36]	0.94	0.86	0.90
	BOUN_REX [141]	0.76	0.69	0.72
Baseline	ChEMU organizers [138]	0.24	0.89	0.38

method based on the co-occurrence information, it obtains a high recall but low precision. Here, all systems outperform the baseline in terms of F_1 score, and Melaxtech [93] obtained the overall best performance using a hybrid combination of deep learning models and pattern-based rules for EE. NextMove/Minesoft [36] proposed a method utilizing parsing information with grammar rules, and BOUN_REX [141] utilized a set of rules to identify the events. All teams performed better than our methods except for the recall of [141].

8.4 Conclusions and Contributions

We evaluate a CNN-based approach and a rule-based approach to extract chemical reaction events from patents. The results show that the CNN-based method outperforms the rule-based methods, especially with the REACTION_STEP classes, as those classes have more instances to train on. Meanwhile, as the rule-based methods do not require training instances to train, they perform better with WORKUP

classes.

The contributions of this work are:

1. RE approaches are often domain-dependent. Here, we evaluated our CNN-based and rule-based based approaches, initially developed on clinical text, to extract relations in the scientific literature to evaluate the generalizability of our approach.
2. EE requires the extraction of multiple events across various entity classes. Here, we conducted an in-depth evaluation of utilizing RE for EE.
3. We demonstrated that rule-based approach performs well when extracting relations with consistent positions than the deep learning approaches (Eg: ARGM-yield, temperature, time)

CHAPTER 9

EXPERIMENT: EXPLORING INPUT REPRESENTATIONS FOR RELATION EXTRACTION IN BIOMEDICAL TEXT

A sentence may contain multiple entity pairs, and relations between an entity pair can belong to multiple relation classes. For example, Fig. 9 in Section 4.4 shows multiple relations between an entity pair in a sentence. The relations of this dataset are similar to the i2b2-2010 dataset (multiple relations per entity pair). It is challenging to distinguish multiple entity pairs in one sentence. Here, we explore various input entity representations to distinguish the targeted entity pair from the rest of the entity pairs in a sentence.

9.1 Methods

In this work, we explore three variations of the input entity representations using the BERT-based approach described in Section 5.4 of Chapter 5.

9.1.1 Entity representations

To determine the relation between two chemical entities, we first locate the sentence where the entity pair is located. Since one sentence can have multiple entity pairs, we explored three variations of input sentence entity representations:

1. Representation A - we input the entire sentence where the entity pair is located. Both the targeted and non-targeted entity pairs are represented as it is.
2. Representation B - we remove the non-targeted entity pairs from the input sentence. Targeted entity pairs are represented as it is.

3. Representation C - we replace the targeted entity pair with its semantic type in the input sentence. Non-targeted entity pairs are represented as it is.

For our Model-1 and Model-2 (general BERT-based models), we explore using general BERT-cased embeddings into a simple feed-forward neural network. The key difference between the Model-1 and Model-2 is the input sentence representation. For Model-1, we remove the other entity pairs in the input sentence except for the targeted entity pair. For Model-2, to represent the entity pair in an input sentence, we use the semantic type of an entity to replace the entity itself. The modified input representation is passed through the pre-trained general BERT-cased model. The output is fed into a dropout layer and then a softmax layer for multi-class classification (6). When there is no relation between a chemical and gene/protein in a sentence, we treat it as an instance of a 'No-Relation' class during the training. For our BioBERT-based model (Model-3), we explore using BioBERT embeddings into a feed-forward network for multi-class classification. Like Model-2, we represent an entity pair in a sentence by replacing the entities with the semantic types (Fig. 24[C]). The maximum input sequence length for the Model-3 is 128. We trim the sentence from both ends if a sentence is longer than the maximum sequence length. We perform this by taking the midpoint between the two entities and extending it by 64 tokens in both directions. We pass the input into the BioBERT-Large model, and embeddings of the [CLS] token are fed into a top model, consisting of a dropout and softmax layers.

9.2 Experimental Details

Tokenization: We used SpaCy and Scipy to extract input sentences and the BERT and BioBERT tokenizers to convert the sentence into tokens.

Training parameters: We used a learning rate of 2e-5 (Model-1 & 2) and 3e-5 (Model-3) and a linear learning rate schedule with $1/10^{th}$ of the total training steps

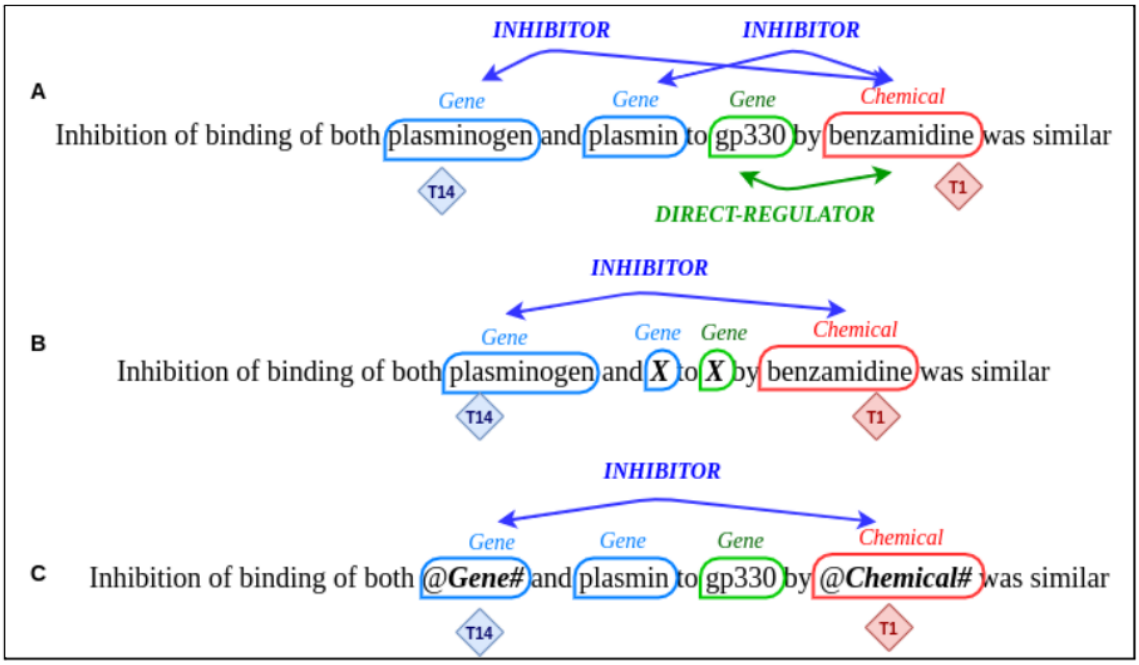

Fig. 24. Different representations of the input sentence used in our models. Model-1 utilizes the input Representation B and the Models 2 & 3 utilize the input Representation C

as a warm-up. We used a batch size of 12 for the training in all models. We applied early stopping to the training for both BioBERT-based models (six epochs) and the BERT-based models (15 epochs).

Downsampling: We downsampled the class that denotes no relation between the entity pairs by 75% to overcome the heavy class imbalance during the training in the BERT-based models.

9.3 Results and Discussion

In this section, we discuss the results of our approaches on the DrugProt dataset described in Section 4.4 of Chapter 4 and compare our results across with previous works. Table 44 and Table 45 shows the Precision, Recall, and F_1 scores for our three models on the development and test data. The bold terms indicate the best F score

of each class for development and test data.

9.3.1 Results of Development Data

Table 9.3.1 shows the Precision, Recall, and F_1 scores for our models on the development set of the DrugProt dataset. We utilized the development set to obtain the best set of weights for our model. The results show that Model-3 (BioBERT-based model) outperformed the other two models (general BERT-based models) except for two classes. Also, we can see a decrease in performance when the number of class instances decreases, especially the three classes AGONIST-ACTIVATOR, AGONIST-INHIBITOR, and SUBSTRATE_PRODUCT-OF, which have the lowest number of instances. This is mainly because these classes do not have enough instances to be differentiated from other classes during training.

Compared to Models 1 & 3, Model-2 could predict instances for the classes AGONIST-ACTIVATOR, AGONIST-INHIBITOR despite fewer training instances. We believe this is because we downsampled the 'No-Relation' (entity pairs with no relation between the entities) due to the heavy class imbalance during training. Downsampling and the input representation of the Model-2 improved the performance of the classes with few instances.

Overall performance of Model-2 is higher than Model-1, but the Recall of Model-1 is higher than Model-2 for most classes. We assume this is due to the difference in the input representation of the models. Since Model-1 eliminates the entities except for the targeted entities, the Recall is high. We experimented with both general BERT-cased and BERT-uncased, and we found that comparatively, BERT-cased performed better.

Therefore, we assume the difference in the casing of the words in the dataset played a role in determining the context of the words. Also, we experimented with

BioBERT-Base and BioBERT-Large and found that BioBERT-Large provided a performance improvement of 1.6%. Again, we assume this is because BioBERT-Large is based on BERT-Large, which has twice as many layers as BERT-base and is trained over a more extensive biomedical-based vocabulary.

Table 42. Precision (P), Recall (R), and F_1 score (F) results for all models over the development set of the DrugProt dataset

Development set	**Model 1**			**Model 2**			**Model 3**		
	P	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
INDIRECT-DOWNREGULATOR	0.48	0.69	0.57	0.62	0.67	0.64	0.75	0.73	**0.74**
INDIRECT-UPREGULATOR	0.33	0.64	0.44	0.58	0.66	0.62	0.76	0.78	**0.77**
DIRECT-REGULATOR	0.35	0.67	0.46	0.48	0.63	0.54	0.72	0.52	**0.61**
ACTIVATOR	0.32	0.61	0.42	0.53	0.63	0.58	0.78	0.75	**0.77**
INHIBITOR	0.50	0.82	0.62	0.65	0.83	0.73	0.86	0.83	**0.85**
AGONIST	0.43	0.63	0.51	0.67	0.68	0.67	0.74	0.75	**0.74**
AGONIST-ACTIVATOR	0.0	0.0	0.0	0.75	0.3	**0.43**	0.0	0.0	0.0
AGONIST-INHIBITOR	0.0	0.0	0.0	0.25	0.5	**0.33**	0.0	0.0	0.0
ANTAGONIST	0.43	0.76	0.55	0.68	0.76	0.72	0.91	0.90	**0.90**
PRODUCT-OF	0.25	0.47	0.33	0.38	0.53	0.44	0.61	0.58	**0.60**
SUBSTRATE	0.31	0.69	0.43	0.44	0.69	0.54	0.72	0.76	**0.74**
SUBSTRATE_PRODUCT-OF	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PART-OF	0.31	0.45	0.37	0.46	0.41	0.44	0.76	0.68	**0.72**
	0.39	0.69	0.50	0.56	0.69	0.62	0.78	0.74	**0.76**

9.3.2 Results of Test Data

Table 43 shows the Precision, Recall, and F_1 scores for our models on the test set of the DrugProt dataset. The observations from the results of the test set are similar to the development set. Overall, Model-3 (BioBERT-based model) outperformed the other two models except for one class. However, the overall results of the test set are lower compared to the development set. Here, also we can see a decrease in

performance when the number of class instances decreases. However, Model-2 could predict all the positive instances correctly (Precision-1.0) for the class AGONIST-INHIBITOR.

From the results of both the development and test sets, Model-2 performed better than Model-1. Therefore, it is safe to assume that replacing the entities with their semantic types is an efficient way of representation than training with the actual entity tokens. Furthermore, since the BioBERT is pre-trained on biomedical articles, it gives more efficient contextualized embeddings than the BERT trained on general English. We believe this is why Model-3 (BioBERT-based model) outperforms the other two models (general BERT-based models).

Table 43. Precision (P), Recall (R), and F_1 score (F) results for all models over the test data of the DrugProt dataset

Test set	**Model 1**			**Model 2**			**Model 3**		
	P	***R***	***F***	***P***	***R***	***F***	***P***	***R***	***F***
INDIRECT-DOWNREGULATOR	0.44	0.57	0.50	0.50	0.72	0.59	0.67	0.72	**0.70**
INDIRECT-UPREGULATOR	0.34	0.63	0.44	0.49	0.68	0.57	0.68	0.75	**0.71**
DIRECT-REGULATOR	0.34	0.55	0.42	0.41	0.6	0.48	0.70	0.57	**0.63**
ACTIVATOR	0.47	0.61	0.53	0.56	0.74	0.63	0.79	0.70	**0.74**
INHIBITOR	0.53	0.75	0.62	0.61	0.78	0.69	0.81	0.79	**0.80**
AGONIST	0.49	0.67	0.57	0.58	0.63	0.61	0.73	0.65	**0.69**
AGONIST-ACTIVATOR	0.0	0..0	0.0	0.0	0.0	**0.0**	0.0	0.0	0.0
AGONIST-INHIBITOR	0.0	0.0	0.0	1.0	0.33	**0.50**	0.0	0.0	0.0
ANTAGONIST	0.54	0.80	0.64	0.65	0.88	0.74	0.86	0.85	**0.86**
PRODUCT-OF	0.33	0.43	0.38	0.42	0.63	0.50	0.61	0.59	**0.60**
SUBSTRATE	0.42	0.44	0.43	0.38	0.53	0.44	0.61	0.55	**0.58**
SUBSTRATE_PRODUCT-OF	0.0	0..0	0.0	0.0	0..0	0.0	0.0	0.0	0.0
PART-OF	0.40	0.38	0.39	0.39	0.49	0.43	0.71	0.61	**0.66**
System	0.33	0.45	0.38	0.46	0.54	0.48	0.55	0.52	**0.54**

9.4 Conclusions and Contributions

Here, we evaluate three contextualized language-based models, a BioBERT-based and two general BERT-based models, to automatically detect relations between chemical compounds/drugs and genes/proteins. Our experimental results demonstrate that the BioBERT-based model outperformed other general BERT-based models. Therefore, we can conclude that BioBERT embeddings represent the tokens effectively when used on biomedical data. Also, replacing the entities with their semantic types is an effective unique representation of the input sentence. The contributions of this work are:

1. We built general BERT-based and BioBERT-based approaches to perform RE in biomedical domain and evaluated the effectiveness of the approaches.

2. We explored two input entity representations to generate an effective unique entity representation of the input sentence: remove the non-targeted entity pairs from the input sentence, replace the targeted entity pair with its semantic type in the input sentence.

CHAPTER 10

EXPERIMENT: UTILIZING GRAPH CONVOLUTIONAL NETWORKS FOR RELATION EXTRACTION

RE approaches extensively use techniques based on neural networks such as CNNs, RNNs, and self-attention based models such as BERT. These techniques capture the local contextual information within a sentence or document well by embedding both semantic and syntactic information in a learned representation, especially BERT [116]. The position and order of the words play a vital role in determining the context of the words in a sentence. However, their ability to capture the long-range dependency on global information in a text is limited. Utilizing the global association information between words outside the sentence boundaries can help generate better representations. Also, combining the local information with the global association information can be beneficial. Here, we propose a GCN-based approaches to effectively capture the global dependencies between terms within a corpus and combine the capability of BERT with a GCN and benefit both local and global information.

10.1 Methods

In this work, we use two GCN-based approaches described in Section 5.5 of Chapter 5: GCN-Vanilla and GCN-BERT. We treat RE task as a binary classification task building a separate model for each trigger word-entity type to determine whether a relation exists between them: 1) Positive class - there is a relation between the trigger word and the entity, 2) Negative class - there is no relation between the trigger word and the entity (no-relation).

10.1.1 GCN-Vanilla Approach

Here, we build one single graph with word and sentence nodes over the entire corpus representing a relation with a sentence node. We use a two-layer GCN allows information passing from one sentence node to another and perform a sentence node classification.

10.1.2 GCN-BERT Approach

Here, we combine GCN and BERT to capture both global and local information, and build together a final representation for relation extraction.

10.1.3 Entity representations

To determine the relation between two chemical entities, we first locate the sentence where the entity pair is located. A sentence can have multiple such entity pairs therefore we need to represent the targeted entity pair in a distinguishable way from other entity pairs. Here, we explored three variations of input sentence entity representations discussed in the section 9.1.1. Fig. 25 shows an example of an input sentence from the CLEF-2020 dataset and how a targeted entity pair is represented differently in each representation.

10.2 Results and Discussion

In this section, we discuss the results of our approaches on the the following benchmark datasets: CLEF-2020 dataset(Chapter 4, Section 4.3), n2c2-2018 dataset (Chapter 4, Section 4.2). Also we compare our results across with previous works.

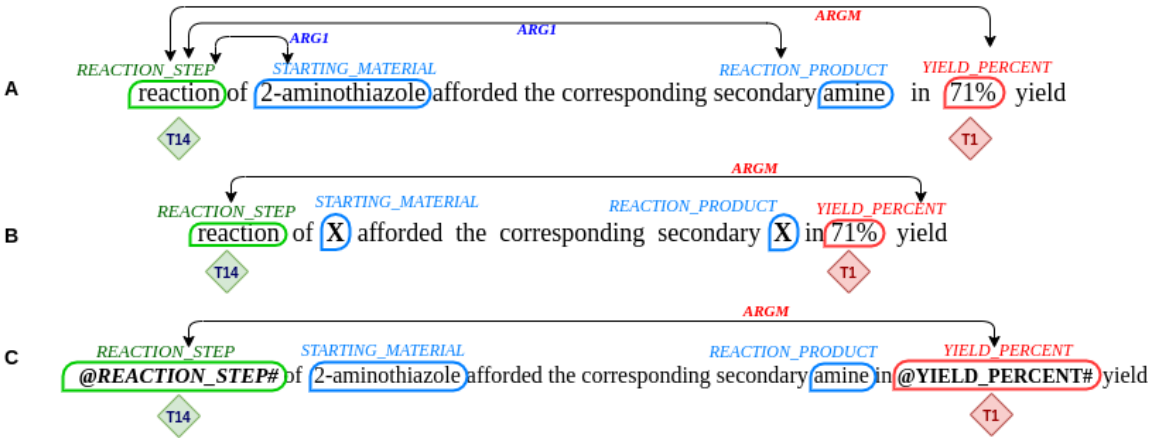

Fig. 25. Different representations of the entity pair in the input sentence from CLEF-2020 dataset used in our models.

10.2.1 Results over CLEF-2020 dataset

In this section, we discuss the results on the development and test set of CLEF-2020 dataset and compare the performance of the approaches with our previous work [126].

10.2.1.1 Development set results

Table 44 shows precision (P), recall (R), and F_1 (F) scores on the development set of the CLEF-2020 dataset for each of our architectures across the three input representations described in Section 10.1.3. The overall results show that the GCN-BERT approach outperformed the GCN-Vanilla approach for all three input representations. Furthermore, the GCN-BERT approach obtained the highest precision, recall, and F_1 scores for all the relations of both REACTION_STEP and WORKUP classes.

Both approaches obtained higher F_1 scores with the REACTION_STEP classes than WORKUP classes. This is mainly because the REACTION_STEP classes have more training instances than most WORKUP classes. Still, the GCN-BERT approach improved the performance of the classes with fewer training instances than the GCN-

Table 44. Precision (P), Recall (R), and F_1 (F) scores on the development set of CLEF-2020 dataset with trigger words identified using our NER system [123])

Method	Argument	Trigger	Entity	# Train	Representation A			Representation B			Representation C		
					P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
GCN-Vanilla	ARG1	REACTION_STEP	OTHER_COMPOUND	161	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			REACTION_PRODUCT	1101	0.86	0.97	0.91	0.87	0.96	0.91	0.87	0.95	0.91
			REAGENT_CATALYST	1272	0.61	0.73	0.67	0.62	0.70	0.66	0.61	0.72	0.66
			SOLVENT	1134	0.63	0.73	0.68	0.63	0.77	0.69	0.59	0.75	0.66
			STARTING_MATERIAL	1747	0.65	0.75	0.70	0.66	0.75	0.70	0.66	0.73	0.70
			Average		*0.55*	*0.64*	*0.59*	*0.56*	*0.64*	*0.59*	*0.55*	*0.63*	*0.59*
		WORKUP	OTHER_COMPOUND	4097	0.64	0.66	0.65	0.65	0.75	0.69	0.65	0.72	0.68
			REACTION_PRODUCT	11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			REAGENT_CATALYST	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			SOLVENT	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			STARTING_MATERIAL	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			Average		*0.13*	*0.13*	*0.13*	*0.13*	*0.15*	*0.14*	*0.13*	*0.14*	*0.14*
	ARGM	REACTION_STEP	TEMPERATURE	813	0.56	0.36	0.44	0.56	0.36	0.44	0.64	0.32	0.43
			TIME	839	0.56	0.62	0.59	0.58	0.67	0.62	0.59	0.58	0.59
			YIELD_OTHER	1043	0.88	0.95	0.91	0.88	0.95	0.91	0.88	0.95	0.91
			YIELD_PERCENT	937	0.87	0.96	0.91	0.87	0.96	0.91	0.87	0.96	0.91
			Average		*0.72*	*0.73*	*0.71*	*0.73*	*0.74*	*0.72*	*0.75*	*0.70*	*0.71*
		WORKUP	TEMPERATURE	242	0.67	0.07	0.13	0.73	0.44	0.55	0.75	0.33	0.46
			TIME	81	0.50	0.14	0.22	0.00	0.00	0.00	0.00	0.00	0.00
			Average		*0.59*	*0.11*	*0.18*	*0.37*	*0.22*	*0.28*	*0.38*	*0.17*	*0.23*
	System				**0.68**	**0.70**	**0.68**	**0.69**	**0.73**	**0.71**	**0.69**	**0.71**	**0.71**
GCN-BERT	ARG1	REACTION_STEP	OTHER_COMPOUND	161	0.54	0.62	0.58	0.00	0.00	0.00	0.82	0.67	0.74
			REACTION_PRODUCT	1101	0.96	0.87	0.91	0.99	0.95	0.97	0.99	0.96	0.97
			REAGENT_CATALYST	1272	0.77	0.50	0.61	0.97	0.91	0.94	0.96	0.92	0.95
			SOLVENT	1134	0.82	0.51	0.63	0.95	0.88	0.91	0.95	0.91	0.93
			STARTING_MATERIAL	1747	0.90	0.49	0.63	0.95	0.84	0.89	0.95	0.85	0.90
			Average		0.80	0.60	0.67	0.77	0.72	0.74	0.93	0.86	0.90
		WORKUP	OTHER_COMPOUND	4097	0.82	0.52	0.64	0.96	0.92	0.94	0.97	0.92	0.95
			REACTION_PRODUCT	11	0.50	0.33	0.40	0.00	0.00	0.00	0.00	0.00	0.00
			SOLVENT	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			STARTING_MATERIAL	4	0.66	0.37	0.48	0.00	0.00	0.00	0.00	0.00	0.00
			Average		*0.50*	*0.31*	*0.71*	*0.24*	*0.23*	*0.24*	*0.24*	*0.23*	*0.24*
	ARGM	REACTION_STEP	TEMPERATURE	813	0.66	0.37	0.48	0.90	0.50	0.65	0.95	0.54	0.69
			TIME	839	0.74	0.43	0.55	0.95	0.76	0.85	0.92	0.79	0.85
			YIELD_OTHER	1043	0.92	0.91	0.91	0.98	0.94	0.96	0.98	0.93	0.96
			YIELD_PERCENT	937	0.96	0.87	0.91	0.99	0.95	0.97	0.98	0.96	0.97
			Average		0.82	0.65	0.71	0.96	0.79	0.86	0.96	0.81	0.87
		WORKUP	TEMPERATURE	242	0.52	0.71	0.60	1.00	0.85	0.92	0.96	0.89	0.92
			TIME	81	0.33	0.29	0.31	1.00	0.71	0.83	1.00	0.71	0.83
			Average		*0.43*	*0.50*	*0.46*	*1.00*	*0.78*	*0.88*	*0.98*	*0.80*	*0.88*
	System				**0.83**	**0.59**	**0.69**	**0.96**	**0.85**	**0.91**	**0.96**	**0.87**	**0.91**

Vanilla approach. This indicates that combining the local information of the text with global information provides additional information for the classification layer than just considering the global information only, especially for the classes with fewer training instances in a class.

The overall analysis of the various input representations showed that Representations B and C obtained similar precision, recall, and F_1 score for both approaches and comparatively higher F_1 score than representation A. This is precisely the case with the GCN-BERT approach. When multiple entity pairs are present in an input sentence, Representation B removes the non-targeted entity pairs, and Representation C replaces the entity pair is with its semantic type, whereas Representation A passes sentence as it is. This indicates that the performance increases when the targeted entities are distinguished from the non-targeted entities. In the GCN-Vanilla approach, there is not a notable difference between the performance of the Representation B and C with A. However, in the GCN-BERT approach, we can see a significant increase in the performance of both Representations B and C compared to A. If we compare the performance of the WORKUP classes, we can see that Representation A managed to predict instances correctly to an extent when both Representations B and C failed to predict any. This shows that Representation A performs better when the training instances are low in a class.

10.2.1.2 Test set results

Table 45 shows precision (P), recall (R), and F_1 (F) scores on the test set of the CLEF-2020 dataset for three input representations mentioned in 10.1.3.

The observations from the test set results are similar to the development set for both approaches. The main observation is that the GCN-BERT approach obtained higher F_1 scores over GCN-Vanilla for all representations except Representation A.

Table 45. Precision (P), Recall (R), and F_1 (F) scores on the development set of CLEF-2020 dataset with trigger words identified using our previous NER model [123]

Method	Argument	Trigger	Entity	# Train	Representation A			Representation B			Representation C		
					P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
GCN-Vanilla	ARG1	REACTION_STEP	OTHER_COMPOUND	161	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			REACTION_PRODUCT	1101	0.85	0.96	0.90	0.85	0.96	0.90	0.85	0.95	0.91
			REAGENT_CATALYST	1272	0.58	0.73	0.65	0.61	0.71	0.65	0.59	0.68	0.63
			SOLVENT	1134	0.58	0.70	0.64	0.58	0.75	0.65	0.58	0.69	0.63
			STARTING_MATERIAL	1747	0.61	0.76	0.68	0.61	0.77	0.68	0.61	0.76	0.68
			Average		*0.52*	*0.47*	*0.51*	*0.87*	*0.79*	*0.82*	*0.64*	*0.65*	*0.63*
		WORKUP	OTHER_COMPOUND	4097	0.59	0.68	0.63	0.62	0.75	0.68	0.63	0.67	0.65
			REACTION_PRODUCT	11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			REAGENT_CATALYST	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			SOLVENT	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			STARTING_MATERIAL	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			Average		*0.18*	*0.09*	*0.12*	*0.19*	*0.17*	*0.18*	*0.19*	*0.18*	*0.18*
	ARGM	REACTION_STEP	TEMPERATURE	813	0.55	0.38	0.45	0.56	0.38	0.45	0.61	0.34	0.44
			TIME	839	0.56	0.63	0.59	0.61	0.64	0.62	0.60	0.58	0.59
			YIELD_OTHER	1043	0.85	0.97	0.91	0.85	0.97	0.91	0.85	0.97	0.91
			YIELD_PERCENT	937	0.85	0.96	0.90	0.85	0.96	0.91	0.86	0.95	0.90
			Average		*0.70*	*0.74*	*0.71*	*0.72*	*0.74*	*0.72*	*0.73*	*0.71*	*0.71*
		WORKUP	TEMPERATURE	242	0.00	0.00	0.00	0.62	0.21	0.31	0.56	0.13	0.21
			TIME	81	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			Average		*0.00*	*0.00*	*0.00*	*0.31*	*0.11*	*0.16*	*0.28*	*0.07*	*0.11*
	System				**0.65**	**0.70**	**0.67**	**0.66**	**0.73**	**0.69**	**0.67**	**0.69**	**0.68**
GCN-BERT	ARG1	REACTION_STEP	OTHER_COMPOUND	161	0.43	0.59	0.50	0.60	0.48	0.53	0.00	0.00	0.00
			REACTION_PRODUCT	1101	0.97	0.85	0.90	0.93	0.90	0.91	0.89	0.90	0.89
			REAGENT_CATALYST	1272	0.00	0.00	0.00	0.94	0.87	0.90	0.55	0.78	0.64
			SOLVENT	1134	0.87	0.40	0.54	0.93	0.85	0.89	0.82	0.70	0.73
			STARTING_MATERIAL	1747	0.78	0.50	0.61	0.95	0.84	0.89	0.96	0.88	0.91
			Average		*0.61*	*0.47*	*0.51*	*0.87*	*0.79*	*0.82*	*0.64*	*0.65*	*0.63*
		WORKUP	OTHER_COMPOUND	4097	0.88	0.43	0.58	0.95	0.85	0.89	0.94	0.88	0.91
			REACTION_PRODUCT	11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			REAGENT_CATALYST	-	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			SOLVENT	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			STARTING_MATERIAL	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
			Average		*0.44*	*0.22*	*0.29*	*0.48*	*0.43*	*0.45*	*0.47*	*0.44*	*0.45*
	ARGM	REACTION_STEP	TEMPERATURE	813	0.55	0.30	0.39	0.90	0.50	0.64	0.89	0.55	0.68
			TIME	839	0.72	0.41	0.52	0.88	0.72	0.79	0.90	0.77	0.83
			YIELD_OTHER	1043	0.88	0.94	0.91	0.99	0.97	0.98	0.94	0.89	0.91
			YIELD_PERCENT	937	0.85	0.96	0.90	0.99	0.92	0.96	0.87	0.93	0.90
			Average		*0.75*	*0.65*	*0.68*	*0.94*	*0.78*	*0.84*	*0.90*	*0.79*	*0.83*
		WORKUP	TEMPERATURE	242	0.00	0.00	0.00	0.87	0.61	0.72	0.00	0.00	0.00
			TIME	81	0.00	0.00	0.00	0.00	0.00	0.00	0.72	0.49	0.58
			Average		*0.00*	*0.00*	*0.00*	*0.44*	*0.31*	*0.36*	*0.36*	*0.25*	*0.29*
	System				**0.82**	**0.48**	**0.61**	**0.94**	**0.81**	**0.87**	**0.87**	**0.75**	**0.81**

Also, GCN-BERT approach obtained the highest precision, recall, and F_1 scores for all the relations of both REACTION_STEP and WORKUP classes. The notable increase in the performance of the GCN-BERT shows the advantage of combining BERT with GCN allowing interactions between the local contextual and global association information. Comparing the results of the REACTION_STEP classes than WORKUP classes, we see that both approaches obtained higher F_1 scores with the REACTION_STEP classes than WORKUP classes. This is because the REACTION_STEP classes have more training instances than most of the WORKUP classes, therefore they can differentiate themselves when training than the WORKUP classes. Despite the lower number of training instances in the WORKUP classes, GCN-BERT comparatively obtained higher F_1 scores than the GCN-Vanilla. This indicates combining the local information of the text with global information provides additional information for the classification layer than just considering the global information only, especially for the classes with fewer training instances in a class.

We use various input entity representations to distinguish the targeted entity pairs. When multiple entity pairs are present in an input sentence, Representation B removes the non-targeted entity pairs, and Representation C replaces the entity pair is with its semantic type, whereas Representation A passes sentence as it is. The overall analysis of the various input entity representations showed that Representation B outperformed the other two input representations by obtaining a higher F_1 score, which shows that masking non-targeted entities helped extract essential information to identify the classes better. All representations obtained similar precision, recall, and F_1 score with the GCN-Vanilla but Representations B and C obtained comparatively higher precision, recall, and F_1 score with the GCN-BERT. This indicates that the performance increases when the targeted entities are distinguished from the non-targeted entities. Since most of the input sentences in the test set are quite long,

we can find multiple entity pairs in one sentence. Therefore, we believe masking non-targeted entities (Representations B) or replacing targeted entities by their semantic types (Representations C) provides a better classification representation.

10.2.1.3 Comparison with previous work

In this section, we compare our approaches with our previous approaches and with state-of-the-art approaches. Table 46 shows a comparison between the top results reported by the CLEF ChEMU-2020 challenge using the CLEF-2020 dataset, the co-occurrence baseline provided by the organizers of the challenge, best overall results of our previous approaches [126] and best results of our current approaches. Bold terms show the best results in each category.

Table 46. Our best results in comparison with our previous results and the top results of the ChEMU-2020 competition for Event Extraction (EE). Baseline is provided by the organizers of the ChEMU-2020 challenge

		P	**R**	**F**
Our current methods	GCN-Vanilla	0.66	0.73	0.69
	GCN-BERT	**0.94**	**0.81**	**0.87**
Our previous methods	Rule-based	0.51	**0.72**	0.60
	CNN-based	**0.81**	0.54	**0.65**
	BERT-based	0.58	0.59	0.58
	BioBERT-based	0.62	0.50	0.55
ChEMU_2020 teams	Melaxtech [93]	**0.96**	**0.95**	**0.95**
	NextMove/Minesoft [36]	0.94	0.86	0.90
	BOUN_REX [141]	0.76	0.69	0.72
Baseline	ChEMU organizers [132]	0.24	0.89	0.38

Comparison of the results between our current and previous approaches shows

that the GCN-BERT outperformed all other approaches and obtained the highest overall precision, recall, and F_1 score. In particular, it outperformed both GCN and BERT alone, which confirmed the advantage of combining them.

Among the models that only use local information, CNN performed better than BERT. If we compare the CNN-based approach and the GCN-Vanilla based approach, we can see the GCN-Vanilla based approach obtained higher recall and F_1 scores but not precision. CNN and BERT capture the local information between words better, whereas GCN captures the global information better. This shows that capturing the global information is beneficial for classifying the relations in the CLEF-2020 dataset. The superior performance of GCN-BERT shows that combining both BERT and GCN and allowing interactions between the two types of information is beneficial.

The baseline provided by the organizers of the ChEMU-2020 challenge obtained a higher recall than our current approaches. Since the baseline is a rule-based approach based on the co-occurrence information, it obtains a high recall but low precision. Our approaches outperformed the baseline in terms of precision and F_1 score. For the top three performing systems in the CLEF ChEMU-2020 challenge, Melaxtech [93] obtained the overall highest F_1 score. They used a hybrid approach combining a deep learning model with pattern matching rules. NextMove/Minesoft [36] utilized parsing information with grammar rules, and BOUN_REX [141] utilized a set of rules to identify the relations. Both Melaxtech [93] and NextMove/Minesoft [36] obtained higher F_1 scores than our approaches. In the future, we plan to explore integrating rule-based information into our GCN-BERT based approach.

10.2.2 Results over n2c2-2018 dataset

In this section, we discuss the results on the test set of n2c2-2018 dataset and compare the performance of the approaches with our previous work [125].

10.2.2.1 Test set results

Table 47. Precision (P), Recall (R), and F_1 (F) scores on the test set of n2c2-2018 dataset for our GCN-based approaches.

		Representation A			**Representation B**			**Representation C**		
		P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
GCN-Vanilla	Strength-Drug	0.86	0.93	0.89	0.86	0.93	0.89	0.86	0.93	0.89
	Duration-Drug	0.90	0.45	0.60	0.90	0.45	0.60	0.90	0.45	0.60
	Route-Drug	0.90	0.78	0.83	0.90	0.77	0.83	0.90	0.78	0.83
	Form-Drug	0.96	0.86	0.91	0.96	0.86	0.91	0.96	0.86	0.91
	ADE-Drug	0.85	0.34	0.49	0.85	0.34	0.49	0.85	0.34	0.49
	Dosage-Drug	0.92	0.86	0.89	0.92	0.86	0.89	0.92	0.86	0.89
	Reason-Drug	0.88	0.30	0.44	0.88	0.30	0.44	0.88	0.30	0.44
	Frequency-Drug	0.90	0.73	0.81	0.90	0.73	0.81	0.90	0.73	0.81
	System (Micro)	**0.90**	**0.73**	**0.81**	**0.90**	**0.73**	**0.81**	**0.90**	**0.73**	**0.81**
	System (Macro)	**0.89**	**0.72**	**0.79**	**0.90**	**0.73**	**0.81**	**0.90**	**0.73**	**0.81**
GCN-BERT	Strength-Drug	0.91	0.90	0.91	0.97	0.89	0.93	0.95	0.92	0.94
	Duration-Drug	0.90	0.45	0.60	0.90	0.45	0.60	0.90	0.45	0.60
	Route-Drug	0.91	0.77	0.83	0.97	0.72	0.83	0.94	0.76	0.84
	Form-Drug	0.96	0.86	0.91	0.96	0.86	0.91	0.97	0.85	0.91
	ADE-Drug	0.86	0.34	0.49	0.85	0.34	0.49	0.86	0.33	0.48
	Dosage-Drug	0.94	0.85	0.89	0.99	0.83	0.90	0.96	0.86	0.91
	Reason-Drug	0.88	0.29	0.44	0.88	0.30	0.44	0.88	0.30	0.44
	Frequency-Drug	0.92	0.73	0.81	0.97	0.68	0.80	0.94	0.73	0.82
	System (Micro)	**0.92**	**0.72**	**0.81**	**0.96**	**0.70**	**0.81**	**0.95**	**0.73**	**0.82**
	System (Macro)	**0.91**	**0.71**	**0.79**	**0.95**	**0.69**	**0.80**	**0.94**	**0.71**	**0.80**

Table 47 shows precision (P), recall (R), and F_1 (F) scores on the test set of the n2c2-2018 dataset for each of our architectures across the three input representations described in Section 10.1.3. The overall results show that both approaches achieved a similar F_1 score on this dataset for all three input representations and achieved an overall precision of 0.95, recall of 0.71, and F_1 score of 0.82. However, GCN-BERT approach achieved comparatively higher precision and lower recall than GCN-Vanilla.

When comparing the results of each relation class, we can see that all classes achieved higher precision, recall, and F_1 score except for the classes Duration-Drug, ADE-Drug, and Reason-Drug. We think this is mainly because by incorporating global association information of the words in the sentence, the model is given irrelevant information, which confused the classifier more than it helped. Analysis of the various input representation showed that differences in the input sentence representation did not help in improving the classification results either. Our supposition for this observation is that multiple non-drug entities in a sentence can distinguish themselves except ADE-Drug, and Reason-Drug. ADE and Reason both include symptoms and incorporating global association information was insufficient to identify the correct ADE or Reason when multiple drugs were in the same sentence.

The GCN-BERT approach achieved slightly higher precision and lower recall than the GCN-Vanilla approach. GCN-BERT takes in both the local contextual and global association information, whereas GCN-Vanilla considers only the global association information. This observation again proves that considering the local information alone is sufficient to extract the information required to determine the relation class.

10.2.2.2 Comparison with Our Previous Work

In this section, we compare our current approaches with our previous approaches [125]. Table 48 reports precision (P), recall (R), and F_1 (F) scores on the test set of the n2c2-2018 dataset and shows a comparison between our approaches:1) rule-based approach using left-only traversal mechanism; 2) CNN-based approach using Segment-CNN, and 3) contextualized language model-based approach using BioBERT, 4) GCN-Vanilla approach using Representation C, and 5) GCN-BERT approach using Representation C. Bold terms show the best results for each relation class.

Table 48. Comparison across our approaches over the n2c2-2018 test set

	Rule-based			**Segment-CNN**			**BioBERT**			**GCN-Vanilla**			**GCN-BERT**		
	P	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Strength-Drug	0.96	0.95	**0.95**	0.91	0.88	0.90	0.86	0.90	0.88	0.86	0.93	0.89	0.95	0.92	0.94
Duration-Drug	0.78	0.69	0.73	0.39	0.90	0.55	0.96	0.93	**0.95**	0.90	0.45	0.60	0.90	0.45	0.60
Route-Drug	0.90	0.89	0.89	0.77	0.89	0.83	0.92	0.97	**0.94**	0.90	0.78	0.83	0.94	0.76	0.84
Form-Drug	0.98	0.98	**0.98**	0.85	0.95	0.90	0.96	0.97	0.96	0.96	0.86	0.91	0.97	0.85	0.91
ADE-Drug	0.46	0.39	0.43	0.32	0.85	0.46	0.95	0.99	**0.97**	0.85	0.34	0.49	0.86	0.33	0.48
Dosage-Drug	0.89	0.89	0.89	0.83	0.92	0.87	0.93	0.96	**0.94**	0.92	0.86	0.89	0.96	0.86	0.91
Reason-Drug	0.48	0.35	0.41	0.27	0.88	0.42	0.96	0.99	**0.97**	0.88	0.30	0.44	0.88	0.30	0.44
Frequency-Drug	0.98	0.98	**0.98**	0.56	0.88	0.69	0.93	0.95	0.94	0.90	0.73	0.81	0.94	0.73	0.82
System (Micro)	0.88	0.83	0.86	0.69	0.90	0.78	0.93	0.95	**0.94**	0.90	0.73	0.81	0.95	0.73	0.82
System (Macro)	0.85	0.80	0.83	0.68	0.90	0.77	0.92	0.95	**0.93**	0.90	0.73	0.81	0.94	0.71	0.80

Both rule-based and BERT-based approaches obtained overall higher F_1 scores than our GCN-based approaches. BERT-based approach outperformed all other approaches except for the Strength-Drug, Frequency-Drug, and Form-Drug pairs which performed well with the rule-based approach. However, ADE-Drug, Reason-Drug performed poorly with both rule and GCN-based approaches. From this, we can say the co-location and global association information are insufficient to identify the correct ADE or Reason when multiple drugs were in the same sentence.

Comparing our GCN-based approaches with our CNN-based approach shows that both GCN-based approaches achieved a higher F_1 score, but the precision of both GCN-based approaches is higher, whereas the recall of the CNN-based approach is higher. We believe this is because of the difference between how CNN and GCN works. CNN is a traditional neural network that learns features by a specific order of words, whereas GCN generates graphs from non-structural data and learns the association between words that is not within the given window. From this observation, it is safe to assume that capturing the information regarding the order of the words helps the classifier improve the recall, whereas capturing the global information of the words helps improve the classifier's recall. When comparing the results of the GCN-BERT approach to the BERT-based approach, we can see the GCN-BERT obtained higher precision, but since the recall is relatively low, GCN-BERT obtained a lower F_1 score.

From the comparison of the results, it is clear that the BERT-based approach achieves the best performance. Therefore, we can conclude that extracting the local contextual information is sufficient to extract relations in this dataset.

10.3 Conclusions and Contributions

Here, we evaluated two approaches based on GCN for RE: GCN-Vanilla and GCN-BERT approaches. GCN-Vanilla utilizes GCN to capture the global struc-

ture information in graph embeddings. In contrast, GCN-BERT combines GCN and BERT to integrate the local contextual and global information between the words. We also explored three input entity representations with both approaches. From the results, we can conclude that combining GCN and BERT and allowing both types of information to interact through the layers of attention mechanism can be beneficial compared to using BERT and GCN alone. We also found that replacing the targeted entities with their semantic types or masking the non-targeted entities in a sentence effectively provides unique entity representations of an input sentence. We found combining GCN with BERT performs well with datasets which have more information on the targeted entities outside the selected scope of the sentence.

The contributions of this work are:

1. We built two GCN-based approaches to perform RE to capture the global structure information in graph embeddings and evaluated the effectiveness of the approaches.

2. We explored whether combining GCN and BERT capturing both local and global information provides a better representation of the words in the text.

3. Determined combination of global and local information is beneficial to identify the chemical relations but not for the clinical relations.

CHAPTER 11

OVERALL CONCLUSIONS AND CONTRIBUTIONS

In this dissertation, we explored different approaches for RE (a rule-based approach utilizing co-location information, a machine learning-based approach with extracted features, a deep learning-based approach utilizing CNNs, a BERT-based approach, and two GCN-based approaches) across different types of relations. We explored each approach, conducted an in-depth analysis, and evaluated the performance of that approach over various datasets.

11.1 Conclusions

We came to the following conclusions from the work we have done thus far:

1. All approaches apply to data from any domain to determine whether a relation exists between one or a few selected entities.

2. Rule-based approach is applicable for data with fewer instances and for relations with consistent positional information. The number of instances in the training set does not affect the rule-based methods.

3. Rule-based approaches perform well with data which is more structured as it is convenient to generate rules.

4. Machine learning and deep learning-based approaches are applicable for labeled data with many training instances.

5. Deep learning-based approaches automatically identify patterns in the data efficiently, whereas rule-based approaches require expert knowledge to generate

rules.

6. BERT-based approaches utilize contextualized word embeddings to convert text into vectors, and they perform better than approaches that use non-contextualized word embeddings.

7. Using pre-trained language representations to fine-tune models is advantageous. They use minimal task-specific parameters and are trained on the downstream tasks by simply fine-tuning all the pre-trained parameters.

8. Deep learning and BERT-based approaches capture the local contextual information within a sentence or document well by embedding both semantic and syntactic information in a learned representation. However, their ability to capture the long-range dependency global information in a text is limited.

9. BioBERT embeddings represent the tokens effectively when used on biomedical data.

10. GCN-based approaches capture the global association information between words that are not next to each other(long range).

11. Combining GCN and BERT and allowing both types of information to interact through the layers of attention mechanism is beneficial compared to using BERT and GCN alone on certain relations.

12. Replacing the targeted entities with their semantic types or masking the non-targeted entities in a sentence provides effective unique entity representations of an input sentence.

13. Hierarchical classification approach alleviates the class imbalance caused by negative relations.

11.2 Contributions

The overall contributions of our dissertation are:

- We developed a rule-based approach using the co-location information of the words.
- We developed a feature-vector based machine learning approach to extract and classify explicit semantic relations in scientific papers.
- We developed three CNN-based architectures for binary, multi-class, and multi-label classification.
- We developed BERT-based approaches that utilized contextualized word embeddings for a better representation of word-vector.
- We developed a GCN-based approach to utilize the global association information to capture the global dependencies between words effectively.
- We developed an approach that combined the capability of GCN and BERT to benefit from capturing both local and global information and allowing them to influence mutually and build together a final representation.
- We explored various input entity representations and generated an effective unique entity representation for an input sentence.
- We built a hierarchical classification schema for the highly imbalanced data to alleviate the class imbalance caused by negative relations.

CHAPTER 12

FUTURE WORK

In this chapter, we discuss the directions we are interested in taking in the future for RE. We have discussed five different RE approaches so far that focus mainly on extracting information to determine the relations between the targeted entities. In the traditional pipeline models, NER is performed initially to identify and label the entities, and RE is performed next to extract and classify relations [142]. Joint entity relation extraction performs NER and RE simultaneously and utilizes the dependency between the two tasks to decrease the error propagation issue in the pipeline model [143]. Named entities are essential to extract relations. Accurate relation classification helps recognize named entities [144]. This hybrid framework can increase the performance of the RE. For example, we can build a Bi-LSTM module for entity extraction and a CNN module for RE [142]. The contextual information obtained on the entities through the BiLSTM would be sent through CNN, which would help improve the RE.

Latest approaches in RE focus on capturing only the semantic relationships between the entities to determine the relations, whereas they require local and global syntactic and semantic dependencies [107, 113]. In the future, we would like to combine semantic information with syntactic information together, especially the dependency trees. Syntactic features analyze the structural role of words in a text with formal grammar rules such as POS tags and n-grams. They target the roles played by each word in a sentence and try to interpret the relation between words and the grammatical structure of sentences. An example of syntactic processing is Depen-

dency Parsing (DP). DP is the process of studying the dependencies between the words or phrases of a sentence to identify its grammatical structure. GCN applied on a graph captures the global structure information in graph embeddings. Applying GCN on the dependency tree would further improve the performance of GCN as it incorporates additional structural information into it, and they have shown to be effective in capturing long-range relations between target entities [113]. Furthermore, we pass the outputs of the GCN through an attention mechanism.

In our approaches, we have used the following transformer-based models: general BERT [10], BioBERT [80], and Clinical BERT [81]. Yang, et al. [78] compared BERT, RoBERTa, and XLNet and reported XLNet-based clinical model achieved the best F_1 score on the n2c2-2018 dataset. In our approaches, BERT based models performed well with the n2c2-2018 dataset [125] but not with the CLEF-2020 dataset [126]. In the future, we plan to explore other transformer-based models too.

In deep neural network architectures introducing more layers can help to reduce the error rate. However, when we increase the number of layers a lot, we would encounter a common problem most of the deep learning models face, the *vanishing/exploding gradient* problem. This would lead to an increase in the training and test error rate [145]. Due to this problem, we limit the layers to two or three in our CNN or GCN-based approaches. In 2015, Microsoft Research introduced a new architecture called Residual Network (ResNet) to solve the problem of the vanishing/exploding gradient [88]. ResNet helps to stack additional layers, which results in improved performance. In the future, we can explore building models with more layers which could improve the performance of RE, especially when the input sentences are long.

In the future, we will explore more ways to improve the performance of our existing approaches. Hybrid approaches that include a deep neural network and a

rule-based model have proved to increase the performance with some datasets [93]. Therefore, we plan to explore building a hybrid model with either BERT or GCN and rule-based methods to increase performance. Also, we plan to build a model that trains GCN and BERT separately and then concatenates the graph and BERT embeddings before feeding them through the final classification layer. We utilized a custom-built word-integer mapping to represent a word node in the vocabulary graph in the GCN-BERT approach. Also, we used random weight vectors initially to generate the graph embeddings. In the future, we would like to use an external pre-built vocabulary and explore the performance of the graph embeddings with various external pre-trained word embeddings.

Appendix A

ABBREVIATIONS

NLP	Natural Language Processing
IE	Information Extraction
NER	Named Entity Recognition
RE	Relation Extraction
EE	Event Extraction
ADE	Adverse Drug Events
BFS	Breadth-First Search
MIMIC	Medical Information Mart for Intensive Care
SVM	Support Vector Machines
RF	Random Forests
NB	Naive Bayes
TF-IDF	Term Frequency-Inverse Document Frequency
CRF	Conditional Random Fields
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
BERT	Bidirectional Encoder Representations from Transformers
GPT	Genera-tive Pre-trained Transformer
GNN	Graph Neural Network
GCN	Graph Convolutional Network
HMM	Hidden Markov Models
POS	Part-Of-Speech
BOW	Bag of Words

UMLS	Unified Medical Language System
GloVe	Global Vectors
CBOW	Continuous Bag of Words Model
CUI	Concept Unique Identifier
CV	Cross Validation
SDP	Shortest Dependency Path
MLP	Multi-Layer Perceptron
BRAT	Brat Rapid Annotation Tool
PMI	Point-wise Mutual Information
OOV	Out-of-vocabulary
LAM	Logical Adjacency Matrix
CDR	Chemical-disease relation
MLM	Masked Language Modeling
NSP	Next Sentence Prediction
ResNet	Residual Network
GRU	Gated Recurrent Unit
SST	Stanford Sentiment Treebank
CoLA	Corpus of Linguistic Acceptability
MLP	Multilayer Perceptron
ADV	Adversarial Training

REFERENCES

[1] Dale W Jorgenson. "Accounting for growth in the information age". In: *Handbook of economic growth* 1 (2005), pp. 743–815.

[2] Charles Hulten and Leonard Nakamura. *Accounting for growth in the age of the internet: The importance of output-saving technical change*. Tech. rep. National Bureau of Economic Research, 2017.

[3] Diana Sousa, Andre Lamurias, and Francisco M Couto. "Using neural networks for relation extraction from biomedical literature". In: *Artificial Neural Networks*. Springer, 2021, pp. 289–305.

[4] Olivier Ferret et al. "How NLP can improve question answering". In: *KO KNOWLEDGE ORGANIZATION* 29.3-4 (2002), pp. 135–155.

[5] Chinatsu Aone, Mary Ellen Okurowski, and James Gorlinsky. "Trainable, scalable summarization using robust NLP and machine learning". In: *Proceedings of the 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics. 1998, pp. 62–66.

[6] Thorsten Brants. "Natural Language Processing in Information Retrieval." In: *Computational Linguistics in the Netherlands*. 2003.

[7] CA Deepa, PC ReghuRaj, and Ajeesh Ramanujan. "Relation Extraction using Deep Learning Methods-A Survey". In: *ICTACT Journal on Soft Computing* 9.3 (2019), pp. 1893–1902.

[8] Shantanu Kumar. "A survey of deep learning methods for relation extraction". In: *arXiv preprint arXiv:1705.03645* (2017).

[9] Matthew E Peters et al. “Deep contextualized word representations”. In: *arXiv preprint arXiv:1802.05365* (2018).

[10] Jacob Devlin et al. “Bert: Pre-training of deep bidirectional transformers for language understanding”. In: *arXiv preprint arXiv:1810.04805* (2018).

[11] Alec Radford et al. “Language models are unsupervised multitask learners”. In: *OpenAI blog* 1.8 (2019), p. 9.

[12] Jie Zhou et al. “Graph neural networks: A review of methods and applications”. In: *AI Open* 1 (2020), pp. 57–81.

[13] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).

[14] Liang Yao, Chengsheng Mao, and Yuan Luo. “Graph convolutional networks for text classification”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 7370–7377.

[15] John R Firth. “A synopsis of linguistic theory, 1930-1955”. In: *Studies in Linguistic Analysis* (1957).

[16] Yifan Peng, Chih-Hsuan Wei, and Zhiyong Lu. “Improving chemical disease relation extraction with rich features and weakly labeled data”. In: *Journal of Cheminformatics* 8.1 (2016), p. 53.

[17] Karen Spärck Jones. “IDF term weighting and IR research lessons”. In: *Journal of Documentation* (2004).

[18] Zhou Guodong et al. “Exploring various knowledge in relation extraction”. In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. 2005, pp. 427–434.

[19] Bruno Stecanella. *What is TF-IDF?* May 2019. URL: https://monkeylearn.com/blog/what-is-tf-idf/.

[20] Katrin Fundel, Robert Küffner, and Ralf Zimmer. "RelEx—Relation extraction using dependency parse trees". In: *Bioinformatics* 23.3 (2006), pp. 365–371.

[21] Alan R Aronson. "Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program." In: *Proceedings of the American Medical Informatics Association*. 2001, p. 17.

[22] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[23] Tomas Mikolov et al. "Distributed representations of words and phrases and their compositionality". In: *Advances in Neural Information Processing Systems*. 2013, pp. 3111–3119.

[24] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[25] Kawin Ethayarajh. "How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings". In: *arXiv preprint arXiv:1909.00512* (2019).

[26] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[27] Thien Huu Nguyen and Ralph Grishman. "Relation extraction: Perspective from convolutional neural networks". In: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*. 2015, pp. 39–48.

[28] Peng Wang et al. “Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification”. In: *Neurocomputing* 174 (2016), pp. 806–814.

[29] Jie Zhou et al. “Graph neural networks: A review of methods and applications”. In: *arXiv preprint arXiv:1812.08434* (2018).

[30] Richard Socher et al. “Recursive deep models for semantic compositionality over a sentiment treebank”. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013, pp. 1631–1642.

[31] Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. “Neural network acceptability judgments”. In: *Transactions of the Association for Computational Linguistics* 7 (2019), pp. 625–641.

[32] Aymé Arango, Jorge Pérez, and Barbara Poblete. “Hate speech detection is not as easy as you may think: A closer look at model validation”. In: *Proceedings of the 42nd international acm sigir conference on research and development in information retrieval*. 2019, pp. 45–54.

[33] Laura Chiticariu, Yunyao Li, and Frederick Reiss. “Rule-based information extraction is dead! long live rule-based information extraction systems!” In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 2013, pp. 827–832.

[34] Qi Li et al. “An end-to-end hybrid algorithm for automated medication discrepancy detection”. In: *BMC Medical Informatics and Decision Making* 15.1 (2015), p. 37.

[35] Jiayuan He et al. “An extended overview of the CLEF 2020 ChEMU Lab”. In: *the Conference and Labs of the Evaluation Forum (CLEF)*. 22-25 September 2020. 2020.

[36] Daniel Lowe and John Mayfield. “Extraction of reactions from patents using grammars”. In: *Central Europe Workshop Proceedings (CEUR-WS)*. 2020.

[37] Lezan Hawizy et al. “ChemicalTagger: A tool for semantic text-mining in chemistry”. In: *Journal of Cheminformatics* 3.1 (2011), p. 17.

[38] Cosmin Adrian Bejan, Wei-Qi Wei, and Joshua C Denny. “Assessing the role of a medication-indication resource in the treatment relation extraction from clinical text”. In: *Journal of the American Medical Informatics Association* 22.e1 (2015), e162–e176.

[39] Michael Kuhn et al. “A side effect resource to capture phenotypic effects of drugs”. In: *Molecular systems biology* 6.1 (2010), p. 343.

[40] Wahab Khan et al. “A survey on the state-of-the-art machine learning models in the context of NLP”. In: *Kuwait Journal of Science* 43.4 (2016).

[41] Jon D Patrick et al. “I2b2 challenges in clinical natural language processing 2010”. In: *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*. 2010.

[42] Cyril Grouin et al. “CARAMBA: concept, assertion, and relation annotation using machine-learning based approaches”. In: *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*. 2010.

[43] Dina Demner-Fushman et al. “NLM’s system description for the fourth i2b2/VA challenge”. In: *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*. 2010.

[44] Illés Solt, Ferenc P Szidarovszky, and Domonkos Tikk. “Concept, Assertion and Relation Extraction at the 2010 i2b2 Relation Extraction Challenge using parsing information and dictionaries”. In: *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data* (2010).

[45] Timothy Miller, Alon Geva, and Dmitriy Dligach. “Extracting adverse drug event information with minimal engineering”. In: *Proceedings of the 2nd Clinical Natural Language Processing Workshop*. 2019, pp. 22–27.

[46] Bryan Rink, Sanda Harabagiu, and Kirk Roberts. “Automatic extraction of relations between medical concepts in clinical texts”. In: *Journal of the American Medical Informatics Association* 18.5 (2011), pp. 594–600.

[47] Xiaodan Zhu et al. “Detecting concept relations in clinical text: Insights from a state-of-the-art model”. In: *Journal of Biomedical Informatics* 46.2 (2013), pp. 275–285.

[48] Berry de Bruijn et al. “NRC at i2b2: one challenge, three practical tasks, nine statistical systems, hundreds of clinical records, millions of useful features”. In: *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*. 2010.

[49] Olivier Bodenreider. “The unified medical language system (UMLS): integrating biomedical terminology”. In: *Nucleic acids research* 32.suppl_1 (2004), pp. D267–D270.

[50] Guerganna Savova et al. "UIMA-based clinical information extraction system". In: *Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP* 39 (2008).

[51] David McClosky. *Any domain parsing: automatic domain adaptation for natural language parsing*. 2010.

[52] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. "Generating typed dependency parses from phrase structure parses." In: *Lrec*. Vol. 6. 2006, pp. 449–454.

[53] Mihaela Porumb et al. "REMed: automatic relation extraction from medical documents". In: *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*. ACM. 2015, p. 19.

[54] Anne-Lyse Minard et al. "Hybrid methods for improving information access in clinical documents: concept, assertion, and relation identification". In: *Journal of the American Medical Informatics Association* 18.5 (2011), pp. 588–593.

[55] Xi Yang et al. "Identifying relations of medications with adverse drug events using recurrent convolutional neural networks and gradient boosting". In: *Journal of the American Medical Informatics Association* 27.1 (2020), pp. 65–72.

[56] Jerome H Friedman. "Stochastic gradient boosting". In: *Computational statistics & data analysis* 38.4 (2002), pp. 367–378.

[57] Peter Anick et al. "I2B2 2010 challenge: machine learning for information extraction from patient records". In: *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data. Boston, MA, USA: i2b2*. 2010.

[58] Marc Moreno Lopez and Jugal Kalita. “Deep Learning applied to NLP”. In: *arXiv preprint arXiv:1703.03091* (2017).

[59] Sunil Kumar Sahu et al. “Relation extraction from clinical texts using domain invariant convolutional neural network”. In: *arXiv preprint arXiv:1606.09370* (2016).

[60] Yuan Luo et al. “Segment convolutional neural networks (Seg-CNNs) for classifying relations in clinical notes”. In: *Journal of the American Medical Informatics Association* 25.1 (2017), pp. 93–98.

[61] Geeticka Chauhan, Matthew McDermott, and Peter Szolovits. “Reflex: Flexible framework for relation extraction in multiple domains”. In: *arXiv preprint arXiv:1906.08318* (2019).

[62] Bin He, Yi Guan, and Rui Dai. “Classifying medical relations in clinical text via convolutional neural networks”. In: *Artificial intelligence in medicine* 93 (2019), pp. 43–49.

[63] Xinbo Lv et al. “Clinical relation extraction with deep learning”. In: *International Journal of Hybrid Information Technology* 9.7 (2016), pp. 237–248.

[64] Alex Graves. “Generating sequences with recurrent neural networks”. In: *arXiv preprint arXiv:1308.0850* (2013).

[65] Yuan Luo. “Recurrent neural networks for classifying relations in clinical notes”. In: *Journal of biomedical informatics* 72 (2017), pp. 85–95.

[66] Fenia Christopoulou et al. “Adverse drug events and medication relation extraction in electronic health records with ensemble deep learning methods”. In: *Journal of the American Medical Informatics Association* 27.1 (2020), pp. 39–46.

[67] Dhanachandra Ningthoujam et al. “Relation extraction between the clinical entities based on the shortest dependency path based LSTM”. In: *arXiv preprint arXiv:1903.09941* (2019).

[68] Zhiheng Li et al. “Integrating shortest dependency path and sentence sequence into a deep learning framework for relation extraction in clinical text”. In: *BMC Medical Informatics and Decision Making* 19.1 (2019), p. 22.

[69] Li Tang et al. “Convolutional LSTM Network with Hierarchical Attention for Relation Classification in Clinical Texts”. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–8.

[70] Qiang Wei et al. “A study of deep learning approaches for medication and adverse drug event extraction from clinical text”. In: *Journal of the American Medical Informatics Association* 27.1 (2020), pp. 13–21.

[71] Jon Patrick and Min Li. “High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge”. In: *Journal of the American Medical Informatics Association* 17.5 (2010), pp. 524–527.

[72] Veera Raghavendra Chikka and Kamalakar Karlapalem. “A hybrid deep learning approach for medical relation extraction”. In: *arXiv preprint arXiv:1806.11189* (2018).

[73] Kenneth W. Church and Patrick Hanks. “Word association norms, mutual information, and Lexicography”. In: *Proceedings of the 27th. Annual Meeting of the Association for Computational Linguistics*. Vancouver, B.C.: Association for Computational Linguistics, 1989, pp. 76–83.

[74] Fei Li and Hong Yu. "An investigation of single-domain and multidomain medication and adverse drug event relation extraction from electronic health record notes using advanced deep learning models". In: *Journal of the American Medical Informatics Association* 26.7 (2019), pp. 646–654.

[75] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. "Dynamic routing between capsules". In: *arXiv preprint arXiv:1710.09829* (2017).

[76] Xilun Chen and Claire Cardie. "Multinomial adversarial networks for multi-domain text classification". In: *arXiv preprint arXiv:1802.05694* (2018).

[77] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. "Adversarial multi-task learning for text classification". In: *arXiv preprint arXiv:1704.05742* (2017).

[78] Xi Yang et al. "Clinical relation extraction using transformer-based models". In: *arXiv preprint arXiv:2107.08957* (2021).

[79] Ilseyar Alimova and Elena Tutubalina. "Multiple features for clinical relation extraction: A machine learning approach". In: *Journal of Biomedical Informatics* 103 (2020), p. 103382.

[80] Jinhyuk Lee et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4 (2020), pp. 1234–1240.

[81] Emily Alsentzer et al. "Publicly available clinical BERT embeddings". In: *arXiv preprint arXiv:1904.03323* (2019).

[82] Qingyu Chen, Yifan Peng, and Zhiyong Lu. "BioSentVec: creating sentence embeddings for biomedical texts". In: *2019 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE. 2019, pp. 1–5.

[83] Qiang Wei et al. "Relation Extraction from Clinical Narratives Using Pre-trained Language Models". In: *Proceedings of the American Medical Informatics Association*. Vol. 2019. 2019, p. 1236.

[84] Zhiheng Li et al. "Exploiting sequence labeling framework to extract document-level relations from biomedical texts". In: *BMC bioinformatics* 21.1 (2020), pp. 1–14.

[85] Yinhan Liu et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[86] Zhilin Yang et al. "Xlnet: Generalized autoregressive pretraining for language understanding". In: *Advances in neural information processing systems* 32 (2019).

[87] Fatema Hasan, Arpita Roy, and Shimei Pan. "Integrating Text Embedding with Traditional NLP Features for Clinical Relation Extraction". In: *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE. 2020, pp. 418–425.

[88] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[89] Meizhi Ju et al. "An ensemble of neural models for nested adverse drug events and medication extraction with subwords". In: *Journal of the American Medical Informatics Association* 27.1 (2020), pp. 22–30.

[90] Yoshimasa Tsuruoka et al. "Developing a robust part-of-speech tagger for biomedical text". In: *Panhellenic conference on informatics*. Springer. 2005, pp. 382–392.

[91] Arpita Roy and Shimei Pan. "Incorporating medical knowledge in BERT for clinical relation extraction". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. 2021, pp. 5357–5366.

[92] Jenny Copara et al. "Named entity recognition in chemical patents using ensemble of contextual language models". In: *arXiv preprint arXiv:2007.12569* (2020).

[93] Jingqi Wang1 Yuankai Ren2 Zhi Zhang and Yaoyun Zhang. "Melaxtech: A report for CLEF 2020–ChEMU Task of Chemical Reaction Extraction from Patent". In: (2020).

[94] Po-Ting Lai and Zhiyong Lu. "BERT-GT: cross-sentence n-ary relation extraction with BERT and Graph Transformer". In: *Bioinformatics* 36.24 (2020), pp. 5678–5685.

[95] Shiao Xu et al. "BERT Gated Multi-window Attention Network for Relation Extraction". In: *Neurocomputing* (2021).

[96] Rongli Yi and Wenxin Hu. "Pre-trained bert-gru model for relation extraction". In: *Proceedings of the 2019 8th International Conference on Computing and Pattern Recognition*. 2019, pp. 453–457.

[97] Kyunghyun Cho et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078* (2014).

[98] Iris Hendrickx et al. "Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals". In: *arXiv preprint arXiv:1911.10422* (2019).

[99] Nanyun Peng et al. "Cross-sentence n-ary relation extraction with graph lstms". In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 101–115.

[100] Jiao Li et al. "BioCreative V CDR task corpus: a resource for chemical disease relation extraction". In: *Database* 2016 (2016).

[101] Chih-Hsuan Wei et al. "Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task". In: *Database* 2016 (2016).

[102] Yifan Peng, Shankai Yan, and Zhiyong Lu. "Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets". In: *arXiv preprint arXiv:1906.05474* (2019).

[103] Hao Zhu et al. "Graph neural networks with generated parameters for relation extraction". In: *arXiv preprint arXiv:1902.00756* (2019).

[104] Li Zhou et al. "A weighted GCN with logical adjacency matrix for relation extraction". In: *ECAI 2020*. IOS Press, 2020, pp. 2314–2321.

[105] Jian Wang et al. "Document-level biomedical relation extraction using graph convolutional network and multihead attention: algorithm development and validation". In: *JMIR Medical Informatics* 8.7 (2020), e17638.

[106] Daojian Zeng, Chao Zhao, and Zhe Quan. "CID-GCN: an effective graph convolutional networks for chemical-induced disease relation extraction". In: *Frontiers in Genetics* 12 (2021), p. 115.

[107] Sunil Kumar Sahu et al. "Inter-sentence Relation Extraction with Document-level Graph Convolutional Neural Network". In: *arXiv preprint arXiv:1906.04684* (2019).

[108] Shuang Zeng et al. “Double graph based reasoning for document-level relation extraction”. In: *arXiv preprint arXiv:2009.13752* (2020).

[109] Kang Zhao et al. “Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction”. In: *Knowledge-Based Systems* 219 (2021), p. 106888.

[110] Yuhao Zhang, Peng Qi, and Christopher D Manning. “Graph convolution over pruned dependency trees improves relation extraction”. In: *arXiv preprint arXiv:1809.10185* (2018).

[111] Wenti Huang et al. “Local-to-global GCN with knowledge-aware representation for distantly supervised relation extraction”. In: *Knowledge-Based Systems* 234 (2021), p. 107565.

[112] Yuanhe Tian et al. “Dependency-driven Relation Extraction with Attentive Graph Convolutional Networks”. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021, pp. 4458–4471.

[113] Bowen Yu et al. “Learning to prune dependency trees with rethinking for neural relation extraction”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. 2020, pp. 3842–3852.

[114] Lianzhe Huang et al. “Text level graph neural network for text classification”. In: *arXiv preprint arXiv:1910.02356* (2019).

[115] Zenan Zhai et al. “Improving chemical named entity recognition in patents with contextualized word embeddings”. In: *arXiv preprint arXiv:1907.02679* (2019).

[116] Zhibin Lu, Pan Du, and Jian-Yun Nie. "VGCN-BERT: Augmenting BERT with Graph Embedding for Text Classification". In: *Advances in Information Retrieval - 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14-17, 2020, Proceedings, Part I*. Vol. 12035. Lecture Notes in Computer Science. Springer, 2020, pp. 369–382.

[117] Sebastian Riedel, Limin Yao, and Andrew McCallum. "Modeling relations and their mentions without labeled text". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2010, pp. 148–163.

[118] Sharmistha Jat, Siddhesh Khandelwal, and Partha Talukdar. "Improving distantly supervised relation extraction using word and entity based attention". In: *arXiv preprint arXiv:1804.06987* (2018).

[119] Daniil Sorokin and Iryna Gurevych. "Context-aware representations for knowledge base relation extraction". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 1784–1789.

[120] Yuan Yao et al. "DocRED: A large-scale document-level relation extraction dataset". In: *arXiv preprint arXiv:1906.06127* (2019).

[121] Yifu Li, Ran Jin, and Yuan Luo. "Classifying relations in clinical narratives using segment graph convolutional and recurrent neural networks (Seg-GCRNs)". In: *Journal of the American Medical Informatics Association* 26.3 (2019), pp. 262–268.

[122] Darshini Mahendran, Chathurika Brahmana, and Bridget McInnes. "SciREL at SemEval-2018 Task 7: A System for Semantic Relation Extraction and Classification". In: *Proceedings of The 12th International Workshop on Semantic Evaluation*. 2018, pp. 853–857.

[123] Darshini Mahendran et al. "NLPatVCU CLEF 2020 ChEMU Shared Task System Description". In: *Conference and Labs of the Evaluation Forum (CLEF) 2020 Working Notes*. 2020.

[124] Darshini Mahendran, Cora Lewis, and Bridget McInnes. "NLP@ VCU: Identifying adverse effects in English tweets for unbalanced data". In: *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*. 2020, pp. 158–160.

[125] Darshini Mahendran and Bridget T McInnes. "Extracting Adverse Drug Events from Clinical Notes". In: *arXiv preprint arXiv:2104.10791* (2021).

[126] Darshini Mahendran et al. "Identifying Chemical Reactions and Their Associated Attributes in Patents". In: *Frontiers in Research Metrics and Analytics* 6 (2021).

[127] Darshini Mahendran et al. "BioCreative VII-Track 1: A BERT-based System for Relation Extraction in Biomedical Text". In: *Proceedings of the Biocreative VII*. 2021.

[128] Darshini Mahendran, Cristina Tang, and Bridget McInnes. "Graph Convolutional Networks for Chemical Relation Extraction". In: *Proceedings of the Semantics-enabled Biomedical Literature Analytics (SeBiLAn)*. 2022.

[129] Özlem Uzuner et al. "2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text". In: *Journal of the American Medical Informatics Association* 18.5 (2011), pp. 552–556.

[130] Sam Henry et al. "2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records". In: *Journal of the American Medical Informatics Association* 27.1 (2020), pp. 3–12.

[131] Alistair EW Johnson et al. "MIMIC-III, a freely accessible critical care database". In: *Scientific data* 3 (2016), p. 160035.

[132] Dat Quoc Nguyen et al. "ChEMU: Named Entity Recognition and Event Extraction of Chemical Reactions from Patents". In: *European Conference on Information Retrieval*. Springer. 2020, pp. 572–579.

[133] Antonio Miranda et al. "Overview of DrugProt BioCreative VII track: quality evaluation and large scale text mining of drug-gene/protein relations". In: *Proceedings of the seventh BioCreative challenge evaluation workshop*. 2021.

[134] Darshini Mahendran and Bridget McInnes. "Extracting Adverse Drug Events from Clinical Notes". In: *Submission to the American Medical Informatics Association* (2020).

[135] Darshini Mahendran and Bridget McInnes. "Hierarchical multi-label classification for clinical relation extraction". In: *Preparation for Journal of BioInformatics* (2022).

[136] Ronan Collobert and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning". In: *Proceedings of the 25th International Conference on Machine learning*. ACM. 2008, pp. 160–167.

[137] François Chollet et al. *Keras*. `https://github.com/fchollet/keras`. 2015.

[138] Jiayuan He et al. "Chemu 2020: Natural language processing methods are effective for information extraction from chemical patents". In: *Frontiers in Research Metrics and Analytics* 6 (2021), p. 12.

[139] Sampo Pyysalo et al. “Distributional Semantics Resources for Biomedical Text Processing”. In: *The 5th International Symposium on Languages in Biology and Medicine*. 2013.

[140] P.W.D. Charles. *Project Title*. `https://github.com/charlespwd/project-title`. 2013.

[141] Hilal Dönmez et al. “BOUN-REX at CLEF-2020 ChEMU Task 2: Evaluating Pretrained Transformers for Event Extraction”. In: (2020).

[142] Suncong Zheng et al. “Joint entity and relation extraction based on a hybrid neural network”. In: *Neurocomputing* 257 (2017), pp. 59–66.

[143] Hao Fei et al. “A span-graph neural model for overlapping entity relation extraction in biomedical texts”. In: *Bioinformatics* 37.11 (2021), pp. 1581–1589.

[144] Qiuyan Xu and Fang Li. “Joint learning of named entity recognition and relation extraction”. In: *Proceedings of 2011 International Conference on Computer Science and Network Technology*. Vol. 3. IEEE. 2011, pp. 1978–1982.

[145] Boris Hanin. “Which neural net architectures give rise to exploding and vanishing gradients?” In: *Advances in neural information processing systems* 31 (2018).