Integrating Text Embedding with Traditional NLP Features for Clinical Relation Extraction

Fatema Hasan[§], Arpita Roy[§], Shimei Pan
Department of Information Systems
University of Maryland, Baltimore County
Baltimore, USA
{fhasan1,arpita2,shimei}@umbc.edu

***Abstract*—Recently, text embedding techniques such as Word2Vec and BERT have produced state-of-the-art results in a wide variety of NLP tasks. As a result, traditional NLP features frequently used in Information Extraction (IE) such as POS tags, dependency relations and semantic types have received less attention. In this paper, we investigate whether traditional NLP features can be combined with word and sentence embeddings to improve relation extraction. We have explored diverse feature sets and different neural network architectures and evaluated our models on a benchmark clinical text dataset. Our new models significantly outperformed all the baselines on the same dataset.**

***Index Terms*—Relation Extraction, IE, Clinical Text, BERT, Word2Vec, Neural Networks, MIMIC-III, i2b2.**

I. Introduction

Electronic Medical Record (EMR) is an electronic version of a patient's medical history that is maintained by healthcare providers over time. Clinical text in EMR includes physicians' notes, surgical records, discharge summaries and laboratory reports. Clinical text contains valuable information about a patient's conditions such as symptoms, diagnoses and treatments. Hence identifying, extracting and mining this information is of great importance to managing and improving patient care. Manually inspecting large amount clinical text is labor-intensive and time-consuming. Therefore, natural language processing (NLP), which can automatically extract information of interest from text is beneficial for clinical research and applications. One of the fundamental tasks of clinical NLP is extracting relations between medical concepts from texts (e.g., extracting relations between diseases and treatments). For example, in the sentence "Ibuprofen reduced inflammation but likely caused heartburn", "inflammation" was effectively treated by "ibuprofen" and "heartburn" is an adverse event caused by the same drug. Identifying these relations is important to understand how patients respond to treatments. Therefore, automated extraction of relations between medical concepts is essential for clinical decision making and patient care.

Researchers have proposed various methods including rule-based methods or supervised/unsupervised machine learning methods (e.g., deep learning methods) to build effective relation extraction systems. In this study, we focus on building neural network models to extract relations from clinical text. More specifically, we develop neural network models that classify the relationship between two medical concepts in a given sentence (e.g., between a treatment such as "ibuprofen" and a medical problem such as "inflammation") . We explore multiple state-of-the-art neural network models such as Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), Residual Neural Network (ResNet) and Graph Convolutional Network (GCN) [1]–[3] to identify the best neural network architecture for the given task.

Moreover, word/sentence embedding, a technique to automatically learn dense vector representations for words and sentences has proven to be effective for relation extraction [4], [5]. Recently dynamic/contextual word embedding (e.g., BERT), where the embedding vector for each word varies with its context has gained much popularity over static word embedding (e.g., Word2Vec) where a fixed embedding vector is learned for each word. While both contextual and static embedding have produced impressive results on diverse downstream NLP tasks, it is unclear whether they can be combined effectively with typical syntactic and semantic features commonly used in traditional relation extracting systems. We specifically compare the effectiveness of BERT contextual embedding with Word2Vec static embedding, especially when they are combined with traditional syntactic and semantic features in relation extraction.

We evaluate our models on a benchmark clinical text dataset, the i2b2-2010 dataset [6] where we classify relations between medical concepts.

The main contributions of our work can be summarized as follows:

- We systematically investigate the effectiveness of different neural network architectures in combining text embedding (e.g., Word2Vec and BERT) with traditional syntactic and semantic features for clinical relation extraction. Our best model achieved a F-score of 0.88, which outperformed the state-of-the art baselines on the same dataset by a large margin.
- We compare the models using static word embedding (e.g., Word2Vec) with models using contextual embedding (e.g., BERT). Our results indicate that although BERT contextual embedding on its own is very effective, models combining BERT with traditional syntactic and semantic

[§]Equal contribution

features performed much worse than models combining static Word2Vec embedding with traditional NLP features.

II. Related Work

[7] is the first to explore relation extraction between medical concepts from clinical text. [7] defined six relation types between diseases, tests and treatments. and trained six different classifiers, one for each relation type. The system employed diverse feature sets included surface features (e.g., distance), lexical features (e.g., lexical trigrams), and shallow syntactic features (e.g., syntactic tree path). Based on this study, the i2b2/VA Challenge [6] was launched in 2010. Similar to [7], most participants of the challenge employed feature engineering and supervised machine learning. SVM was the most commonly used classifier [8]–[15]. In addition, [16] used a hybrid approach where SVM was used to predict classes with a large number of instances and a rule-based approach was used to predict classes with fewer samples. [17], [18] applied bootstrapping on unlabeled data for training semi-supervised models. Additionally, [17] addressed the imbalanced dataset issue by downsampling the training set. A wide range of hand crafted features were used in these systems including context features, lexical features, syntactic features, as well as semantic features extracted from external knowledge resources such as UMLS, cTAKES, and Medline. Among them, [17], [19] derived concept mapping and concept types based on UMLS. [15], [17] employed Medline to calculate Pointwise Mutual Information (PMI) between two concepts. GENIA[1] was the most commonly used tool to generate Part of speech (POS), phrase chunking and dependency tree features.

To avoid labor-intensive feature engineering, recently, researchers have shifted their attention towards building relation extraction systems using deep learning frameworks. Several neural network based models have been proposed to extract relations between medical concepts. Among them, [1] used convolution neural network (CNN) to learn features automatically with superior classifier performance. [20] proposed a Segment Convolutional Neural Network (Seg-CNN) where relations between two concepts are identified by simultaneously learning separate representations for each text segment. [2], [21] presented relation extraction systems based on the shortest dependency path (SDP) generated from the dependency tree of a sentence. Among them, [2] used LSTM for sentence representation and CNN/LSTM to represent SDP. In contrast, the system proposed by [21] takes only the words in SDP as the input to LSTM. Commonly used features in these systems are word representations, word types, POS tags, IOB encoding of semantic concepts, relative distance, Chunk tags, and dependency relations. While most of these works used static word embedding such as Word2Vec, [22] evaluated the performance of relation extraction model using multiple contextual embedding models such as bioelmo [23], BERT [24], BioBERT [25], BlueBERT [22]. They however did not explore how contextual embedding features can be combined with traditional NLP features to improve performance. [26] is the first to combine BERT embeddings with traditional IOB tags. It however did not explore whether BERT embeddings can be effectively combined with other syntactic and semantic features commonly used in relation extraction. Unlike [26], which takes the BERT token embedding and uses it as the input to LSTM, we used the sentence embedding from BERT and combine it with the sequential representation of syntactic, semantic and surface features learned by LSTM.

Despite a substantial body of research on relation extraction, it is still an open question in terms of which neural network architecture and feature sets may result in the best system. To the best of our knowledge, there is no existing study that systematically investigates the effectiveness of combining a comprehensive set of traditional NLP features with BERT contextual embedding in neural network-based relation extraction.

III. System Description

In this section, we describe the diverse feature sets as well as multiple neural network architectures we have explored to combine traditional NLP features with text embeddings for clinical relation extraction.

A. Input Features

Word embedding: Given an input sentence $\mathcal{S} = \langle w_1, w_2, w_3, ..., ..., w_n \rangle$, the word embedding of each word $w_i \in \mathcal{S}$ is defined as $E_w(w_i)$. To derive $E_w(w_i)$, we train a Word2Vec model on the Medical Information Mart for Intensive Care (MIMIC)-III clinical corpus [27] and the i2b2 dataset to ensure we generate good quality word features for our target application domain.

To compare the effectiveness of static word embedding techniques (e.g., Word2Vec) with the state-of-the-art contextual word embedding techniques, we also included the embeddings generated by BERT [24]. Here we used the embeddings generated by ClinicalBert [28], a pre-trained BERT model in the clinical domain. ClinicalBert is trained on the texts from BookCorpus, English Wikipedia, biomedical articles from PubMed and EMRs from the (MIMIC)-III dataset [27].

POS embedding: We derive the part of speech tag of each word, $\texttt{POS}(w_i)$ using Stanford CoreNLP [29]. Then a POS embedding for each word w_i as $E_{pos}(w_i)$ is learnt using a Keras Embedding layer.

IOB encoding: To encode whether each word in $\mathcal{S}$ belongs to any of the target concepts, we assign the IOB encoding $\texttt{IOB}(w_i)$ to each word. There are a total of seven IOB tags in our system: $\langle \texttt{I}_{\texttt{treatment}}, \texttt{I}_{\texttt{test}}, \texttt{I}_{\texttt{problem}}, \texttt{B}_{\texttt{treatment}}, \texttt{B}_{\texttt{test}}, \texttt{B}_{\texttt{problem}}, \texttt{and}\texttt{O} \rangle$, where the prefix *B* represents the beginning of a concept (E.g., $\texttt{B}_{\texttt{treatment}}$ represents the first word of a `treatment` concept), and *I* indicates a word inside a concept (e.g., $\texttt{I}_{\texttt{treatment}}$ represents an internal word of a `treatment`). The tag *O* represents other words not related to any of the target concepts. Thereafter, each IOB tag is transformed into an IOB embedding $E_{iob}(w_i)$ using a Keras embedding layer in

[1] http://www.nactem.ac.uk/GENIA/tagger/

our neural network architecture. Table I shows an example of the IOB encoding of the words in a sentence.

Relative Distance: We employ a relative distance encoding for each of the two concepts c_1, c_2 in a relation. We do this by marking the positions of all the words in a target concept as 0. Every word to its right is assigned an incrementally higher distance number and every word to its left is assigned an incrementally lower number. We can calculate the embedding of the relative positions as $E_{r1}(w_i)$ and $E_{r2}(w_i)$ with a Keras embedding layer. Table I shows an example of the relative distance encoding for the two concepts c_1, c_2 in a relation.

Concept Embedding: Concept embedding is generated by taking the average of the embeddings of the words presented in a concept. For example, the embedding of c_2 is generated by averaging the word embeddings of "left", "", "carotid", "ophthalmic" and "aneurysm".

Dependency Tree: Dependency tree is represented as a directed graph, with m nodes corresponding to each of the m words in a sentence. Figure 1 shows the dependency tree for an example sentence. We extract the dependency tree using Stanford CoreNLP [29]. To create a fix sized data structure irrespective of the length of the sentence and the depth of the dependency tree, we converted the tree to an $n \times n$ adjacency matrix, $\mathcal{A}$, where n is a pre-determined fixed sentence length, If there is a edge between w_i and w_j in the dependency tree, then $\mathcal{A}_{ij} = \mathcal{A}_{ji} = 1$ and 0 otherwise. Following [30], we add self-loop and normalization into the adjacency matrix as these operations have shown to improve effectiveness. We add self loop by $\tilde{\mathcal{A}} = (\mathcal{A} + I)$ where I is an $n \times n$ identity matrix. We then perform normalization for each row of $\tilde{\mathcal{A}}$ so that $\tilde{\mathcal{A}}_i = \tilde{\mathcal{A}}_i / d_i$ where $d_i = \sum_{j=1}^{n} A_{ij}$ is the degree of the word at the ith position in the graph.

B. Model Architecture:

Here we explore various commonly used neural network architectures to combine diverse text embeddings with traditional syntactic, semantic and surface features: (i) Convolutional Neural Networks (CNN) [31] (ii) Graph Convolutional Networks (GCN) [32] (iii) Residual Networks (ResNet) [33], and (iv) Bidirectional Long Short Term Memory (BiLSTM) [34] networks. Since some of the neural network architectures such as CNN and BiLSTM are well-known in the NLP community, here we focus on explaining architectures that are less well-known (e.g.,GCN).

Figure 2a shows the architecture of the basic BiLSTM, CNN, ResNet, 2b shows the architecture of GCN model and 2c shows the architecture of an extended GCN model with additional Concept Dependency Tree features. In general, words in sentences, words in concepts, relative distances, POS tags and IOB tags are first input into an embedding layer to learn their embeddings. For CNN, ResNet and BiLSTM, the embedding features are then concatenated and input to these neural networks: $E(w_i) = E_w(w_i) \oplus E_{pos}(w_i) \oplus E_{iob}(w_i) \oplus E_{r1}(w_i) \oplus E_{r2}(w_i) \oplus \tilde{\mathcal{A}}_i$, where $\oplus$ operator denotes a concatenation operation. Therefore, $E_{CNN/ResNet/BiLSTM} = \langle E(w_1), E(w_2), E(w_3), ..., ..., E(w_n) \rangle$ forms our input features for all words in a sentence. Typical neural network architectures we explored include:

BiLSTM: Our system uses Bidirectional LSTM to aggregate word-level features and generate sentence-level representations [34]. BiLSTM is able to learn the word order as well as the long term dependency in a sentence. The advantage of using BiLSTM is that we can leverage the information from neighboring words at both sides. Input features to BiLSTM model is as discussed above and the output of BiLSTM is combined with concept embedding for relation classification.

CNN: CNN [31] is capable of learning local features such as short phrases or recurring n-grams. Our system uses CNN in a similar way to BiLSTM with the same input and output combination.

ResNet: ResNet [33] provides the benefit of CNN while reducing vanishing gradient problem in deep networks. The input and output of ResNet remains the same as those for CNN.

GCN: GCN [32] is an adaption of CNN for encoding graphs. It operates directly on graph structures (e.g., an adjacency matrix encoding a dependency tree). For a starting node i and the ending node j in a graph, in an L-layer of GCN, if we denote the input vector by $h_i^{(l-1)}$ and the output vector for node i at the l-th layer by $h_i^{(l)}$, the graph convolution operation can be defined as:

$$h_i^{(l)} = \sigma\ (\sum_{j=1}^{n} \mathcal{A}_{ij} W^{(l)} h_j^{(l-1)} + b^l) \tag{1}$$

where, $\mathcal{A}_{ij}$ is an adjacency matrix, $W^{(l)}$ is a weight matrix, b^l a bias term and σ a nonlinear function (e.g., RELU). Intuitively, during each graph convolution, each node gathers and summarizes information from its neighboring nodes in the graph. Therefore, the output of GCN is a node representation, in our case word representation. For GCN, the concatenation of embedding features are, $E(w_i) = E_w(w_i) \oplus E_{pos}(w_i) \oplus E_{iob}(w_i) \oplus E_{r1}(w_i) \oplus E_{r2}(w_i)$ and therefore for all the words in a sentence the input features are $E_{GCN} = \langle E(w_1), E(w_2), E(w_3), ..., ..., E(w_n) \rangle$ and Adjajency matrix, $\tilde{\mathcal{A}}$. Note that we do not concatenate $\tilde{\mathcal{A}}_i$ in the embedding features. The output of GCN is combined with the concept embeddings of c_1 and c_2 for relation classification.

We also implement a variation of GCN model called GCN-CDT (Figure 2b) to take full advantage of its ability to capture information from its neighboring nodes in a graph. In addition to the GCN for the entire sentence, in GCN-CDT we also feed two concept embeddings as well as their concept dependency trees (CDTs) to two additional GCNs. We extract a concept dependency tree from each of the two concepts by following the same method described in (section III-A) for a sentence dependency tree.

Sentence, $\mathcal{S}$	She is status post [coiling]$_{treatment}$ with stent placement for [left carotid ophthalmic aneurysm]$_{problem}$.
c_1	coiling
c_2	left carotid ophthalmic aneurysm
Relative Distance (c_1)	$\langle -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 \rangle$
Relative Distance (c_2)	$\langle -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 0, 0, 0, 1 \rangle$
IOB Encoding	$\langle$ O, O, O, O, B$_{\texttt{treatment}}$, O, O, O, O, B$_{\texttt{problem}}$, I$_{\texttt{problem}}$, I$_{\texttt{problem}}$, I$_{\texttt{problem}}$, O $\rangle$

TABLE I: A Example of Relative Distance and IOB Encoding

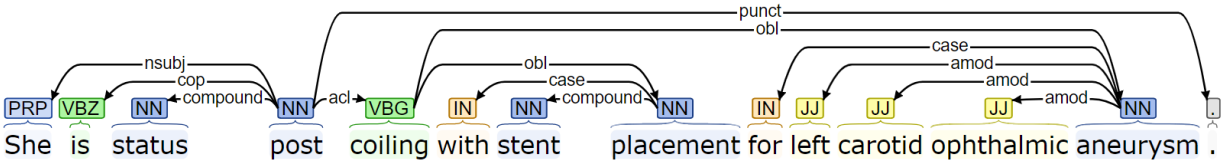
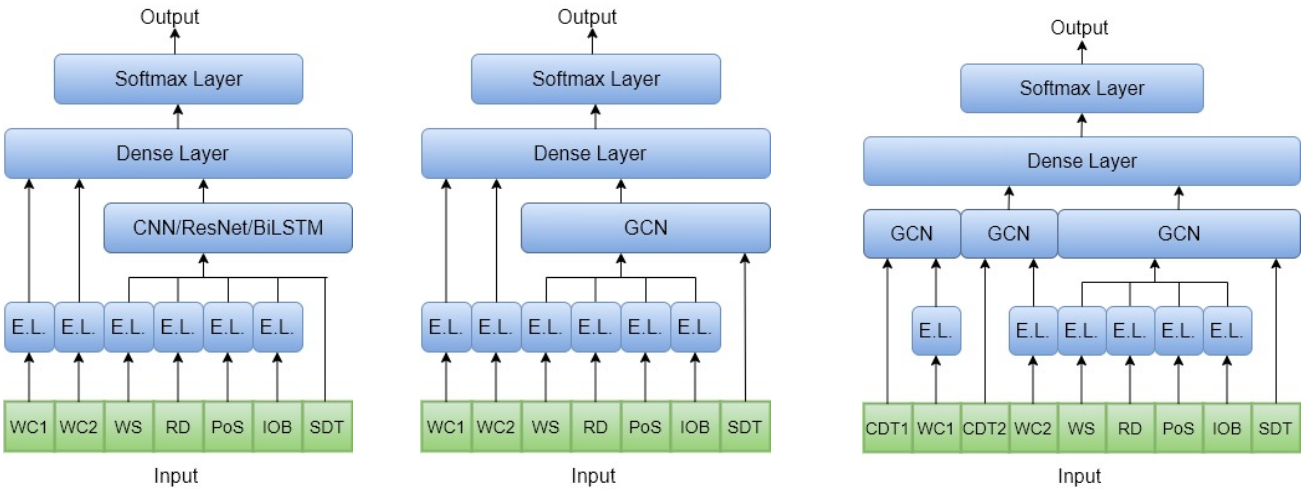


Fig. 1: An Example of A Sentence Dependency Tree and POS Tags



(a) Model with CNN / RestNet / BiLSTM (b) Model with GCN (c) Model with GCN-CDT

Fig. 2: Model Architecture (E.L.=Embedding Layer, WS=words in sentence, WC1=words in concept1, WC2=words in concept2, SDT=sentence dependency tree, CDT1=dependency tree of concept1, CDT2=dependency tree of concept2, RD=relative distance, IOB=IOB tag)

IV. Experiments

A. Dataset

For this study, we use the dataset from the 2010 i2b2/VA on Natural Language Processing Challenges for Clinical Records. This dataset contains discharge summary and progress report from different healthcare providers. Our research focuses on the relation extraction task which is to identify eight target relations among 3 medical concepts such as treatments, problems and tests. The dataset used for relation extraction during the 2010 i2b2/VA challenge includes a total of 394 training reports, 477 test reports, and 877 un-annotated reports. After the challenge however, only a part of the data was publicly released. The dataset we downloaded from the i2b2 website only includes 170 documents for training and 256 documents for testing. Descriptions and statistics of the target relations can be found in table II.

B. Baseline Systems

To select baselines, we consider systems that employ the same number of training instances and define the classification task with the the same granularity level as ours. So far, we have found only two existing systems [1], [2] meeting these criteria.

Baseline 1: [2] takes the entire sentence along with the relative positions as the input and generates a sequential representation of the input features using a BiLSTM. In addition, it utilizes both CNN and BiLSTM to capture the syntactic context of the two target entities using the shortest dependency path between them. Finally the concatenation of the output from these two modules are used by a fully connected layer for classification. For word embedding, they used pre-trained `Word2Vec` on the **MIMIC-III** corpus where all the concepts were replaced with their concept types.

Baseline 2: [1] employs a convolutional neural network

Relation Type	Train Report	Test Report	Total
Treatment improve or cure medical problem (**TrIP**)	51	152	203
Treatment worsen medical problem (**TrWP**)	24	109	133
Treatment caused medical problems (**TrCP**)	184	342	526
Treatment administered medical problem (**TrAP**)	885	1732	2617
Treatment was not administered because of medical problem (**TrNAP**)	62	112	174
Test reveal medical problem (**TeRP**)	993	2060	3053
Test conducted to investigate medical problem (**TeCP**)	166	338	504
Medical problem indicates medical problems (**PIP**)	755	1448	2203
No Relation (**None**)	7111	12821	19932

TABLE II: Statistics of the relation extraction dataset from the 2010 i2b2/VA challenge

with 6 discrete word features as the input: (i) a word itself, (ii) distance from the first concept, (iii) distance from the second concept, (iv) POS tags, (v) chunk tag of the word and (vi) concept types. For word embedding, they used pre-trained word vectors trained on Pubmed articles using Word2Vec. The authors did not include three infrequent classes (TrWP, TrIP, TrNAP) along with their instances. Therefore, for the sake of fair comparison, we also calculate the F1 score with the support of 6 class.

C. Experiment Setting

In this section we provide the implementation details of each of the models. To implement the **CNN** model, we use 7 one-dimensional convolution layers with kernel size 128 and filter size 3; each layer is followed by a Batch Normalization [35] layer and a ReLU activation function. The **ResNet** model shares the same kernels, filter size, normalisation and activation function as those of the CNN model. The difference being the first convolution layer is followed by 3 ResNet blocks where each block consists of two convolution layers with a shortcut connection going across each block. The **BiLSTM** model is implemented with 2 stacked bidirectional LSTM layers with 128 cells in each layer, each using the $\texttt{tanh}(\cdot)$ activation function. The **GCN** model is constructed with 2 graph convolution layers, each containing 128 units. For all the models, we use two fully connected layers before the softmax layer with 128 and 64 neurons respectively. We use the Adam optimizer [36] with a 0.001 learning rate for all the models. We use a batch size of 256, with 10% word-level dropout and 10% recurrent dropout in training BiLSTM. For CNN, ResNet and GCN, we use a batch size of 32 with 10% dropout rate. To avoid over-fitting, we employ 10% dropout in the fully connected layer. The word embeddings and concept embeddings are initialized using 100-dimensional Word2Vec pretrained on Mimic-III and i2b2 datasets. The embedding layers for POS, IOB encoding, Relative Positions are 20, 5, 50 respectively and randomly initialized. For relative position embedding, we use a specialized Position Embedding layer[2] that maps integers (negative, positive, zero) to a embedding space. We implement all our models using Keras[3] with a Tensorflow[4] backend. As this is a relatively small corpus, we combine the training and testing data and perform a 5-fold cross-validation in our experiments.

D. Performance Evaluation

In this section, we evaluate (i) the effectiveness of different neural network architectures, (ii) the effectiveness of different feature sets, and (iii) the effectiveness of static versus contextual text embedding.

*1) **Effectiveness of Different Architectures**:* To investigate the effectiveness of different neural network architectures, we include all the features (e.g., words, relative distances, POS tags, IOB tags, concept embedding and dependency tree) in all the models (e.g., BiLSTM, CNN, GCN, and ResNet). We calculate per class (9 class) and weighted F1 score and compared the results with the two baselines. As shown in Table III, the CNN model outperforms both baselines [1], [2] by 7% and 12% based on the weighted F1 score.Since the basic CNN architecture achieved better performance than the two baselines, we also investigated whether incorporating residual connections into CNN can further enhance the result as they have been shown to improve the learning capacity of convolutional models [37]. We noticed that not only ResNet achieved much higher F1 score over the two baselines (12% and 16% respectively), the per class F1 scores are also better or close to the baselines. We next apply GCN as it might be able to better utilize the dependency tree information through its built-in learnable node representation scheme. However, GCN provides only 6% and 9% improvement over the two baselines. We hypothesised that the dependency tree of the full sentence may be too noisy as only a portion of the dependency tree is directly related to the relations between two target concepts. To verify our hypothesis, we add the concept-specific dependency relations to the model (in addition to the dependency tree of the whole sentence). This effort is proven to be fruitful as the GCN-CDT model not only improves the weighted F1 score by 9% and 14% over the two baselines, the class specific F1 scores for PIP, TeRP have improved significantly as well.

Finally, we employed the BiLSTM model to learn a sequential representation of the input features to capture the long term dependency between them. As we can see, the BiLSTM

[2]https://github.com/CyberZHG/keras-pos-embd

[3]https://keras.io/

[4]https://www.tensorflow.org/

Model Name	No Relation	PIP	TeCP	TeRP	TrAP	TrCP	TrIP	TrNAP	TrWP	9 Class F1 Score	6 Class F1 Score
Baseline1 [2]	N/A	0.6333	0.6117	0.8444	0.7974	0.6213	0.6159	0.4227	0.4457	0.7434	N/A
Baseline2 [1]	N/A	0.6492	0.5056	0.8125	0.6923	0.5644	N/A	N/A	N/A	N/A	0.7116
CNN	**0.9010**	**0.7514**	0.1903	0.7572	0.6493	0.2870	0	0.0110	0	**0.8136**	**0.8280**
GCN	**0.8964**	**0.6597**	0.2095	0.8113	0.6619	0.2199	0	0	0	**0.8094**	**0.8094**
GCN-CDT	**0.9032**	**0.7129**	0.4692	0.8144	0.7096	0.5111	0.3686	0.2867	0.1011	**0.8369**	**0.8469**
ResNet	**0.9165**	**0.7504**	0.5977	0.8378	0.7728	0.5978	0.4167	0.3564	0.3302	**0.8624**	**0.8710**
BiLSTM	**0.9275**	**0.7896**	**0.6437**	**0.8685**	**0.8057**	**0.6320**	0.5000	0.4025	0.2262	**0.8808**	**0.8894**
Support	19932	2203	504	3053	2617	526	203	174	133	29345	28835

TABLE III: Overall System Performance

model is the best model outperforming all the other model architectures and achieving 14% and 18% improvements over the two baselines.

In the following, we investigate the effectiveness of different feature sets using BiLSTM, the best model we identified earlier.

*2) **Feature Effectiveness**:* To investigate the contribution of each feature type on the relation extraction task, we conducted additional studies.

First we train a BiLSTM model, our best performing model, with each of the individual feature set separately. Table IV shows the results. Among the features, word embedding from Word2Vec seems to perform the best (F1=0.6974), followed by Relative Distance (F1=0.6449), IOB Encoding (F1=0.6173), POS Tags (F1=0.5986), Concept embedding (F1=0.5528) and Dependency Tree (F1=0.5495).

Since it is difficult to test all the possible feature combinations, we add each feature one by one based on their ranks shown in Table IV until all the features are added. As shown in Table V, adding *Relative Distances* after *word embedding* produces the most significant score boost (15%) followed by IOB encoding (3%). Adding POS Tags and Concept embeddings however does not seem to help. Finally, adding dependency relations improves the final F1 score by 1%.

Based on this experiment, word embedding, relative position, IOB encoding and dependency relation all seem to help relation extraction. In contrast, the information encoded in POS tags and Concept embeddings do not seem to be very useful.

Feature	F1 Score	Rank
Word2Vec	0.6974	1
relative_distance	0.6449	2
IOB_encoding	0.6173	3
POS tag	0.5986	4
concept	0.5528	5
dependency_tree	0.5495	6

TABLE IV: Feature Ranking in BiLSTM Model

*3) **Effectiveness of Different Text Embeddings**:* In this section, we compare the effectiveness of different text embedding techniques especially between a traditional static embedding technique such as Word2Vec with a state-of-the-art contextual embedding technique such as BERT. We also test whether they can be combined effectively with other features to improve the performance of relation extraction.

To combine BERT contextual embedding with other features, we have two options: (a) early fusion: we can combine BERT embedding with other features at the word/token level and then use BiLSTM to learn a sequential representation of all the word features. With this strategy, feature fusion occurs at the word level. (b) late fusion: we can combine the sentence representation learned by BERT with the sequential representation of all the other features learned by BiLSTM. With this strategy, feature fusion occurs at the sentence level.

Added Features	F1 Score	Impact
Word2Vec	0.6974	N/A
+ relative_distance	0.8429	0.1455
+ IOB Encoding	0.8766	0.0337
+ POS tag	0.8759	-0.0007
+ concept	0.8709	-0.005
+ dependency_tree	0.8808	0.0099

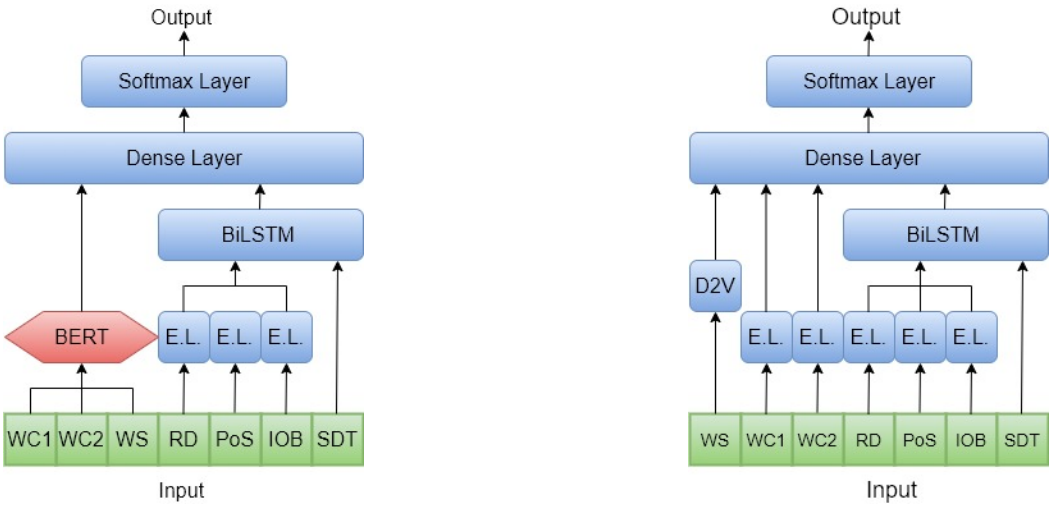
TABLE V: Impact of Word2Vec Feature Addition in BiLSTM Model

Added Features	F1-score	Impact
BERT	0.7447	N/A
+ relative_distance	0.7706	0.0259
+ IOB Encoding	0.7668	-0.0038
+ POS tag	0.7694	0.0026
+ concept	0.7647	-0.0047
+ dependency_tree	0.7646	- 0.0001

TABLE VI: Impact of Feature Addition to BERTcombinedLSTM

Added Features	F1 Measure	Impact
Doc2Vec	0.5622	N/A
+ relative_distance	0.6548	0.0926
+ IOB	0.7730	0.1182
+ POS tag	0.8198	0.0468
+ concept	0.8066	-0.0132
+ dependency_tree	0.8249	0.0183

TABLE VII: Impact of Doc2Vec Feature Addition to BiLSTM Model



(a) BERTcombinedLSTM

(b) Doc2Vec with LSTM

Fig. 3: Model Architecture with Late Fusion (E.L.=Embedding Layer, WS=words in sentence, WC1=words in concept1, WC2=words in concept2, SDT=sentence dependency tree, RD=relative distance, IOB=IOB tag)

Previous research suggested that using BERT sentence embedding for classification is more effective than using BERT word/token embedding and then input them to a LSTM to generate a sentence embedding [26]. As BERT already captures long term dependency between words in a sentence, it could be redundant to use LSTM to capture contextual information again. For this reason in our model shown in Figure 3a, we adopted a late fusion strategy where we directly use the sentence embedding learnt by BERT and combine it with a sequential representation of other word-level features (POS tag, relative distances, IOB encoding and dependency tree information) learnt by LSTM. In addition, we learn concept embeddings by averaging the embeddings of the tokens in each concept. Finally all the representations are merged together and passed through a fully connected layer for classification. In this model all the existing BERT parameters as well as the new parameters in BiLSTM are fine-tuned during relation classification.

As shown in table VI, BERT sentence embedding on its own is the most effective feature for relation extraction (F1=0.7447 versus Word2Vec F1=0.6974). This result highlights the power of BERT in capturing the semantics of a sentence. Adding other features to BERT embedding however provides only 2% improvement of the F1 score. In contrast, adding the same features to static Word2Vec embedding has resulted in a 18% increase in performance. The performance of the full model with the BERT sentence embedding (F1=0.7646) is 11.6% lower than that with Word2Vec embedding (F1=0.8808). This result is quite surprising as it implies that there is some incompatibility between BERT and other traditional NLP features which prevents them from being combined effectively. In contrast, static word embeddings (e.g., Word2Vec) do not seem to suffer from the same problem.

One possible explanation could be because the late fusion strategy we adopted to combine BERT embeddings with other features, which is different from the early fusion strategy we used to combine Word2Vec embeddings. To test this hypothesis, instead of merging word embedding with other word-level features before inputting to BiLSTM, we first adopted Doc2Vec (D2V) [38] which employs an embedding learning method similar to Word2Vec to learn sentence embeddings. Similar to Word2Vec, our D2V model was trained on the Medical Information Mart for Intensive Care (MIMIC)-III clinical corpus [27] and the i2b2 dataset. As shown in Figure 3b, we adopt late fusion to combine Doc2Vec-based sentence embedding with the output of BiLSTM.

As shown in Table VII, Doc2Vec-based sentence embedding on its own (F1=0.5622) is much worse than either BERT sentence embedding (F1=0.7447) or Word2Vec (F1=0.6974). When combining Doc2Vec with other features however, the performance improvement is 26%, which is the highest among all the text embedding models. The full model with Doc2Vec embeddings (F1=0.8249) also outperformed the model with BERT embeddings (F1=0.7646) by (6%). Since the model with Doc2Vec also adopts a late fusion strategy, late fusion may not directly cause the poor performance of BERT models.

In summary, if used alone, BERT embedding is the best for relation extraction. However, Word2Vec-based static embedding works the best when combined with other NLP features. In contrast, models with BERT sentence embedding worked poorly, which is surprising.

V. Conclusions

In this research, we investigate how different neural network architectures (e.g., BiLSTM, CNN, ResNet, GCN) and diverse features sets (e.g., BERT contextual embedding, Word2Vec static embedding, POS, dependency relations, IOB encoding, surface position) can be used in relation extraction from

clinical texts. Our experiment results demonstrate that BiLSTM model with static Word2Vec embedding plus traditional syntactic and semantic features is most effective for such a task. It significantly outperformed two baselines on the same dataset. Although contextual embedding learned by BERT on its own is very effective, it performed poorly when combined with other traditional NLP features. More investigations are needed to understand why.

REFERENCES

[1] S. Sahu, A. Anand, K. Oruganty, and M. Gattu, "Relation extraction from clinical texts using domain invariant convolutional neural network," in *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, Berlin, Germany, Aug. 2016.

[2] Z. Li, Z. Yang, C. Shen, J. Xu, Y. Zhang, and H. Xu, "Integrating shortest dependency path and sentence sequence into a deep learning framework for relation extraction in clinical text," *BMC medical informatics and decision making*, vol. 19, no. 1, p. 22, 2019.

[3] Z. Guo, Y. Zhang, and W. Lu, "Attention guided graph convolutional networks for relation extraction," in *Proceedings of the 57th Annual Meeting of the ACL*, 2019.

[4] A. Roy, Y. Park, T. Lee, and S. Pan, "Supervising unsupervised open information extraction models," in *Proceedings of the 2019 Conference on EMNLP-IJCNLP*, Hong Kong, China, Nov. 2019.

[5] Y. Shen and X. Huang, "Attention-based convolutional neural network for semantic relation extraction," in *Proceedings of COLING 2016*, Osaka, Japan, Dec. 2016.

[6] Ö. Uzuner, B. R. South, S. Shen, and S. L. DuVall, "2010 i2b2/va challenge on concepts, assertions, and relations in clinical text," *JAMIA*, no. 5, pp. 552–556, 2011.

[7] Ö. Uzuner, J. Mailoa, R. Ryan, and T. Sibanda, "Semantic relations for problem-oriented medical records," *Artificial intelligence in medicine*, vol. 50, no. 2, pp. 63–73, 2010.

[8] K. Roberts, B. Rink, and S. Harabagiu, "Extraction of medical concepts, assertions, and relations from discharge summaries for the fourth i2b2/va shared task," in *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data. Boston, MA, USA: i2b2*, 2010.

[9] G. Divita, O. Treitler, Y. Kim *et al.*, "Salt lake city va's challenge submissions," in *proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*, 2010.

[10] I. Solt, F. P. Szidarovszky, and D. Tikk, "Concept, assertion and relation extraction at the 2010 i2b2 relation extraction challenge using parsing information and dictionaries," *Proc. of i2b2/VA Shared-Task. Washington, DC*, 2010.

[11] J. Patrick, D. Nguyen, Y. Wang, and M. Li, "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. Boston, MA, USA: i2b2*, 2010.

[12] S. Jonnalagadda and G. Gonzalez, "Can distributional statistics aid clinical concept extraction," in *Proceedings of the 2010 i2b2/VA workshop on challenges in natural language processing for clinical data. Boston, MA, USA: i2b2*, 2010.

[13] P. Anick, P. Hong, N. Xue, and D. Anick, "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.

[14] A. M. Cohen, K. Ambert, J. Yang, R. Felder, R. Sproat, B. Roark, K. Hollingshead, and K. Baker, "Ohsu/portland vamc team participation in the 2010 i2b2/va challenge tasks," in *Proceedings of the 2010 i2b2/VA Workshop on Challenges in Natural Language Processing for Clinical Data*. i2b2, Boston, MA, USA, 2010.

[15] D. Demner-Fushman, E. Apostolova, R. Islamaj Dogan, F.-M. Lang, J. Mork, A. Neveol, S. Shooshan, M. Simpson, and A. Aronson, "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. Boston, MA, USA: i2b2*, 2010.

[16] C. Grouin, A. B. Abacha, D. Bernhard, B. Cartoni, L. Deleger, B. Grau, A.-L. Ligozat, A.-L. Minard, S. Rosset, and P. Zweigenbaum, "Caramba: concept, assertion, and relation annotation using machine-learning based approaches," in *i2b2 Medication Extraction Challenge Workshop*, 2010.

[17] B. de Bruijn, C. Cherry, S. Kiritchenko, J. Martin, and X. Zhu, "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. Boston, MA, USA: i2b2*, 2010.

[18] R. J. Ryan, "Groundtruth budgeting: a novel approach to semi-supervised relation extraction in medical language," Ph.D. dissertation, Massachusetts Institute of Technology, 2011.

[19] A.-L. Minard, A.-L. Ligozat, and B. Grau, "Multi-class SVM for relation extraction from clinical reports," in *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*. Hissar, Bulgaria: Association for Computational Linguistics, Sep. 2011.

[20] Y. Luo, Y. Cheng, Ö. Uzuner, P. Szolovits, and J. Starren, "Segment convolutional neural networks (seg-cnns) for classifying relations in clinical notes," *JAMIA*, vol. 25, no. 1, pp. 93–98, 2018.

[21] D. Ningthoujam, S. Yadav, P. Bhattacharyya, and A. Ekbal, "Relation extraction between the clinical entities based on the shortest dependency path based lstm," *arXiv preprint arXiv:1903.09941*, 2019.

[22] Y. Peng, S. Yan, and Z. Lu, "Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets," in *Proceedings of the 18th BioNLP Workshop and Shared Task*, 2019, pp. 58–65.

[23] Q. Jin, B. Dhingra, W. Cohen, and X. Lu, "Probing biomedical embeddings from language models," *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pp. 82–89, 2019.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the NAACL-HLT, Volume 1*, pp. 4171–4186, 2019.

[25] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[26] Q. Wei, Z. Ji, Y. Si, J. Du, J. Wang, F. Tiryaki, S. Wu, C. Tao, K. Roberts, and H. Xu, "Relation extraction from clinical narratives using pre-trained language models," in *AMIA Annual Symposium Proceedings*, vol. 2019. American Medical Informatics Association, 2019, p. 1236.

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

[28] E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jindi, T. Naumann, and M. McDermott, "Publicly available clinical bert embeddings," *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pp. 72–78, 2019.

[29] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the ACL: system demonstrations*, 2014, pp. 55–60.

[30] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," in *Proceedings of the 2018 Conference on EMNLP*, 2018, pp. 2205–2215.

[31] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[32] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[34] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *ICML*, 2015, pp. 448–456.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[37] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *NIPS*, 2016, pp. 550–558.

[38] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *ICML*, 2014, pp. 1188–1196.