A Graph-Based Framework for Information Extraction

by

Yujie Qian

B.Eng., Tsinghua University (2017)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2019

Signature redacted

Author ....
Department of Electrical Engineering and Computer Science
May 23, 2019

Signature redacted

Certified by..
Regina Barzilay
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Signature redacted

Accepted by ....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

A Graph-Based Framework for Information Extraction

by

Yujie Qian

Submitted to the Department of Electrical Engineering and Computer Science
on May 23, 2019, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Most modern Information Extraction (IE) systems are implemented as sequential taggers and only model local dependencies. Non-local and non-sequential context is, however, a valuable source of information to improve predictions. In this thesis, we introduce a graph-based framework (GraphIE) that operates over a graph representing a broad set of dependencies between textual units (i.e. words or sentences). The algorithm propagates information between connected nodes through graph convolutions, generating a richer representation that can be exploited to improve word-level predictions. Evaluation on three different tasks — namely textual, social media and visual information extraction — shows that GraphIE consistently outperforms the state-of-the-art sequence tagging model by a significant margin.

Thesis Supervisor: Regina Barzilay
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

I would like to express my deepest appreciation to my advisor, Prof. Regina Barzilay, for her valuable guidance and support during the past two years. She kindly helps me adapt to the life and study in this country, and is always patient to listen to my thoughts. Her research vision and insights always inspire me to think creatively and work hard to make innovations. I believe it will be an amazing PhD journey to work with her in the following years.

I am also grateful to my labmates at MIT NLP group, who explore the research area together with me and share with me their brilliant ideas. I have learned a lot from them and I feel fortunate to have these great friends.

I would also like to thank my wonderful collaborators – Enrico Santus, Jiang Guo, and Zhijing Jin. This thesis is partially based on our prior peer-reviewed publication [24].

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Information Extraction

Information extraction (IE) is the task of automatically extracting structured information from unstructured documents. Most modern Information Extraction systems are implemented as sequential taggers. While such models effectively capture relations in the local context, they have limited capability of exploiting non-local and non-sequential dependencies. In many applications, however, such dependencies can greatly reduce tagging ambiguity, thereby improving overall extraction performance. For instance, when extracting entities from a document, various types of non-local contextual information such as co-references and identical mentions may provide valuable cues. See for example Figure 1-1, in which the non-local relations are crucial to discriminate the entity type of the second mention of *Washington* (i.e. PERSON or LOCATION).

Most of the prior work looking at the non-local dependencies incorporates them by constraining the output space in a structured prediction framework [4, 26, 6]. Such approaches, however, mostly overlook the richer set of structural relations in the input space. With reference to the example in Figure 1-1, the co-referent dependencies would not be readily exploited by simply constraining the output space, as they would not necessarily be labeled as entities (e.g. pronouns). In the attempt to capture non-local dependencies in the input space, alternative approaches define a graph

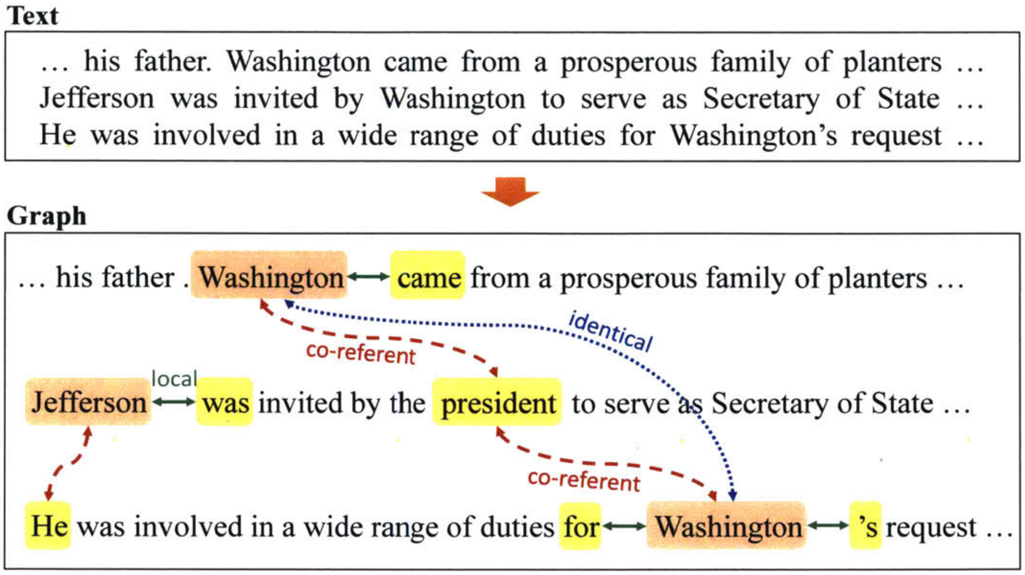

Figure 1-1: Example of the entity extraction task with an ambiguous entity mention (i.e. "...for Washington's request..."). Aside from the sentential forward and backward edges (green, solid) which aggregate local contextual information, non-local relations — such as the co-referent edges (red, dashed) and the identical-mention edges (blue, dotted) — provide additional valuable information to reduce tagging ambiguity (i.e. PERSON or LOCATION).

that outlines the input structure and engineer features to describe it [25]. Designing effective features is however challenging, arbitrary and time consuming, especially when the underlying structure is complex. Moreover, these approaches have limited capacity of capturing node interactions informed by the graph structure.

1.2 Proposed Framework: GraphIE

In this thesis, we propose a graph-based information extraction framework (GraphIE). The framework improves IE predictions by automatically learning the interactions between local and non-local dependencies in the input space. Specifically, our approach integrates a graph module with the encoder-decoder architecture for sequence tagging. The algorithm operates over a graph, where nodes correspond to textual units (i.e. words or sentences) and edges describe their relations. At the core of our model, a recurrent neural network sequentially encodes local contextual representations and then the graph module iteratively propagates information between neighboring nodes using graph convolutions [9]. The learned representations are finally projected back to a recurrent decoder to support tagging at the word level.

We evaluate the proposed GraphIE framework on three IE tasks, namely textual, social media, and visual [1] information extraction. For each task, we provide in input a simple task-specific graph, which defines the data structure without access to any major processing or external resources. Our model is expected to learn from the relevant dependencies to identify and extract the appropriate information. Experimental results on multiple benchmark datasets show that GraphIE consistently outperforms a strong and commonly adopted sequential model (SeqIE, i.e. a bi-directional long-short term memory (BiLSTM) followed by a conditional random fields (CRF) module). Specifically, in the textual IE task, we obtain an improvement of 0.5% over SeqIE on the CoNLL03 dataset, and an improvement of 1.4% on the chemical entity extraction [10]. In the social media IE task, GraphIE improves over SeqIE by 3.7% in extracting the Education attribute from twitter users. In visual IE, finally, we outperform the baseline by 1.2%. Our code and data are available at `https://github.com/thomas0809/GraphIE`.

1.3 Related Work

The problem of incorporating non-local and non-sequential context to improve information extraction has been extensively studied in the literature. The majority of methods have focused on enforcing constraints in the output space during inference, through various mechanisms such as posterior regularization or generalized expectations [4, 17, 26, 15, 6]. Research capturing non-local dependencies in the input space have mostly relied on feature-based approaches. Roberts et al. [27] and Swampillai and Stevenson [30] have designed intra- and inter-sentential features based on discourse and syntactic dependencies (e.g., shortest paths) to improve relation extraction. Quirk and Poon [25] used document graphs to flexibly represent multiple types of relations between words (e.g., syntactic, adjacency and discourse relations).

Graph-based representations can be also learned with neural networks. The most related work to ours is the graph convolutional network by Kipf and Welling [9], which was developed to encode graph structures and perform node classification. In our framework, we adapt GCN as an intermediate module that learns non-local context, which — instead of being used directly for classification — is projected to the decoder to enrich local information and perform sequence tagging.

A handful of other information extraction approaches have used graph-based neural networks. Miwa and Bansal [20] applied Tree LSTM [31] to jointly represent sequences and dependency trees for entity and relation extraction. On the same line of work, Peng et al. [21] and Song et al. [29] introduced Graph LSTM, which extended the traditional LSTM to graphs by enabling a varied number of incoming edges at each memory cell. Zhang et al. [34] exploited graph convolutions to pool information over pruned dependency trees, outperforming existing sequence and dependency-based neural models in a relation extraction task. These studies differ from ours in several respects. First, they can only model word-level graphs, whereas our framework can learn non-local context either from word- or sentence-level graphs, using it to reduce ambiguity during tagging at the word level. Second, all these studies achieved improvements only when using dependency trees. We extend the graph-based approach to validate the benefits of using other types of relations in a broader range of tasks, such as co-reference in named entity recognition, *followed-by* link in social media, and layout structure in visual information extraction.

Chapter 2

GraphIE Framework

2.1 Problem Definition

We first formalize a novel problem named *graph-based information extraction.* Rather than simply modeling inputs as sequences, we assume there exists a graph structure in the data that can be exploited to capture non-local and non-sequential dependencies between textual units, namely words or sentences.

In *graph-based information extraction*, the input consists of a set of sentences $S = \{s_1, \ldots, s_N\}$ and an auxiliary graph $G = (V, E)$, where $V = \{v_1, \ldots, v_M\}$ is the node set and $E \subset V \times V$ is the edge set. Each sentence is a sequence of words. We consider two different designs of the graph:

(1) *sentence-level graph*, where each node is a sentence (i.e. $M = N$), and the edges encode sentence dependencies;

(2) *word-level graph*, where each node is a word (i.e. M is the number of words in the input), and the edges connect pairs of words, such as co-referent tokens.

The edges $e_{i,j} = (v_i, v_j)$ in the graph can be either directed or undirected. Multiple edge types can also be defined to capture different structural factors underlying the task-specific input data.

To obtain the extracted text span, we use the BIO (Begin, Inside, Outside) tagging scheme in this thesis. For each sentence $s_i = (w_1^{(i)}, w_2^{(i)}, \dots, w_k^{(i)})$,[1] we sequentially tag each word as $\mathbf{y}_i = (y_1^{(i)}, y_2^{(i)}, \dots, y_k^{(i)})$.

2.2 Method

In this thesis, we propose a general framework for graph-based information extraction named GraphIE. GraphIE jointly learns local and non-local dependencies by iteratively propagating information between node representations. Our framework has three components:

- an *encoder*, which generates local context-aware hidden representations for the textual unit (i.e. word or sentence, depending on the task) with a recurrent neural network;
- a *graph module*, which captures the graph structure, learning non-local and non-sequential dependencies between textual units;
- a *decoder*, which exploits the contextual information generated by the graph module to perform labelling at the word level.

Figure 2-1 illustrates the overview of GraphIE and the model architectures for both sentence- and word-level graphs. In the following sections, we first introduce the case of the sentence-level graph, and then we explain how to adapt the model for the word-level graph.

2.2.1 Encoder

In GraphIE, we first use an encoder to generate text representations. Given a sentence $s_i = (w_1^{(i)}, w_2^{(i)}, \dots, w_k^{(i)})$ of length k, each word $w_t^{(i)}$ is represented by a vector $\mathbf{x}_t^{(i)}$, which is the concatenation of its word embedding and a feature vector learned with a character-level convolutional neural network (CharCNN; Kim et al. [7]). We encode

[1]While sentences may have different lengths, for notation simplicity we use a single variable k.

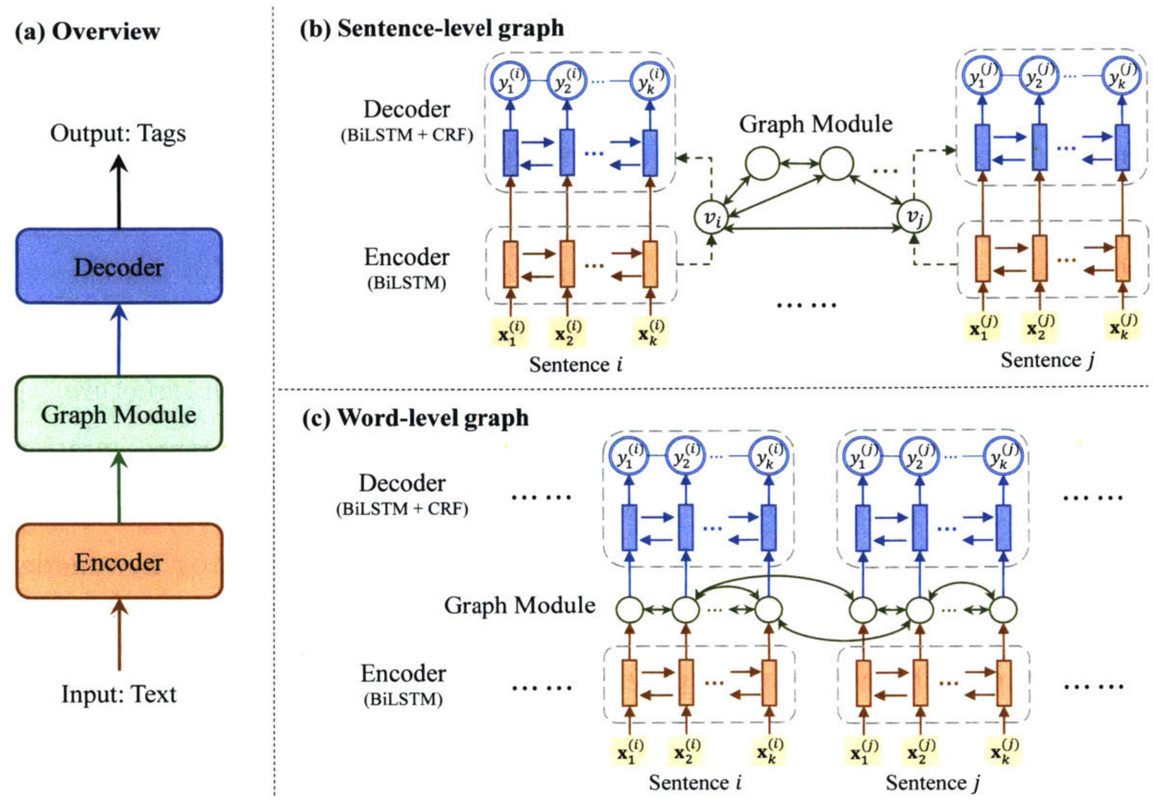

Figure 2-1: GraphIE framework: (a) an overview of the framework; (b) architecture for *sentence-level graph*, where each sentence is encoded to a node vector and fed into the graph module, and the output of the graph module is used as the initial state of the decoder; (c) architecture for *word-level graph*, where the hidden state for each word of the encoder is taken as the input node vector of the graph module, and then the output is fed into the decoder.

the sentence with a recurrent neural network (RNN), defining it as

$$\mathbf{h}_{1:k}^{(i)} = \texttt{RNN}\left(\mathbf{x}_{1:k}^{(i)}\ ; \mathbf{0}, \Theta_{\text{enc}}\right), \tag{2.1}$$

where $\mathbf{x}_{1:k}^{(i)}$ denotes the input sequence $[\mathbf{x}_1^{(i)}, \cdots, \mathbf{x}_k^{(i)}]$, $\mathbf{h}_{1:k}^{(i)}$ denotes the hidden states $[\mathbf{h}_1^{(i)}, \cdots, \mathbf{h}_k^{(i)}]$, $\mathbf{0}$ indicates the initial hidden state is zero, and Θ_{enc} represents the encoder parameters. We implement the RNN as a bi-directional LSTM [5], and encode each sentence independently.

We obtain the sentence representation for s_i by averaging the hidden states of its words, i.e. $\texttt{Enc}(s_i) = \frac{1}{k}\left(\sum_{t=1}^{k} \mathbf{h}_t^{(i)}\right)$. The sentence representations are then fed into the graph module.

2.2.2 Graph Module

The graph module is designed to learn the non-local and non-sequential information from the graph. We adapt the graph convolutional network (GCN) to model the graph context for information extraction.

Given the sentence-level graph $G = (V, E)$, where each node v_i (i.e. sentence s_i) has the encoding $\texttt{Enc}(s_i)$ capturing its local information, the graph module enriches such representation with neighbor information derived from the graph structure.

Our graph module is a GCN which takes as input the sentence representation, i.e. $\mathbf{g}_i^{(0)} = \texttt{Enc}(s_i)$, and conducts graph convolution on every node, propagating information between its neighbors, and integrating such information into a new hidden representation. Specifically, each layer of GCN has two parts. The first gets the information of each node from the previous layer, i.e.

$$\boldsymbol{\alpha}_i^{(l)} = \mathbf{W}_v^{(l)}\ \mathbf{g}_i^{(l-1)}, \tag{2.2}$$

where $\mathbf{W}_v^{(l)}$ is the weight to be learned. The second aggregates information from the neighbors of each node, i.e. for node v_i, we have

$$\boldsymbol{\beta}_i^{(l)} = \frac{1}{d(v_i)} \cdot \mathbf{W}_e^{(l)} \left(\sum_{e_{i,j} \in E} \mathbf{g}_j^{(l-1)} \right), \tag{2.3}$$

where $d(v_i)$ is the degree of node v_i (i.e. the number of edges connected to v_i) and is used to normalize $\boldsymbol{\beta}_i^{(l)}$, ensuring that nodes with different degrees have representations of the same scale.[2] In the simplest case, where the edges in the graph are undirected and have the same type, we use the same weight $\mathbf{W}_e^{(l)}$ for all of them. In a more general case, where multiple edge types exist, we expect them to have different impacts on the aggregation. Thus, we model these edge types with different weights in Eq. 2.3, similar to the relational GCN proposed by Schlichtkrull et al. [28]. When edges are directed, i.e. edge $e_{i,j}$ is different from $e_{j,i}$, the propagation mechanism should mirror

[2] We choose this simple normalization strategy instead of the two-sided normalization in Kipf and Welling [9], as it performs better in the experiments. The same strategy is also adopted by Zhang et al. [34].

such difference. In this case, we consider directed edges as two types of edges (forward and backward), and use different weights for them.

Finally, $\boldsymbol{\alpha}_i^{(l)}$ and $\boldsymbol{\beta}_i^{(l)}$ are combined to obtain the representation at the l-th layer,

$$\mathbf{g}_i^{(l)} = \sigma\left(\boldsymbol{\alpha}_i^{(l)} + \boldsymbol{\beta}_i^{(l)} + b^{(l)}\right), \tag{2.4}$$

where $\sigma(\cdot)$ is the non-linear activation function, and $b^{(l)}$ is a bias parameter.

Because each layer only propagates information between directly connected nodes, we can stack multiple graph convolutional layers to get a larger receptive field, i.e. each node can be aware of more distant neighbors. After L layers, for each node v_i we obtain a contextual representation, $\texttt{GCN}(s_i) = \mathbf{g}_i^{(L)}$, that captures both local and non-local information.

2.2.3 Decoder

To support tagging, the learned representation is propagated to the decoder.

In our work, the decoder is instantiated as a BiLSTM+CRF tagger [12]. The output representation of the graph module, $\texttt{GCN}(s_i)$, is split into two vectors of the same length, which are used as the initial hidden states for the forward and backward LSTMs, respectively. In this way, the graph contextual information is propagated to each word through the LSTM. Specifically, we have

$$\mathbf{z}_{1:k}^{(i)} = \texttt{RNN}\left(\mathbf{h}_{1:k}^{(i)}\,; \texttt{GCN}(s_i), \Theta_{\text{dec}}\right), \tag{2.5}$$

where $\mathbf{h}_{1:k}^{(i)}$ are the output hidden states of the encoder, $\texttt{GCN}(s_i)$ represents the initial state, and Θ_{dec} is the decoder parameters. A simpler way to incorporate the graph representation into the decoder is concatenating with its input, but the empirical performance is worse than using as the initial state.

Finally, we use a CRF layer [11] on top of the BiLSTM to perform tagging,

$$\mathbf{y}_i^* = \underset{\mathbf{y}\in\mathbf{Y}_k}{\arg\max}\, p\left(\mathbf{y} \mid \mathbf{z}_{1:k}^{(i)}\,; \Theta_{\text{crf}}\right), \tag{2.6}$$

where $\mathbf{Y}_k$ is the set of all possible tag sequences of length k, and Θ_{crf} represents the CRF parameters, i.e. transition scores of tags. CRF combines the local predictions of BiLSTM and the transition scores to model the joint probability of the tag sequence.[3]

2.2.4 Adaptation to Word-level Graphs

GraphIE can be easily adapted to model word-level graphs. In such case, the nodes represent words in the input, i.e. the number of nodes M equals the total number of words in the N sentences. At this point, each word's hidden state in the encoder can be used as the input node vector $\mathbf{g}_i^{(0)}$ of the graph module. GCN can then conduct graph convolution on the word-level graph and generate graph-contextualized representations for the words. Finally, the decoder directly operates on the GCN's outputs, i.e. we change the BiLSTM decoder to

$$\mathbf{z}_{1:k}^{(i)} = \texttt{RNN}\left(\left[\texttt{GCN}(w_1^{(i)}), \cdots, \texttt{GCN}(w_k^{(i)})\right]; \mathbf{0}, \Theta_{\text{dec}}\right), \tag{2.7}$$

where $\texttt{GCN}(w_t^{(i)})$ is the GCN output for word $w_t^{(i)}$. In this case, the BiLSTM initial states are set to the default zero vectors. The CRF layer remains unchanged.

As it can be seen in Figure 2-1(c), the word-level graph module differs from the sentence-level one because it directly takes the word representations from the encoder and feeds its output to the decoder. In sentence-level graph, the GCN operates on sentence representations, which are then used as the initial states of the decoder BiLSTM.

2.3 Applications

The proposed GraphIE framework can be applied to various information extraction tasks with specified graphs. In this thesis, we evaluate GraphIE on three different IE

[3]In GraphIE, the graph module models the input space structure, i.e. the dependencies between textual units (i.e. sentences or words), and the final CRF layer models the sequential connections of the output tags. Even though loops may exist in the input graph, CRF operates sequentially, thus the inference is tractable.

Task	Graph Type	Node	Edge
Textual IE	word-level	word	1. non-local (identical mentions) 2. local sentential (forward, backward)
Social Media IE	sentence-level	tweets	*followed-by*
Visual IE	sentence-level	text box	spatial layout (horizontal, vertical)

Table 2.1: Comparison of graph structure in the three IE tasks used for evaluation.

tasks:

(1) **textual information extraction**, where we focus on named entity recognition at discourse level;

(2) **social media information extraction**, where we extract information from users' posts in online social networks;

(3) **visual information extraction**, where we aim to extract several attribute values from documents formatted in various layouts.

For each of these tasks, we created a simple task-specific graph topology, designed to easily capture the underlying structure of the input data without any major processing. Table 2.1 summarizes the three tasks. In the subsequent three chapters, we will present the experimental setup and results for the three tasks.[4]

[4] Our code and data are available at `https://github.com/thomas0809/GraphIE`.

Chapter 3

Textual Information Extraction

In the first task, we focus on named entity recognition at discourse level (DiscNER). In contrast to traditional sentence-level NER (SentNER), where sentences are processed independently, in DiscNER, long-range dependencies and constraints across sentences have a crucial role in the tagging process. For instance, multiple mentions of the same entity are expected to be tagged consistently in the same discourse. Here we propose to use this (soft) constraint to improve entity extraction.

3.1 Experimental Setup

3.1.1 Dataset

We conduct experiments on two NER datasets: the CoNLL-2003 dataset (CoNLL03) [32], and the CHEMDNER dataset for chemical entity extraction [10]. We follow the standard split of each corpora. Statistics are shown in Table 3.1.

3.1.2 Graph Construction

In this task, we use a word-level graph where nodes represent words. We create two types of edges for each document:

- *Local edges*: forward and backward edges are created between neighboring words in each sentence, allowing local contextual information to be utilized.

Dataset		Train	Dev	Test
CoNLL03	#doc	946	216	231
	#sent	14,987	3,466	3,684
Chemdner	#doc	3,500	3,500	3,000
	#sent	30,739	30,796	26,399

Table 3.1: Statistics of the CoNLL03 and the Chemdner datasets.

- *Non-local edges*: re-occurrences of the same token other than stop words are connected, so that information can be propagated through, encouraging global consistency of tagging.[1]

3.1.3 Baseline

We implement a two-layer BiLSTM with a conditional random fields (CRF) tagger as the sequential baseline (SeqIE). This architecture and its variants have been extensively studied and demonstrated to be successful in previous work on information extraction [12, 16]. The baseline shares the same encoder and decoder architecture with GraphIE, but without the graph module.

3.2 Results

Table 3.2 describes the NER accuracy on the CoNLL03 [32] and the Chemdner [10] datasets.

For CoNLL03, we list the performance of existing approaches. Our baseline SeqIE obtains competitive scores compared to the best methods. The fact that GraphIE significantly outperforms it, highlights once more the importance of modeling non-local and non-sequential dependencies and confirms that our approach is an appropriate method to achieve this goal.[2]

[1]Note that other non-local relations such as co-references (cf. the example in Figure 1-1) may be used for further improvement. However, these relations require additional resources to obtain, and we leave them to future work.

[2]We achieve the best reported performance among methods not using the recently introduced ELMo [23] and BERT [3], which are pretrained on extra-large corpora and computationally demand-

DATASET	Model	F1
CoNLL03	Lample et al. [12]	90.94
	Ma and Hovy [16]	91.21
	Ye and Ling [33]	91.38
	SeqIE	91.16
	GraphIE	**91.74***
CHEMDNER	Krallinger et al. [10]	87.39
	SeqIE	88.28
	GraphIE	**89.71***

Table 3.2: NER accuracy on the CoNLL03 and the CHEMDNER datasets. Scores for our methods are the average of 5 runs. * indicates statistical significance of the improvement over SeqIE ($p < 0.01$).

For CHEMDNER, we show the best performance reported in Krallinger et al. [10], obtained with a feature-based method. Our baseline outperforms the feature-based method, and GraphIE further improves the performance by 1.4%.

Analysis To understand the advantage of GraphIE, we first investigate the importance of graph structure to the model. As shown in Figure 3-1, using random connections clearly hurts the performance, bringing down the F1 score of GraphIE from 95.12% to 94.29%. It indicates that the task-specific graph structures introduce beneficial inductive bias. Trivial feature augmentation also does not work well, confirming the necessity of learning the graph embedding with GCN.

We further conduct error analysis on the test set to validate our motivation that GraphIE resolves tagging ambiguity by encouraging consistency among identical entity mentions (cf. Figure 1-1). Here we examine the word-level tagging accuracy. We define the words that have more than one possible tags in the dataset as *ambiguous*. We find that among the 1.78% tagging errors of SeqIE, 1.16% are *ambiguous* and 0.62% are *unambiguous*. GraphIE reduces the error rate to 1.67%, with 1.06% to be *ambiguous* and 0.61% *unambiguous*. We can see that most of the error reduction indeed attributes to the *ambiguous* words.

ing.

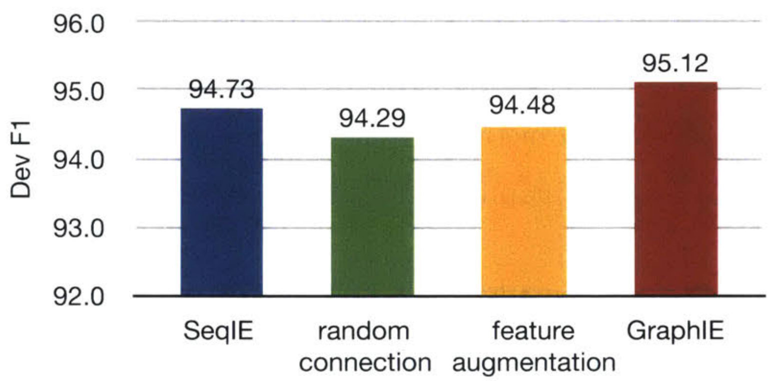

Figure 3-1: Analysis on the CoNLL03 dataset. We compare with two alternative designs: (1) *random connection*, where we replace the constructed graph by a random graph with the same number of edges; (2) *feature augmentation*, where we use the average embedding of each node and its neighbors as the input to the decoder, instead of the GCN which has additional parameters. We report F1 scores on the development set.

Chapter 4

Social Media Information Extraction

Social media information extraction refers to the task of extracting information from users' posts in online social networks [2, 14]. In this chapter, we aim at extracting *education* and *job* information from users' tweets. Given a set of tweets posted by a user, the goal is to extract mentions of the organizations to which they belong. The fact that the tweets are short, highly contextualized and show special linguistic features makes this task particularly challenging.

4.1 Experimental Setup

4.1.1 Dataset

We construct two datasets, EDUCATION and JOB, from the Twitter corpus released by Li et al. [14]. The original corpus contains millions of tweets generated by ≈ 10 thousand users, where the *education* and *job* mentions are annotated using distant supervision [18]. We sample the tweets from each user, maintaining the ratio between positive and negative posts.[1] The obtained EDUCATION dataset consists of 443,476 tweets generated by 7,208 users, and the JOB dataset contains 176,043 tweets generated by 1,772 users. Dataset statistics are reported in Table 4.1.

[1]Positive and negative refer here to whether or not the *education* or *job* mention is present in the tweet.

	EDUCATION	JOB
Users	7,208	1,772
Edges	11,167	3,498
Positive Tweets	49,793	3,694
Negative Tweets	393,683	172,349

Table 4.1: Statistics of the EDUCATION and JOB datasets.

The datasets are both split in 60% for training, 20% for development, and 20% for testing. We perform 5 different random splits and report the average results.

4.1.2 Graph Construction

We construct the graph as *ego-networks* [13], i.e. when we extract information about one user, we consider the subgraph formed by the user and his/her direct neighbors. Note that we use the sentence-level graph architecture in this task. Each node corresponds to a Twitter user, who is represented by the set of posted tweets.[2] Edges are defined by the *followed-by* link, under the assumption that connected users are more likely to come from the same university or company. An example of the social media graph is reported in Figure 4-1.

4.2 Results

Table 4.2 shows the results for the social media information extraction task. We first report a simple dictionary-based method as a baseline. Neural IE models achieve much better performance, showing that meaningful patterns are learned by the models rather than simply remembering the entities in the training set. The proposed GraphIE outperforms SeqIE in both the EDUCATION and JOB datasets, and the improvements are more significant for the EDUCATION dataset (3.7% versus 0.3%). The reason for such difference is the variance in the affinity scores [19] between the two datasets. Li et al. [14] underline that affinity value for EDUCATION is 74.3 while for

[2] As each node is a set of tweets posted by the user, we encode every tweet with the encoder, and then average them to obtain the node representation. In the decoding phase, the graph module's output is fed to the decoder for each tweet.

Dataset	Dictionary			SeqIE			GraphIE		
	P	R	F1	P	R	F1	P	R	F1
Education	78.7	93.5	85.4	85.2	93.6	89.2	92.9	92.8	**92.9***
Job	55.7	70.2	62.1	66.2	66.7	66.2	67.1	66.1	**66.5**

Table 4.2: Extraction accuracy on the Education and Job datasets. *Dictionary* is a naive method which creates a dictionary of entities from the training set and extracts their mentions during testing time. SeqIE is the same as described in Section 3.1.3. Scores are the average of 5 runs. * indicates the improvement over SeqIE is statistically significant (Welch's t-test, $p < 0.01$).

Job it is only 14.5, which means that in the datasets neighbors are 5 times more likely to have studied in the same university than worked in the same company. We can therefore expect that a model like GraphIE, which exploits neighbors' information, obtains larger advantages in a dataset characterized by higher affinity.

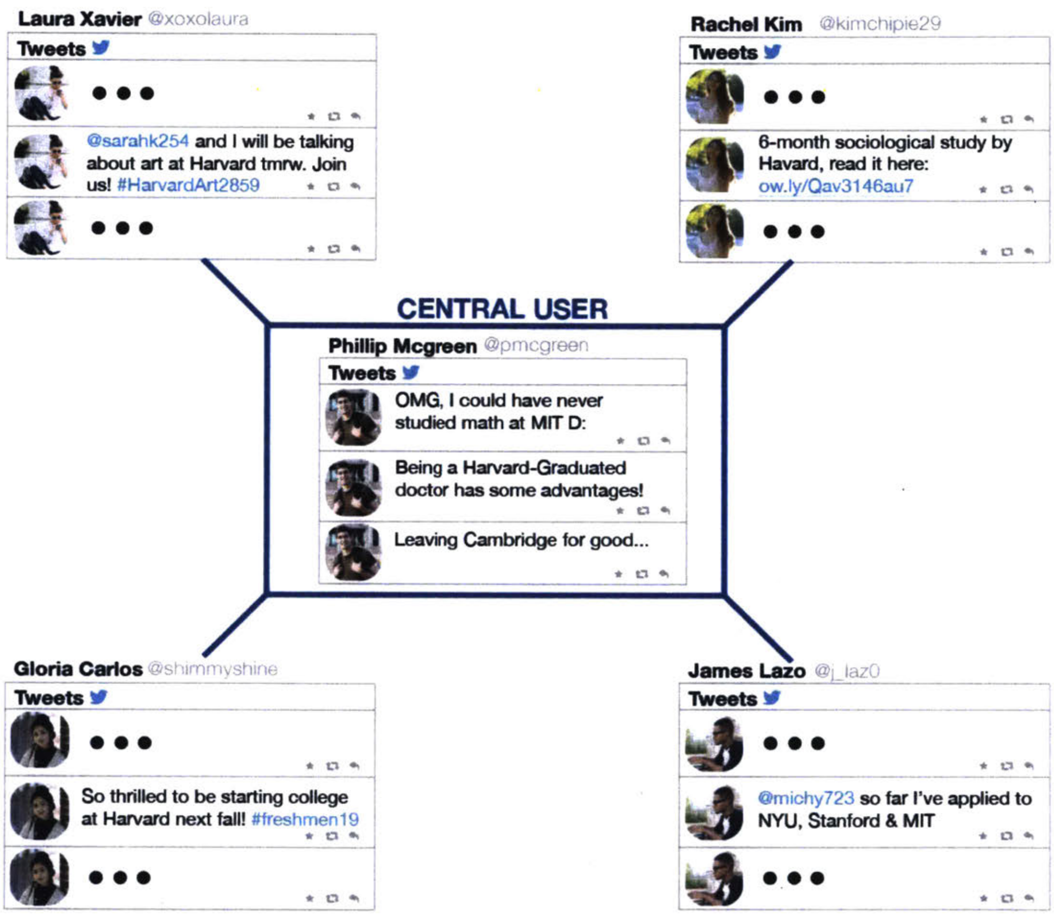

Figure 4-1: Mock-up example of Social Media Information Extraction. Nodes are represented as users and edges are *follow-by* relations.

Chapter 5

Visual Information Extraction

Visual information extraction refers to the extraction of attribute values from documents formatted in various layouts. Examples include invoices and forms, whose format can be exploited to infer valuable information to support extraction.

5.1 Experimental Setup

5.1.1 Dataset

The corpus consists of 25,200 Adverse Event Case Reports (AECR) recording drug-related side effects. Each case contains an average of 9 pages. Since these documents are produced by multiple organizations, they exhibit large variability in the layout and presentation styles (e.g. text, table, etc.).[1] The collection is provided with a separate human-extracted ground truth database that is used as a source of distant supervision.

Our goal is to extract eight attributes related to the patient, the event, the drug and the reporter (cf. Table 5.1 for the full list). Attribute types include dates, words and phrases — which can be directly extracted from the document.

The dataset is split in 50% cases for training, 10% for development, and 40% for testing.

[1]This dataset cannot be shared for patient privacy and proprietary issues.

Attribute	SeqIE			GraphIE		
	P	R	F1	P	R	F1
Patient Initials	93.5	92.4	**92.9**	93.6	91.9	92.8
Patient Age	94.0	91.6	92.8	94.8	91.1	**92.9**
Patient Birthday	96.6	96.0	**96.3**	96.9	94.7	95.8
Drug Name	71.2	51.2	59.4	78.5	50.4	**61.4**
Event	62.6	65.2	63.9	64.1	68.7	**66.3**
Reporter First Name	78.3	95.7	86.1	79.5	95.9	**86.9**
Reporter Last Name	84.5	68.4	75.6	85.6	68.2	**75.9**
Reporter City	88.9	65.4	75.4	92.1	66.3	**77.1**
Avg. (macro)	83.7	78.2	80.3	85.7	78.4	**81.1**†
Avg. (micro)	78.5	73.8	76.1	80.3	74.6	**77.3**†

Table 5.1: Extraction accuracy on the AECR dataset. To increase the competitiveness of the SeqIE baseline (cf. Section 3.1.3), we sequentially concatenate the horizontally aligned text boxes and feed into the model, therefore fully utilizing the horizontal edges of the graph. Scores are the average of 5 runs. † indicates statistical significance of the improvement over SeqIE ($p < 0.05$).

5.1.2 Graph Construction

We first turn the PDFs to text using PDFMiner,[2] which provides words along with their positions in the page (i.e. bounding-box coordinates). Consecutive words are then geometrically joined into *text boxes*. Each text box is considered as a "sentence" in this task, and corresponds to a *node* in the graph.

Since the page layout is the major structural factor in these documents, we work on page-by-page basis, i.e. each page corresponds to a graph. The *edges* are defined to horizontally or vertically connect *nodes* (text boxes) that are close to each other (i.e. when the overlap of their bounding boxes, in either the vertical or horizontal direction, is over 50%). Four types of edge are considered: left-to-right, right-to-left, up-to-down, and down-to-up. When multiple nodes are aligned, only the closest ones are connected. An example of visual document graph is reported in Figure 5-1.

[2] https://euske.github.io/pdfminer/

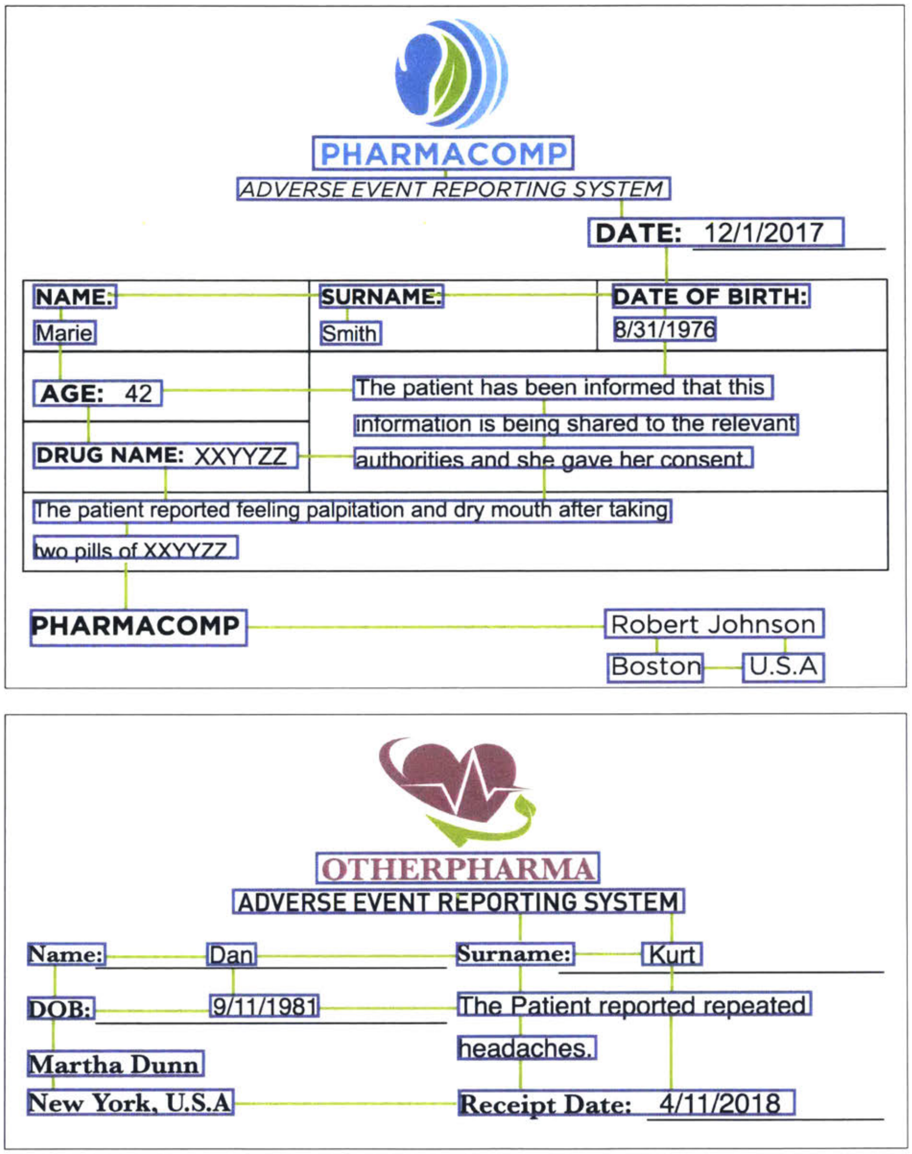

Figure 5-1: Mock-up example of Visual Information Extraction. The two forms have different layouts. Graphical dependencies are shown as green lines connecting text in blue bounding-boxes.

Model	Dev F1
GraphIE	77.8
– Edge types	77.0 (↓ 0.8)
– Horizontal edges	74.7 (↓ 3.1)
– Vertical edges	72.4 (↓ 5.4)
– CRF	72.1 (↓ 5.7)

Table 5.2: Ablation study on the AECR dataset. Scores are micro average F1 on the development set. "–" means removing the element from GraphIE.

5.2 Results

Table 5.1 shows the results in the visual information extraction task. GraphIE outperforms the SeqIE baseline in most attributes, and achieves 1.2% improvement in the mirco average F1 score. It confirms that the benefits of using layout graph structure in visual information extraction.

The extraction performance varies across the attributes, ranging from 61.4% for *Drug Name* to 95.8% for *Patient Birthday* (similar variations are visible in the baseline). Similarly, the gap between GraphIE and SeqIE varies in relation to the attributes, ranging between −0.5% in *Patient Birthday* and 2.4% in *Event.*

In the ablation test described in Table 5.2, we can see the contribution of: using separate weights for different edge types (+0.8%), horizontal edges (+3.1%), vertical edges (+5.4%), and CRF (+5.7%).

Generalization We also assess GraphIE's capacity of dealing with unseen layouts through an extra analysis. From our dataset, we sample 2,000 reports containing the three most frequent templates, and train the models on this subset. Then we test all models in two settings: 1) *seen templates*, consisting of 1,000 additional reports in the same templates used for training; and 2) *unseen templates*, consisting of 1,000 reports in two new template types.

The performance of GraphIE and SeqIE is reported in Figure 5-2. Both models achieve good results on *seen templates*, with GraphIE still scoring 2.8% higher than SeqIE. The gap becomes even larger when our model and the sequential one are

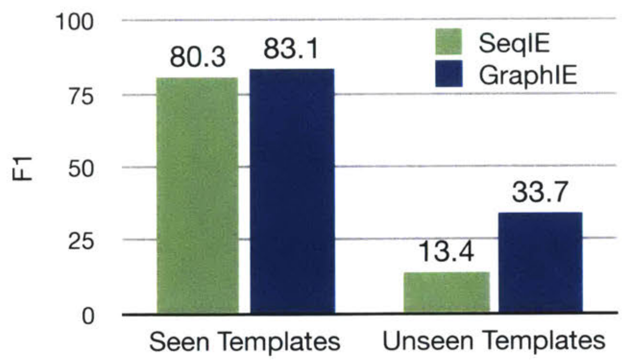

Figure 5-2: Micro average F1 scores tested on *seen* and *unseen* templates.

tested on *unseen templates* (i.e. 20.3%), demonstrating that by explicitly modeling the richer structural relations, GraphIE achieves better generalizability.

Chapter 6

Conclusions

In this thesis, we introduced GraphIE — an information extraction framework that learns local and non-local contextual representations from graph structures to improve predictions. The system operates over a task-specific graph topology describing the underlying structure of the input data. GraphIE jointly models the node (i.e. textual units, namely words or sentences) representations and their dependencies. Graph convolutions project information through neighboring nodes to finally support the decoder during tagging at the word level.

We evaluated our framework on three IE tasks, namely textual, social media and visual information extraction. Results show that it efficiently models non-local and non-sequential context, consistently enhancing accuracy and outperforming the competitive SeqIE baseline (i.e. BiLSTM+CRF).

Appendix A

Implementation Details

The models are trained with `Adam` [8] to minimize the CRF objective. For regularization, we choose dropout with a ratio of 0.1 on both the input word representation and the hidden layer of the decoder. The learning rate is set to 0.001. We use the development set for early-stopping and the selection of the best performing hyperparameters. For CharCNN, we use 64-dimensional character embeddings and 64 filters of width 2 to 4 [7]. The 100-dimensional pretrained GloVe word embeddings [22] are used in textual and social media IE, and 64-dimensional randomly initialized word embeddings are used in visual IE. We use a two-layer GCN in textual IE, and a one-layer GCN in social media and visual IE. The encoder and decoder BiLSTMs have the same dimension as the graph convolution layer. In visual IE, we concatenate a positional encoding to each text box's representation by transforming its bounding box coordinates to a vector of length 32, and then applying a tanh activation.

Bibliography

[1] Yonatan Aumann, Ronen Feldman, Yair Liberzon, Benjamin Rosenfeld, and Jonathan Schler. Visual information extraction. *Knowl. Inf. Syst.*, 10(1):1–15, July 2006.

[2] Edward Benson, Aria Haghighi, and Regina Barzilay. Event discovery in social media feeds. In *Proceedings of ACL*, pages 389–398. ACL, 2011.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL*, pages 363–370. ACL, 2005.

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[6] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. In *Proceedings of ACL*, pages 2410–2420, 2016.

[7] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In *Proceedings of AAAI*, pages 2741–2749. AAAI Press, 2016.

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[10] Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julen Oyarzabal, and Alfonso Valencia. Chemdner: The drugs and chemical names extraction challenge. *Journal of cheminformatics*, 7(1):S1, 2015.

[11] John D Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*, pages 282–289, 2001.

[12] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270, San Diego, California, June 2016. ACL.

[13] Jure Leskovec and Julian J Mcauley. Learning to discover social circles in ego networks. In *NIPS*, pages 539–547, 2012.

[14] Jiwei Li, Alan Ritter, and Eduard Hovy. Weakly supervised user profile extraction from twitter. In *Proceedings of ACL*, volume 1, pages 165–174, 2014.

[15] Qi Li, Heng Ji, and Liang Huang. Joint event extraction via structured prediction with global features. In *Proceedings of ACL*, volume 1, pages 73–82, 2013.

[16] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of ACL*, pages 1064–1074, Berlin, Germany, August 2016. ACL.

[17] Gideon S Mann and Andrew McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of Machine Learning Research*, 11(Feb):955–984, 2010.

[18] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL*, pages 1003–1011. ACL, 2009.

[19] Alan Mislove, Bimal Viswanath, Krishna P Gummadi, and Peter Druschel. You are who you know: inferring user profiles in online social networks. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining*, pages 251–260. ACM, 2010.

[20] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*, 2016.

[21] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *TACL*, 5:101–115, 2017.

[22] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of EMNLP*, pages 1532–1543, 2014.

[23] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, volume 1, pages 2227–2237, 2018.

[24] Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. GraphIE: A graph-based framework for information extraction. In *Proceedings of NAACL-HLT*, 2019.

[25] Chris Quirk and Hoifung Poon. Distant supervision for relation extraction beyond the sentence boundary. In *Proceedings of ACL*, volume 1, pages 1171–1182, 2017.

[26] Roi Reichart and Regina Barzilay. Multi event extraction guided by global constraints. In *Proceedings of NAACL-HLT*, pages 70–79. ACL, 2012.

[27] Angus Roberts, Robert Gaizauskas, and Mark Hepple. Extracting clinical relationships from patient narratives. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 10–18. ACL, 2008.

[28] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[29] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. N-ary relation extraction using graph-state lstm. In *Proceedings of EMNLP*, pages 2226–2235, 2018.

[30] Kumutha Swampillai and Mark Stevenson. Extracting relations within and across sentences. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 25–32, 2011.

[31] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*, 2015.

[32] Kim Sang Tjong, F Erik, and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of NAACL-HLT*, pages 142–147. ACL, 2003.

[33] Zhixiu Ye and Zhen-Hua Ling. Hybrid semi-markov crf for neural sequence labeling. In *Proceedings of ACL*, pages 235–240, 2018.

[34] Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of EMNLP*, 2018.