EVALUATING A JOINT NEURAL MODEL WITH GLOBAL FEATURES FOR DOCUMENT-LEVEL END-TO-END INFORMATION EXTRACTION

by

HAORAN WANG

A THESIS

Presented to the Department of Computer and Information Science
and the Division of Graduate Studies of the University of Oregon
in partial fulfillment of the requirements
for the degree of
Master of Science

June 2021

THESIS APPROVAL PAGE

Student: Haoran Wang

Title: Evaluating a Joint Neural Model with Global Features for Document-Level End-to-End Information Extraction

This thesis has been accepted and approved in partial fulfillment of the requirements for the Master of Science degree in the Department of Computer and Information Science by:

Thien Nguyen	Chair

and

Andy Karduna	Interim Vice Provost for Graduate Studies

Original approval signatures are on file with the University of Oregon Division of Graduate Studies.

Degree awarded June 2021

THESIS ABSTRACT

Haoran Wang

Master of Science

Department of Computer and Information Science

June 2021

Title: Evaluating a Joint Neural Model with Global Features for Document-Level End-to-End Information Extraction

Information Extraction (IE) is one of the most important fields in Natural Language Processing (NLP). The goal for IE tasks is to extract structured knowledge from unstructured text. While most datasets focus on sentence-level IE and paragraph-level IE, a document-level IE dataset is needed for research on processing long documents. Fortunately, researchers at Allen Institute for AI published a comprehensive and challenging document-level IE dataset (SCIREX) for the IE research community to study. Performing end-to-end IE tasks on SCIREX requires global understanding of the full document as relations can span across beyond sentences or even sections. This thesis applies a joint neural model with global features (ONEIE) to perform two end-to-end IE tasks on SCIREX, named entity extraction (NER) and relation extraction (RE). The performance of ONEIE is compared to SCIREX baseline model and DYGIE++, the state-of-the-art end-to-end IE model.

CURRICULUM VITAE

NAME OF AUTHOR: Haoran Wang

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, OR, USA
Purdue University, West Lafayette, IN, USA

DEGREES AWARDED:

Master of Science, Computer Science, 2021, University of Oregon
Bachelor of Science, Computer Science, 2019, Purdue University

AREAS OF SPECIAL INTEREST:

Artificial Intelligence
Machine Learning
Deep Learning
Natural Language Processing
Software Engineering

PROFESSIONAL EXPERIENCE:

Graduate Teaching Fellow, University of Oregon, 2020-2021

GRANTS, AWARDS AND HONORS:

Benjamin Fellowship, Montana State University, 2021

PUBLICATIONS:

ACKNOWLEDGEMENTS

My sincerest gratitude to professor Thien Nguyen, who introduced me to the field of Natural Language Processing and came up with the idea for this thesis. I would also like to thank all my friends and faculties at the Department of Computer and Information Science who gave me great support and guidence in the last two years. Finally, I would like to thank my family for their support through my schooling. They have always provided me with a wealth of encouragement.

TABLE OF CONTENTS

Chapter Page

I. INTRODUCTION 1

Recent Trends for IE Models 1

Recent Trends for IE Datasets 3

II. BACKGROUND 5

Information Extraction Tasks 5

Dynamic Graph IE (DYGIE) 6

DYGIE++ 8

SCIREX Model 9

Ⅲ. MODEL 11

Ⅳ. DATA 14

Data Creation 14

Dataset Breakdown.. 16

V. EXPERIMENTS... 18

Experiment Overview.. 18

Baselines 18

Global Features 19

Performance 20

Analysis...21

Chapter

Page

Ⅵ. FUTURE WORK 23

VII. CONCLUSION 25

REFERENCES CITED 26

LIST OF FIGURES

Figure Page

1. Overview of general Information Extraction (IE) pipeline that includes pre-processing, Named Entity Recognition, Relation Extraction, Coreference Resolution, and Named Entity Linking 5

2. Overview of DYGIE model. Span representations are refined by using broader context from the propagation of neighboring relation types and co-referred entities in the graph. This figure is copied from the original paper Luan et al. (2019) 7

3. SCIREX model overview. It uses BERT+BiLSTM for embeddings, a CRF layer to identify mentions, and a final classification layer for relation extraction. This figure is copied from the original paper Jain, van Zuylen, Hajishirzi and Beltagy (2020) 9

4. ONEIE model performs end-to-end IE in four stages: encoding, identification, classification, and decoding. This figure is copied from the original paper Lin, Ji, Huang and Wu (2020)...11

5. Examples of inter-dependencies between elements. This figure is copied from the original paper Lin et al. (2020) 13

6. ONEIE decoding algorithm. At each step, each candidate graph is expanded by adding a new node and possible edges between it and existing nodes. Then all expanded graphs are ranked and the top one is kept. This figure is copied from the original paper Lin et al. (2020) 13

7. Example showing annotations for named entities (Dataset, Metric, Task, Method), coreferences are indicated by arrows. This figure is from the original paper. Jain et al. (2020)................................15

LIST OF TABLES

Table Page

1. Named Entity Types and Relations in SCIREX 15
2. Statistics of SCIREX. Add dataset statistics are per-document averages. The statistics are provided by the original paper Jain et al. (2020) 16
3. Confusion Matrix for the mention-level corrections (change type, add span, or delete span). Values are average percentages per document, not per type. The statistics are provided by the original paper Jain et al. (2020)....................................... 17
4. ONEIE input format and its description. 18
5. Hyperparameters for our best performing ONEIE model 19
6. Global feature categories of ONEIELin et al. (2020) that can be used on SCIREXJain et al. (2020).. 20
7. Performance of ONEIE without global features 21
8. Performance of ONEIE with global features 21
9. Evaluating the F-1 score for NER and Relation Extraction for baselines and ONEIE with global features.. 21

CHAPTER I

INTRODUCTION

In today’ s digital age, there is an enormous amount of information that need to be processed daily in the form of news, emails, documents, social media, etc. Most of this information is unstructured, making it hard to reason about and interpret it. Therefore, there is a need for research that can deliver an efficient and sophisticated tool to automatically handle information given text inputs. Natural Language Processing (NLP) refers the study of extracting structured information from unstructured text as Information Extraction (IE).

Although existing joint neural models have achieved good results compared to the pipelined models, most of these models use local task-specific classifiers to predict individual IE tasks regardless of their interactions. This results in the failure of capturing cross-subtask and cross-instance inter-dependencies among local predictors. This thesis evaluates the performance of using a joint neural model with global features (ONEIE) Lin et al. (2020) for end-to-end information extraction tasks on a challenging document-level dataset (SCIREX) Jain et al. (2020). In order to understand the importance of ONEIE and SCIREX in IE field, a brief introduction of the recent trends for IE models and datasets is necessary.

Recent Trends for IE Models

With the recent advancements in deep learning, IE has moved from machine learning based systems to deep learning based neural models. However, these approaches typically perform IE in a pipelined fashion, which leads to error propagation and does not allow interactions among the components in the pipeline.

Some earlier approaches involved using joint inference and joint modeling methods to improve local prediction. Roth and Yih Roth and Yih (2004) developed a linear programming formulation to simultaneously learn named entities and relations. They modeled inference as an optimization problem, and converted it to a linear program. This allowed the authors to solve very large linear programming problems in a short amount of time. Other works that jointly perform IE tasks, including Markov Logic NetworksRiedel, Chun, Takagi and Tsujii (2009), Structured PerceptronLi, Ji and Huang (2013), and Graphical ModelsYang and Mitchell (2016).

More recently, due to the success of deep learning, Luan et al Luan et al. (2019) created a global inference model by designing neural networks with embedding features to jointly model multiple sub tasks. This general model called Dynamic Graph IE (DYGIE), uses dynamically constructed span graphs to represent relations and coreferences. It achieved state-of-the-art performance on multiple datasets. However, like previous approaches, it still uses separate local task-specific classifiers in the final layer. This results in the failure of capturing inter-dependencies among tasks and instances.

To address the issue mentioned above, Lin et al. Lin et al. (2020) developed a new joint neural model called ONEIE. Instead of predicting separate knowledge elements using local classifiers, ONEIE extracts a globally optimal information network for the input. During the decoding process, ONEIE not only considers individual label scores for each knowledge element, but also evaluates cross-subtask and cross-instance interactions in the network. This model achieved performance better than or comparable to the previous state-of-the-art approach on ACE05Sanh, Wolf and Ruder (2018), a benchmark dataset for IE.

Recent Trends for IE Datasets

Conventional datasets for IE focus on within-sentence relations. However, researchers recently started working on datasets for short paragraphs, such as abstracts of scientific articles. SCIERC Luan, He, Ostendorf and Hajishirzi (2018) is a dataset of 500 richly annotated scientific abstracts including annotations for scientific entities, relations, and coreference clusters. These abstracts are taken from 12 AI conference/workshop proceedings. Although this dataset has brought new challenges for IE on paragraph-level, there is still lack of comprehensive IE datasets annotated at the document level for researchers to study.

A newly released dataset SCIREX Jain et al. (2020) has solved this problem and brought new challenges for document-level IE research. To overcome the annotation challenges, which requires annotators to have proper domain knowledge and identifying relations that span across the whole document, the researchers performed both automatic and manual annotations. The end result is a dataset of 438 fully annotated documents that contains annotations for entities, salient entities, N-ary relations and coreferences. Since this is the first document-level dataset for IE, it poses multiple challenges, including aggregating coreference information from across documents in an end-to-end manner, identifying salient entities and perform N-ary relation extraction of those entities.

With this thesis, we aim to solve the specific challenges that SCIREX proposes by training ONEIE model on SCIREX and study the performance benefits of using an end-to-end neural model with global features. We focus on two IE tasks, entity extraction and relation extraction. We measure the performance by the model’s F-1 score on the test set. The primary goal of this thesis is to understand ONEIE’s performance on document-level IE tasks. We do this by comparing ONEIE’ s performance with DYGIE++’ s performance on SCIREX, a benchmark model for end-to-end information extraction tasks.

CHAPTER II

BACKGROUND

Information Extraction Tasks

Figure 1 shows a general IE pipeline, which involves various tasks such as Named Entity Recognition (NER), Relation Extraction(RE), Named Entity Linking (NEL), Coreference Resolution (CR), etc. Some low-level IE tasks such as NER are the fundamental building blocks of complex NLP tasks such as Knowledge Graph Construction, Question-Answering, and so on. In this thesis, we focus on two IE tasks: named entity extraction and relation extraction.

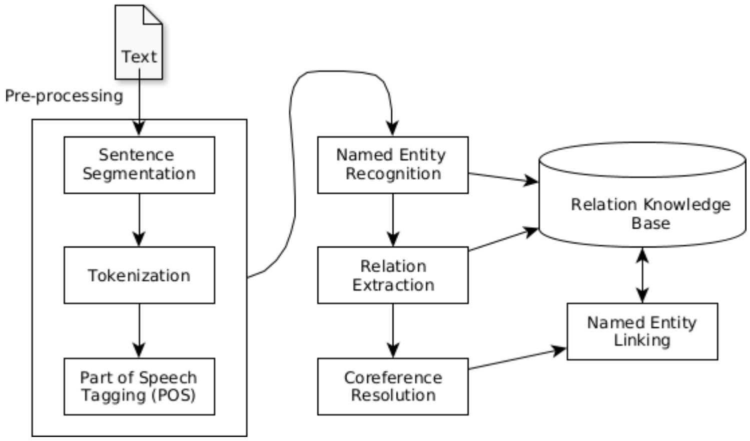


Figure 1. Overview of general Information Extraction (IE) pipeline that includes pre-processing, Named Entity Recognition, Relation Extraction, Coreference Resolution, and Named Entity Linking

- Named Entity Recognition (NER): The goal of this task is to recognize Named Entities that occur in the text. These Named Entities include Person (PER), Location (LOC), and Geo-Political Entities (GPE), etc. For instance, in the statement ” Guiliani, 58, proposed to Nathan, a former nurse, during a business trip to Paris” , NER extracts Guiliani and Nathan which refers to person and Paris to location.

- Relation Extraction (RE): This task detects and classifies pre-defined relationships between entities identified in the text. It transforms unstructured text into structured form which can be used in search engine, question answering, etc. For instance, given the statement " In interviews last year, Guiliani said Nathan gave him 'tremendous emotion support' through his treatment for prostate cancer as he led New York City during the Sept.11,2001 terror attacks." Pre-defined relations can be in the form of Leader-Of that holds between a PER and LOC. In this case, RE extracts the relation that Guiliani is the leader of New York City.

Dynamic Graph IE (DYGIE)

Dynamic Graph IE (DYGIE) Luan et al. (2019) uses dynamically constructed graphs to perform multiple IE tasks that capture contextualized entities, relations and coreferences. DYGIE achieved this by constructing dynamic graphs with refined span representations, as illustrated in Figure 2. The nodes in the dynamic graph are dynamically selected from a beam of highly-confident mentions, and the edges are weighted according to the confidence scores of relation types or coreferences. Unlike previous joint approaches that only rely on the first layer LSTM (Long Short-Term Memory)Hochreiter and Schmidhuber (1997) to share span representations between various tasks, the dynamic graph in DYGIE allows coreferences and relation type confidence to repeatedly refine the span representations by selecting the most confident entity spans.

DYGIE model has five layers, a token representation layer, a span representation layer, a coreference propagation layer, a relation representation layer, and a final prediction layer. The token representation layer uses a bidirectional LSTM to obtain word embeddings by stacking the forward and backward LSTM

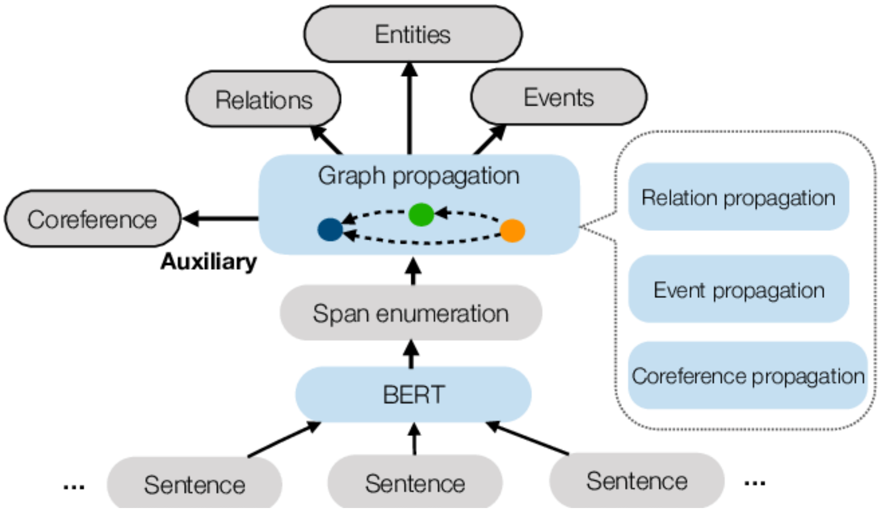


Figure 2. Overview of DYGIE model. Span representations are refined by using broader context from the propagation of neighboring relation types and co-referred entities in the graph. This figure is copied from the original paper Luan et al. (2019)

hidden states. After the token representation layer, the vectors are sent into a span representation layer to enumerate all text spans in each sentence and compute a locally-contextualized vector space representation of each span. In the coreference propagation layer and relation propagation layer, DYGIE employs a dynamic span graph to embed global information into its span representation identifying the text spans that are most likely to represent entities. Those spans are treated as nodes in the graph. Then, a confidence-weighted arc is constructed for each node based on its predicted coreference and relation links with other nodes in the graph. Then, DYGIE refines the span representations by propagating the coreference and relation type confidences through the graph. In the final prediction layer, these refined span representations are used to predict entity types, relation types, and coreference links in a multi-task fashion.

The key contribution of DYGIE is its dynamic graph approach, it incorporates the interactions across tasks that allows the model to learn information from broader context. It achieved significant improvement across different IE tasks including entity, relation extraction over the previous state-of-the-art model on ACE05Sanh et al. (2018) dataset.

DYGIE++

DYGIE++Wadden, Wennberg, Luan and Hajishirzi (2019) is an improved version of the original DYGIE modelLuan et al. (2019). DYGIE++ uses BERTDevlin, Chang, Lee and Toutanova (2018) as contextualized word embeddings rather than Bi-LSTM to capture relationships among entities in the same or adjacent sentences, while dynamic span graph updates model long-range cross-sentence relationships. This allows DYGIE++ to capture both local (within-sentence) and global (cross-sentence) context.

DYGIE++ uses a ” sliding window” approach for BERT encoding, each sentence is fed to BERT together with a ***size-L*** neighborhood of surrounding sentences. Wadden et alWadden et al. (2019) found that by increasing the input window size, BERT encodings are able to capture important within and adjacent-sentence context, which improves the performance on all tasks. They also found contextual encoding through message passing updates enables the model to incorporate cross-sentence dependencies, which improves performance on IE tasks in specialized domains.

DYGIE++ achieved state-of-the-art results for named entity recognition, relation extraction and event extraction tasks on four different benchmark datasets: ACE05 (coreference annotations from OntoNotesPradhan, Moschitti, Xue, Uryupina and Zhang (2012)), SciERCLuan et al. (2018), GENIAOhta, Kim, Pyysalo, Wang and Tsujii (2009), and WLPCKulkarni, Xu, Ritter and Machiraju (2018).

SCIREX Model

Jain et al Jain et al. (2020) developed a neural model that performs document-level IE tasks jointly in an end-to-end fashion. As shown in Figure 3, this model has an embedding layer, an identification layer, and a classification layer.

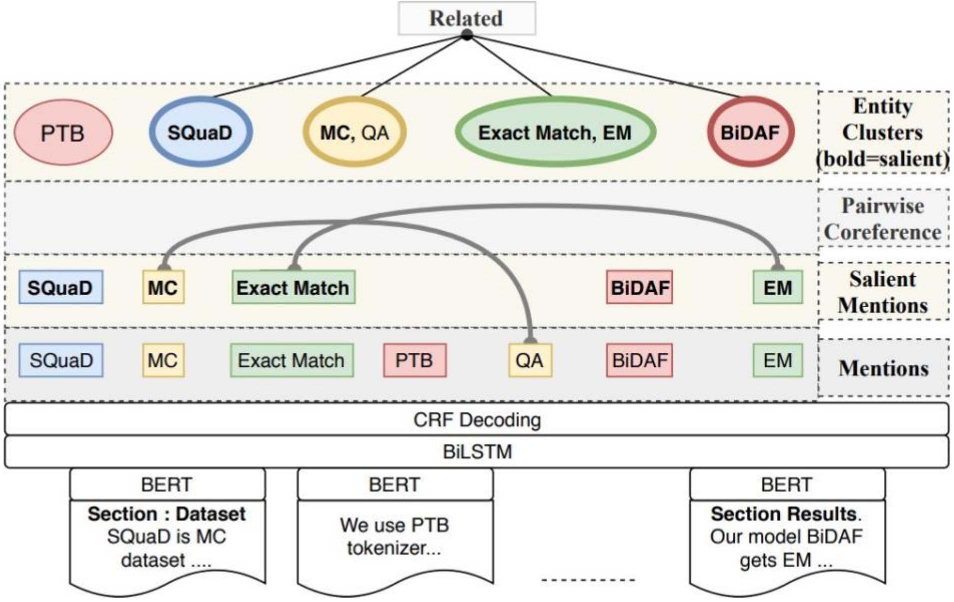


Figure 3. SCIREX model overview. It uses BERT+BiLSTM for embeddings, a CRF layer to identify mentions, and a final classification layer for relation extraction. This figure is copied from the original paper Jain et al. (2020)

The embedding layer uses a pre-trained SciBERTBeltagy, Cohan and Lo (2019) to get contextualized word embeddings for each section. Then section-level word embeddings are concatenated and a Bi-LSTMSchuster and Paliwal (1997) is added on top of them. This allows the model to take into account cross-section dependencies.

A conditional random field (CRF)Sutton and McCallum (2010) sequence tagger is trained to identify mentions and classify their types. After the mentions have been identified, they are clustered into binary and 4-tuple clusters based on if they are expressed or not in the document. Each relation is encoded into a single vector by constructing a section embedding and aggregating them to generate a document level embedding. Then, the document level embedding is passed through a FFN for relation classification.

CHAPTER III

MODEL

ONEIELin et al. (2020) is a joint neural framework that extracts globally optimal IE results as a graph in four stages as shown in Figure 3. First, during the encoding stage, it encodes the input sentence using a pre-trained BERT encoder Devlin et al. (2018). Second, during the identification stage, it identifies entity mentions and event triggers in the sentence. Then, during the classification stage, it computes the type label scores for all nodes and pairwise edges among them. Finally, during the decoding stage, it discovers possible information networks using beam search and returns the one with the highest global score.

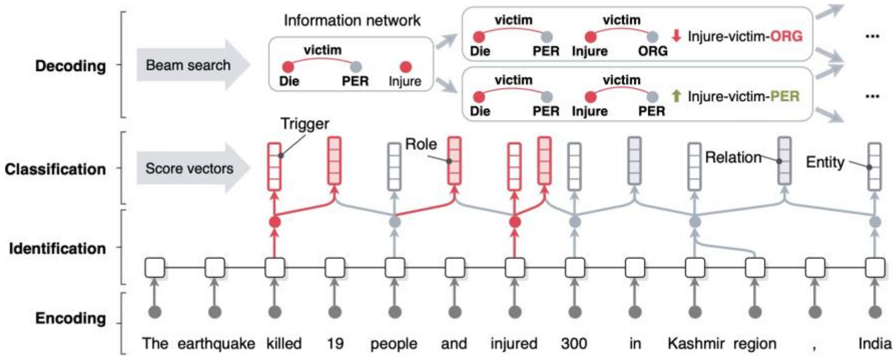


Figure 4. ONEIE model performs end-to-end IE in four stages: encoding, identification, classification, and decoding. This figure is copied from the original paper Lin et al. (2020)

BERT Devlin et al. (2018) is a transformer based model that uses an attention mechanism Vaswani et al. (2017) to learn contextual relations between words in a sentence. BERT embeddings perform significantly better than non-contextualized embeddings like word2vecMikolov, Sutskever, Chen, Corrado and Dean (2013) or GloVe Pennington, Socher and Manning (2014). By using a pre-trained BERT encoder, ONEIE can capture the context for input sentences. While previous methods typically use the last layer of BERT, Lin et al Lin et al.

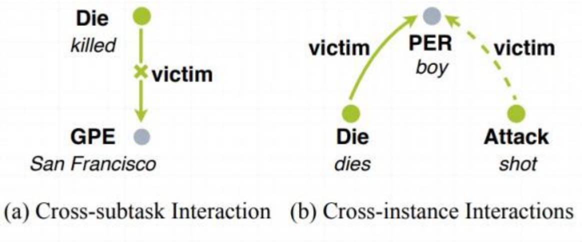
(2020) found that using the output of the third to last layer of BERT substantially improved the performance of ONEIE on most tasks.

After the encoding stage, the vectors are sent into a feed-forward neural network to compute a score vector for each word. Each value in the vector represents the score for a tag in the BIO target tag set. Then, a conditional random field (CRF) is used to capture the dependencies between predicted tags. Finally, a tag path is calculated and trained by maximizing the log-likelihood of the gold-standard tag path during the training. ONEIE uses separate taggers to extract entity mentions and event triggers, so that it can make a joint decision for all knowledge elements at the decoding stage to prevent error propagation.

During the classification stage, each identified node is represented by averaging its word representations. Then, separate task-specific feed-forward neural networks are used to calculate label scores for each node. To obtain the label vectors for the edges, ONEIE concatenates the span representations of the nodes to obtain a vector, and calculates its label score. Finally, the model is trained by minimizing the cross-entropy loss between label vectors and the true label vectors. If global features are not considered, a locally best graph can be generated by simply predicting the label with the highest score for each knowledge element.

However, one limitation of the local classifiers is that they cannot capture the inter-dependencies between knowledge elements in an information network. This could result in local classifiers predicting contradictory results or failing to predict difficult edges that require information from other elements. These inter-dependencies (cross-subtask interactions and cross-instance interactions) require global features to provide context. ONEIE takes in a template of user-defined global features and learns the weight for each feature during training.

Then, a global score is calculated by summing up the local feature score and global feature score. ONEIE makes the assumption that the gold-standard graph for each sentence should achieve the highest global score. Therefore, it minimizes the loss between the graph predicted by local classifiers and the gold-standard graph during training.



(a) Cross-subtask Interaction (b) Cross-instance Interactions

Figure 5. Examples of inter-dependencies between elements. This figure is copied from the original paper Lin et al. (2020)

In the final decoding stage, ONEIE makes a joint decision for all nodes and their pairwise edges to obtain the globally optimal graph. This is achieved by calculating the global score for each candidate graph and selecting the best one using a beam search based decoder.

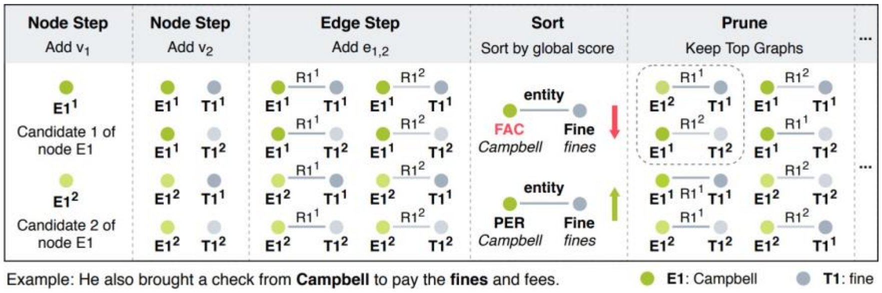


Figure 6. ONEIE decoding algorithm. At each step, each candidate graph is expanded by adding a new node and possible edges between it and existing nodes. Then all expanded graphs are ranked and the top one is kept. This figure is copied from the original paper Lin et al. (2020)

CHAPTER IV
DATA

SCIREX Jain et al. (2020) is currently the largest document-level IE dataset. Before SCIREX, there was a lack of comprehensive IE datasets annotated at the document level. Recent work by Hou et al Hou, Jochim, Gleize, Bonin and Ganguly (2019) and Jie et al Jie and Lu (2019) built datasets for document-level relation extraction by using distant supervision annotations. Both datasets formulate the relation extraction task as a binary classification to check whether a triplet of ground-truth entities is expressed in the document. SCIREX, on the other hand, focuses on relation extraction in addition to a comprehensive list of IE tasks.

Data Creation

Building a large-scale document-level IE dataset is challenging as it requires a global understanding of the document-level relations that span beyond sentences or even paragraphs. To address this issue, Jain et al Jain et al. (2020) developed a method to build SCIREX dataset with little annotation effort. This method combines distant supervision from an existing knowledge base (KB) and noisy automatic labeling to provide a simpler annotation task. The supervision is distant because PwC (Paper with Code) does not provide where exactly the result tuple is mentioned in the article.

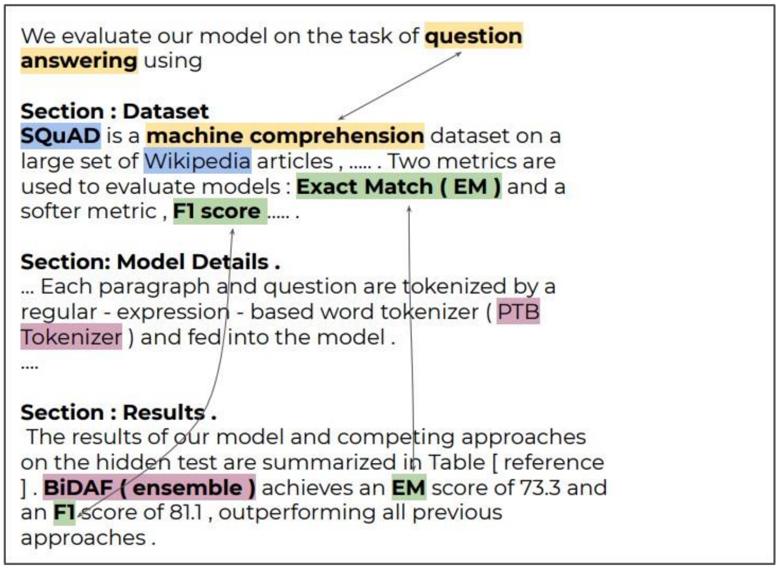
For pre-processing, Jain et alJain et al. (2020) use Papers with Code (PwC) dataset [1] as the knowledge base. PwC is a corpus of 1,170 articles published in machine learning (ML) conferences with result tuple annotations for Dataset, Metric, Method, Task and Score, as shown in Table 1. Then, they extract clean document text with no figures/tables/equations from the PDF files of the papers in

[1] https://github.com/paperswithcode/paperswithcode-data

PwC dataset. For the annotation process, they simplify the human annotation task by automatically labeling the data with noisy labels, then an expert annotator only needs to fix the labeling mistakes.

Named Entity	Example
Method	BiDAF(ensemble)
Metric	F1 score
Task	question answering
Material	SQuAD

Table 1. Named Entity Types and Relations in SCIREX



We evaluate our model on the task of **question answering** using

Section : Dataset
SQuAD is a **machine comprehension** dataset on a large set of Wikipedia articles , ... Two metrics are used to evaluate models : **Exact Match (EM)** and a softer metric , **F1 score** ..... .

Section: Model Details .
... Each paragraph and question are tokenized by a regular - expression - based word tokenizer (PTB Tokenizer) and fed into the model .
....

Section : Results .
The results of our model and competing approaches on the hidden test are summarized in Table [reference] . **BiDAF (ensemble)** achieves an **EM** score of 73.3 and an **F1** score of 81.1 , outperforming all previous approaches .

Figure 7. Example showing annotations for named entities (Dataset, Metric, Task, Method), coreferences are indicated by arrows. This figure is from the original paper. Jain et al. (2020)

Jain et alJain et al. (2020) achieve automatic labeling by training a standard BERT+CRF sequence labeling model on the SCIERC dataset Luan et al. (2018). This trained model provides automatic but noisy predictions for mention span identification. To determine which predictions are noisy, they compute the Jaccard similarity between each mention predicted by the model and each of the PwC entities. Each mention is linked to the entity if the threshold exceeds a certain $\boldsymbol{E}$. To determine $\boldsymbol{E}$, two expert annotators manually went through 10 documents to mark identified mentions with entity names, and chose the $\boldsymbol{E}$ that maximize the probability. After identifying the noisily labeled data, annotators perform necessary corrections to generate high-quality annotations. Table 3 shows the confusion matrix for automatic labeling.

Dataset Breakdown

The result is a dataset of 438 fully annotated documents. Table 2 provides the dataset statistics. Jain et al found that the majority of the relations in SCIREX, especially 4-ary relations span multiple sentences or even multiple sections in the document: 57% of binary and 99% of 4-ary relations occur across sentences; 20% binary and 55% 4-ary relations occur across sections. These cross-sentence and cross-section relations highlight the need for document level IE models.

Statistics (avg per doc)	SCIREX
Words	5,737
Sections	22
Mentions	360
Binary Relations	16
4-ary Relations	5

Table 2. Statistics of SCIREX. Add dataset statistics are per-document averages. The statistics are provided by the original paper Jain et al. (2020)

SCIERX has four named entity types: Method, Task, Metric and Dataset (named as Material in the json files). SCIREX has two relation types: N-ary Relations and Method Subrelations. The N-ary relation includes binary, 3-ary, and 4-ary relations between a collection of entities of named type (Method, Task, Metric and Dataset). The 4-ary relation cannot be split into multiple binary

	Dataset	Metric	Task	Method	Deleted
Dataset	3.55	0.01	0.07	0.16	0.03
Metric	0.02	7.95	0.00	0.03	0.00
Task	0.32	0.07	17.92	0.44	0.01
Method	0.65	0.21	0.24	53.27	0.02
Added	2.40	1.30	2.82	8.50	-

Table 3. Confusion Matrix for the mention-level corrections (change type, add span, or delete span). Values are average percentages per document, not per type. The statistics are provided by the original paper Jain et al. (2020)

relations because a dataset might have multiple tasks. Each task may have its own metric, so the metric cannot be decided solely based on the dataset or the task. The Method Subrelations annotate methods that may be subdivided into simpler submethods. For example, ” DLDL+VGG-Face” is broken into two methods ” DLDL” , ” VGG-Face” .

CHAPTER V

EXPERIMENTS

Experiment Overview

ONEIE takes specific input format, as shown in Table 4. We preprocessed SCIERX’ s input to match ONEIE’ s input format.

Name	Description
doc_id	document id
sent_id	sentence id
entity_mentions	list of entities and their mentions
relation_mentions	list of relations and their mentions
tokens	list of tokens (words)
token_lens	list of token lens for each token
sentence	untokenized text input

Table 4. ONEIE input format and its description.

We use bert-large-cased as the BERT model, and AdamW Loshchilov and Hutter (2017) as the optimizer. Both learning rate and weight decay for BERT are set to 1e-5. For local classifiers, we use two-layer FFNs with a dropout rate of 0.4. We use 150 hidden units for entity and relation extraction. For global features, we set β_v and β_e to 2 and set θ to 10. The standard Precision, Recall and F-1 score are used to evaluate the performance across all tasks. After tuning the hyperparameters, we found that ONEIE performs better with small batch size and trained for a large number of epochs. Table 5 shows the configuration of our best performed ONEIE model.

Baselines

We use the following models as our baselines on SCIREX dataset.

Hyperparameter	**Value**
batch size	10
evaluation batch size	10
max epoch	60
learning rate	1e-3
weight decay	1e-3

Table 5. Hyperparameters for our best performing ONEIE model

- **DYGIE++ Wadden et al. (2019)** The state-of-the-art end-to-end IE model that utilizes multi-sentence BERT encodings and span graph propagation to predict entity types, relation types in a multi-task fashion.
- **SCIREX Model Jain et al. (2020)** An end-to-end neural model that uses a two-level BERT+BiLSTM method for token representation, a CRF layer to identify mentions, and a final classification layer to predict relations.

For DYGIE++, being a span enumeration type model, it only works on paragraph level and extracts relations between mentions in the same sentence. Therefore, it cannot be trained directly on SCIREX dataset. Jain et alJain et al. (2020) subdivided SCIREX documents into sections and formulate each section as a single training example. They map each binary mention-level relation returned to entity-level by mapping the span to its gold cluster label if it appears in one. We use this special training dataset for DYGIE++.

Global Features

Table 6 shows the global feature template categories of ONEIE. These templates can capture cross-subtask and cross-instance interactions. The model fills in all possible types to generate features while learning the weight of each feature during training.

Given a graph G, we represent its global feature vector as f_G = $f_1(G), ..., f_M(G)$, where M is the number of global features and $f_i(.)$ is a function that evaluates a certain feature and returns a scalar. For example, $f_i(G)$ returns a scalar from 0 to 1 to represent the number of occurrence of a certain entity type and relation type combination.

Next, ONEIE learns a weight vector u and calculates the global feature score of G as the dot product of f_G and u. The global score of G is calculated as the sum of its local score and global feature score, $s(G) = s^l(G) + uf_G$. During training, we minimize the following loss function, $L^G = s(\hat{G}) - s(G)$, where $hatG$ is the graph predicted by local classifiers and G is the gold standard graph. Finally, we optimize the following joint objective function during training $L = L^I + {}_{t \in T} L^t + L^G$.

Category	Description
6	The number of occurrences of $< entity\_type_i >$, $< entity\_type_j >$, and $< relation\_type_j >$ combination.
7	The number of occurrences of $< entity\_type_i >$ and $< relation\_type_j >$ combination.
9	The number of entities that have a $< relation\_type_i >$ relation with multiple entities.
10	The number of entities involving in $< relation\_type_i >$ and $< relation\_type_j >$ relations simultaneously.

Table 6. Global feature categories of ONEIELin et al. (2020) that can be used on SCIREXJain et al. (2020)

We performed experiments to test ONEIE’ s performance on SCIREX both with and without global features.

Performance

Table 7 and Table 8 list the performance of our ONEIE model, both with and without global features. The performance is measured by precision, recall, and F-1 score.

Task	Precision	Recall	F1
NER	70.3	71.5	70.9
Relation	48.6	76.4	59.4

Table 7. Performance of ONEIE without global features

Model	Precision	Recall	F1
NER	70.8	71.9	71.3
Relation	47.8	79.3	59.6

Table 8. Performance of ONEIE with global features

Table 9 lists the performance comparison between ONEIE with global features and the baselines. The performance is measured by F-1 score.

Task	DYGIE++	SCIREX Mdoel	ONEIE
NER	67.8	71.2	**71.3**
Relation	**61.9**	61.1	59.6

Table 9. Evaluating the F-1 score for NER and Relation Extraction for baselines and ONEIE with global features

As shown in Table 9, ONEIE with global features outperform SCIREX model for NER task, and is close to the performance of DYGIE++ for relation extraction task.

Analysis

Our experiments showed that global feature improves the performance of ONEIE on both named entity recognition task and relation extraction task. Both tasks benefit from the document-level context provided by the global features.

While ONEIE with global features outperforms the baselines for NER, it does not outperform the baselines for relation extraction. We suspect that this is due to the potential error accumulation from identification stage to classification stage in ONEIE. The classification feed-forward neural network uses the tag path from identification stage to calculate the score vector for the edges between nodes in the classification stage, which could lead to potential error accumulation. SCIREX baseline model also suffers from this problem. Jain et al Jain et al. (2020) found that there is quite a drop in the end-to-end performance compared to the component-wise performance. It is particularly clear with relation extraction, even though the relation extraction component performance is reasonably good in isolation, its end-to-end performance is quite low because of the accumulation of errors in previous steps. Since DYGIE++ uses a dynamic graph to construct information network and then predict the best graph, there is less error accumulation through the layers. Therefore, it achieved the best performance on relation extraction across all three models.

CHAPTER VI
FUTURE WORK

Future work for this thesis involves solving end-to-end document-level IE using a more recent joint IE model. FourIEM. V. Nguyen, Lai and Nguyen (2021) solves four different IE tasks (entity mention extraction, relation extraction, event trigger detection, and argument extraction) simultaneously in a single model. Hence, it is called FourIE. Compared to prior joint IE models, FourIE features two novel contributions to capture inter-dependencies among tasks. First, at the representation level, it uses an interaction graph to enrich the prediction representation for instances of the four IE tasks. Second, at the label level, it uses a dependency graph for the information types in the four IE tasks that captures the connections between the types expressed in an input sentence. FourIE also uses a novel regularization mechanism to enforce the consistency between the golden and predicted type dependency graphs to improve representation learning.

FourIE has three major components: (i) Span Detection, (ii) Instance Interaction, and (iii) Type Dependency-based Regularization. Span Detection aims to identify spans of entity mentions and event triggers in input sentences that would be used to form the nodes in the interaction graph between different instances of the four IE tasks. The span detection problems are formulated as sequence labeling tasks where each word is associated with two BIO tags to capture the span information for entity mentions and event triggers. After the tag sequences are labeled from Span Detection component, two separate span sets are obtained for the entity mentions and event triggers in the sentence. Then, Instance Interaction component leverages the span representation vectors to form instance representations and enrich them with instance interactions to perform necessary predictions in IE. Finally, the Type Dependency-based Regularization component obtains the type dependencies across the tasks and use them to supervise the model in the training process to improve the representation vectors for IE.

We believe since ONEIE only computes predictive representation vectors for instances of the tasks independently, it fails to explicitly present the connections between related instances of different tasks and encode them into the representation learning process. FourIE, on the other hand, creates a graph structure to explicitly capture the interactions between related instances of the IE tasks in a sentence. Then, FourIE uses a graph constitutional network (GCN)T. Nguyen and Grishman (2018) to enrich the representation vector for an instance with those from the neighboring instances for IE. Therefore, FourIE could potentially achieve better performance on SCIREX.

CHAPTER VII
CONCLUSION

We evaluated how a joint neural model with global features performs on a document-level IE dataset. SCIREX, being a document-level IE dataset, requires an understanding of the full document to extract entities and relations. Therefore, we need a joint neural model, ONEIE to perform IE tasks that require cross-instance and cross-subtask inter-dependencies. ONEIE incorporates global features to capture the inter-dependency between knowledge elements. Experiments show that ONEIE with global features outperforms the SCIREX model for NER, and is close to the performance of DYGIE++ for relation extraction.

REFERENCES CITED

Beltagy, I., Cohan, A. & Lo, K. (2019). Scibert: Pretrained contextualized embeddings for scientific text. *CoRR*, *abs/1903.10676*. Retrieved from http://arxiv.org/abs/1903.10676

Devlin, J., Chang, M., Lee, K. & Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*. Retrieved from http://arxiv.org/abs/1810.04805

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

Hou, Y., Jochim, C., Gleize, M., Bonin, F. & Ganguly, D. (2019, July). Identification of tasks, datasets, evaluation metrics, and numeric scores for scientific leaderboards construction. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 5203 – 5213). Florence, Italy: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P19-1513 doi: 10.18653/v1/P19-1513

Jain, S., van Zuylen, M., Hajishirzi, H. & Beltagy, I. (2020, July). SciREX: A challenge dataset for document-level information extraction. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 7506 – 7516). Online: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/2020.acl-main.670 doi: 10 .18653/v1/2020.acl-main.670

Jie, Z. & Lu, W. (2019, November). Dependency-guided LSTM-CRF for named entity recognition. In ***Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*** (pp. 3862 – 3872). Hong Kong, China: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D19-1399 doi: 10.18653/v1/D19-1399

Kulkarni, C., Xu, W., Ritter, A. & Machiraju, R. (2018, June). An annotated corpus for machine reading of instructions in wet lab protocols. In ***Proceedings of the 2018 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 2 (short papers)*** (pp. 97 – 106). New Orleans, Louisiana: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/N18-2016 doi: 10.18653/v1/N18-2016

Li, Q., Ji, H. & Huang, L. (2013, August). Joint event extraction via structured prediction with global features. In ***Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)*** (pp. 73 – 82). Sofia, Bulgaria: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P13-1008

Lin, Y., Ji, H., Huang, F. & Wu, L. (2020, July). A joint neural model for information extraction with global features. In ***Proceedings of the 58th annual meeting of the association for computational linguistics*** (pp. 7999 – 8009). Online: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/2020.acl-main.713 doi: 10.18653/v1/2020.acl-main.713

Loshchilov, I. & Hutter, F. (2017). Fixing weight decay regularization in adam. ***CoRR*, *abs/1711.05101*** . Retrieved from http://arxiv.org/abs/1711.05101

Luan, Y., He, L., Ostendorf, M. & Hajishirzi, H. (2018, October-November). Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In ***Proceedings of the 2018 conference on empirical methods in natural language processing*** (pp. 3219 – 3232). Brussels, Belgium: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D18-1360 doi: 10.18653/v1/D18-1360

Luan, Y., Wadden, D., He, L., Shah, A., Ostendorf, M. & Hajishirzi, H. (2019, June). A general framework for information extraction using dynamic span graphs. In ***Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*** (pp. 3036 – 3046). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/N19-1308 doi: 10.18653/v1/N19-1308

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *CoRR*, *abs/1310.4546* . Retrieved from http://arxiv.org/abs/1310.4546

Nguyen, M. V., Lai, V. D. & Nguyen, T. H. (2021). Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. *CoRR*, *abs/2103.09330* . Retrieved from https://arxiv.org/abs/2103.09330

Nguyen, T. & Grishman, R. (2018, Apr.). Graph convolutional networks with argument-aware pooling for event detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32* (1). Retrieved from https://ojs.aaai.org/index.php/AAAI/article/view/12039

Ohta, T., Kim, J.-D., Pyysalo, S., Wang, Y. & Tsujii, J. (2009). Incorporating genetag-style annotation to genia corpus. In *Proceedings of the workshop on current trends in biomedical natural language processing* (p. 106 – 107). USA: Association for Computational Linguistics.

Pennington, J., Socher, R. & Manning, C. (2014, October). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532 – 1543). Doha, Qatar: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D14-1162 doi: 10.3115/v1/D14-1162

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O. & Zhang, Y. (2012, July). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In ***Joint conference on EMNLP and CoNLL - shared task*** (pp. 1 – 40). Jeju Island, Korea: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W12-4501

Riedel, S., Chun, H.-W., Takagi, T. & Tsujii, J. (2009, June). A Markov Logic approach to bio-molecular event extraction. In ***Proceedings of the BioNLP 2009 workshop companion volume for shared task*** (pp. 41 – 49). Boulder, Colorado: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W09-1406

Roth, D. & Yih, W.-t. (2004, May 6 - May 7). A linear programming formulation for global inference in natural language tasks. In ***Proceedings of the eighth conference on computational natural language learning (CoNLL-2004) at HLT-NAACL 2004*** (pp. 1 – 8). Boston, Massachusetts, USA: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/W04-2401

Sanh, V., Wolf, T. & Ruder, S. (2018). A hierarchical multi-task approach for learning embeddings from semantic tasks. ***CoRR***, ***abs/1811.06031*** . Retrieved from http://arxiv.org/abs/1811.06031

Schuster, M. & Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673-2681. doi: 10.1109/78.650093

Sutton, C. & McCallum, A. (2010). *An introduction to conditional random fields.*

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. *CoRR*, *abs/1706.03762* . Retrieved from http://arxiv.org/abs/1706.03762

Wadden, D., Wennberg, U., Luan, Y. & Hajishirzi, H. (2019). Entity, relation, and event extraction with contextualized span representations. *CoRR*, *abs/1909.03546* . Retrieved from http://arxiv.org/abs/1909.03546

Yang, B. & Mitchell, T. M. (2016, June). Joint extraction of events and entities within a document context. In *Proceedings of the 2016 conference of the north American chapter of the association for computational linguistics: Human language technologies* (pp. 289 – 299). San Diego, California: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/N16-1033 doi: 10.18653/v1/N16-1033