Multi-task graph-based information extraction with global context

Yi Luan

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2019

Reading Committee:

Mari Ostendorf, Chair

Hannaneh Hajishirzi, Chair

Luke Zettlemoyer

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Multi-task graph-based information extraction with global context

Yi Luan

Co-Chairs of the Supervisory Committee:
Professor Mari Ostendorf
Electrical and Computer Engineering

Assistant Professor Hannaneh Hajishirzi
Computer Science and Engineering

With growing numbers of written documents in the world, it is crucial to leverage automatic language processing so that people can make better use of the information. The main challenge stems from the fact that the information in written text is not as easily used as information in a structured database. Therefore it is very important to understand and automatically extract structured information from large amount of unstructured texts. To tackle this problem, Information Extraction (IE) is the widely studied task of retrieving structured information from text. In this thesis, our goal is to develop a general high performance IE system that can work across many different domains and tasks, but particularly the less well studied domain of scientific literature.

Towards achieving this goal, we propose a series of general IE frameworks that addresses the task of entity recognition, relation extraction and coreference resolution. This thesis research addresses challenges common to all such IE systems: 1) how to leverage large unannotated data when annotated training data are limited; and 2) how to model the interactions between different tasks so that the tasks can best benefit each other.

In this thesis, we first develop an efficient way of improving the performance of supervised neural systems through semi-supervised learning. We introduce a method of integrating a graph-based semi-supervised algorithm together with a confidence-based self-training scheme to leverage

unannotated articles. We also introduce two general IE frameworks, Span-based IE (SPANIE) and Dynamic Graph IE (DYGIE) for coupling multiple information extraction tasks through shared span representations. Our frameworks are effective for all three tasks, demonstrating a benefit from incorporating broader context learned from relation and coreference annotations. The DYGIE model achieves state of the art in 5 different datasets covering a range of domains including News, Scientific Literature to Biomedical and Wetlab Reports. We further apply the approach to construct knowledge graph for scientific papers. We create a dataset SciERC for scientific information extraction, which includes expert annotations of scientific terms, relation categories and co-reference links. The resulting knowledge graph is used for paper abstract generation and academic trend analysis.

TABLE OF CONTENTS

Page

List of Figures iii

List of Tables v

Chapter 1: Introduction 1
1.1 Overview of IE tasks 3
1.2 Our approach and Contributions 5
1.3 Dissertation Overview 8

Chapter 2: Background 10
2.1 Structured Prediction for IE 10
2.2 Leveraging unannotated data 14
2.3 IE for scientific Literature 16
2.4 Summary 17

Chapter 3: Semi-supervised Learning on Sequential Tagging for Scientific IE 19
3.1 Problem Definition and Data 21
3.2 Neural Architecture Model 22
3.3 Semi-supervised Learning 24
3.4 Experimental Setup 28
3.5 Experimental Results 29
3.6 Conclusion 34

Chapter 4: Span-based Multi-Task Identification of Entities, Relations, and Coreference 36
4.1 SciERC Dataset 36
4.2 Model 38
4.3 Experimental Setup 42

4.4 Experimental Results . . . 45
4.5 Related Work . . . 47
4.6 Conclusion . . . 47

Chapter 5: A General Framework for Information Extraction using Dynamic Span Graphs 48
5.1 Model . . . 49
5.2 Experiments . . . 54
5.3 Analysis of Graph Propagation . . . 59
5.4 Related Work . . . 62
5.5 Conclusion . . . 64

Chapter 6: Scientific Knowledge Graph Construction . . . 65
6.1 Knowledge Graph Construction . . . 66
6.2 Summary . . . 71

Chapter 7: Conclusion . . . 73
7.1 Summary . . . 73
7.2 Impact . . . 74
7.3 Future directions . . . 74

Bibliography . . . 78

Appendix A: Annotation Guideline for SciERC dataset . . . 94
A.1 Annotation Guideline . . . 94
A.2 Annotation Examples . . . 98

Appendix B: Scientific Knowledge Graph Examples . . . 99

LIST OF FIGURES

Figure Number Page

1.1 A text passage illustrating interactions between entities, relations and coreference links. Some relation and coreference links are omitted. 2

1.2 Example annotation: phrases that refer to the same scientific concept are annotated into the same coreference cluster, such as *MORphological PAser MORPA*, *it* and *MORPA* (marked as red). 7

2.1 Neural Network Structure 11

3.1 Annotated ScienceIE examples. 20

3.2 Label propagation. Gray nodes indicates labeled data while white nodes are unlabeled. Bold font word indicates the current token. The assumption is if two instances are similar according to the graph, the output labels should be similar. 24

3.3 Lattice representation of ULM. Dashed box is the uncertain token which is going to be marginalized over. Arrows and grey nodes are paths to be summed over during training. When all tokens are confident, the score of only one path is calculated. 27

4.1 Overview of the multitask setup, where all three tasks are treated as classification problems on top of shared span representations. Dotted arcs indicate the normalization space for each task. 39

5.1 Overview of our DyGIE model. Dotted arcs indicate confidence weighted graph edges. Solid lines indicate the final predictions. 49

5.2 F1 score of each layer on ACE development set for different number of iterations. $N = 0$ or $M = 0$ indicates no propagation is made for the layer. 61

5.3 Relation F1 broken down by number of entities in each sentence. The performance of relation extraction degrades on sentences containing more entities. Adding relation propagation alleviates this problem. 64

6.1 Knowledge graph construction process. 66

6.2 A part of an automatically constructed scientific knowledge graph with the most frequent neighbors of the scientific term *statistical machine translation (SMT)* on the graph. For simplicity we denote *Used-for (Reverse)* as *Uses*, *Evaluated-for (Reverse)* as *Evaluated-by*, and replace common terms with their acronyms. The original graph is given in Appendix B. 67

6.3 Frequency of detected entities with and without coreferece resolution: using coreference reduces the frequency of the generic phrase *detection* while significantly increasing the frequency of specific phrases. Linking entities through coreference helps disambiguate phrases when generating the knowledge graph. 68

6.4 Frequency of relation types between pairs of entities: (*left*) automatic speech recognition (ASR) and machine translation (MT), (*right*) conditional random field (CRF) and graphical model (GM). We use the most frequent relation between pairs of entities in the knowledge graph. 68

6.5 Precision/pseudo-recall curves for human evaluation by varying cut-off thresholds. The AUC is 0.751 with coreference, and 0.695 without. 69

6.6 Historical trend for top applications of the keyphrase *neural network* in NLP, speech, and CV conference papers we collected. y-axis indicates the ratio of papers that use *neural network* in the task to the number of papers that is about the task. . . . 72

A.1 Annotation example 1 from ACL 98

B.1 An example of our automatically generated knowledge graph centered on *statistical machine translation*. This is the original figure of Figure 6.2. 100

LIST OF TABLES

Table Number Page

3.1 Overall span-level F1 results for keyphrase identification (SemEval Subtask A) and classification (SemEval Subtask B). * indicates tranductive setting. $^{+}$ indicates not documented as either transductive or inductive. - indicates score not reported or not applied. 29

3.2 Ablation study showing impact of neural network configurations of our DYGIE(supervised) model on the dev set. 30

3.3 F1 score on the dev and test sets for using different sources of data for pretraining. 31

3.4 F1 scores of semi-supervised Learning approaches; * shows transductive models. 32

3.5 F1 score results on the test set for different categories: T indicates TASK, P indicates PROCESS, M is MATERIAL and K is Keyword identification (SubTask A). * indicates a transductive model. 33

3.6 Common errors, where blue means golden label our system misses, red means falsely predicted results, and green means correctly predicted spans. 34

4.1 Dataset statistics for our dataset SCIERC and two previous datasets on scientific information extraction. All datasets annotate 500 documents. 38

4.2 Comparison with previous systems on the development and test set for our three tasks. For coreference resolution, we report the average P/R/F1 of MUC, B^3, and $CEAF_{\phi_4}$ scores. 44

4.3 Ablation study for multitask learning on SCIERC development set. Each column shows results for the target task. 45

4.4 Results for scientific keyphrase extraction and extraction on SemEval 2017 Task 10, comparing with previous best systems. 46

5.1 Datasets for joint entity and relation extraction and their statistics. *Ent*: Number of entity categories. *Rel*: Number of relation categories. 54

5.2 F1 scores on the joint entity and relation extraction task on each test set, compared against the previous best systems. * indicates relation extraction system that takes gold entity boundary as input. 56

5.3 Datasets for overlapping entity extraction and their statistics. *Ent*: Number of entity categories. *Overlap*: Percentage of sentences that contain overlapping entities. 58

5.4 Performance on the overlapping entity extraction task, compared to previous best systems. We report F1 of extracted entities on the test sets. 59
5.5 Ablations on the ACE05 development set with different graph propagation setups. −CorefProp ablates the coreference propagation layers, while −RelProp ablates the relation propagation layers. Base is the system without any propagation. 60
5.6 Ablations on the SciERC development set on different graph progation setups. CorefProp has a much smaller effect on entity F1 compared to ACE05. 60
5.7 Entity extraction performance on pronouns in ACE05. CorefProp significantly increases entity extraction F1 on hard-to-disambiguate pronouns by allowing the model to leverage cross-sentence contexts. 62
5.8 Difference in the confusion matrix counts for ACE05 entity extraction associated with adding CorefProp. 63

ACKNOWLEDGMENTS

I would like to thank my advisors Prof. Mari Ostendorf and Prof. Hannaneh Hajishirzi for giving me great support throughout my six years life at UW as a PhD student. They not only provide me with technical support on tackling a certain problem, but also the confidence and vision of developing insightful research ideas in general. I would never have gone this far without their guidance.

I would also like to thank the rest of my thesis committee, Luke Zettlemoyer and Bill Howe for their insightful comments and advice on research and career. I also want to thank Michel Galley, Jianfeng Gao, Chris Brocket, Bill Dolan, Shinji Watanabe and Boyang Li for for their invaluable guidance during my internships at MSR, MERL and Disney Research. I would also like to thank all UW-NLP members and my co-authors who contribute great effort for the process of producing high quality research: Luheng He, Dave Wadden, Rik Koncel-Kedziorski, Ulme Wennberg, Heng Ji, Lifu Huang, Qinyun Wang, Yangfeng Ji, Gina Levow, Richard Wright, Valerie Freeman and Julian Chan.

I would like to thank dear roommate Fei Zhao, my friends and lab mates Ruiyi Li, Yuzhong Liu, Yushi Tan, Hao Fang, Hao Cheng, Yishen Wang, Tong Zhang, Vicky Zayats, Aaron Jaech, Ji He, Trang Tran, Kevin Lybarger, Farah Nadeem, Trang Tran, Aida Amini, James Ferguson, Sachin Mehta, Ellen Wu, Roy Lu, Sewon min, Colin lockard Alex Marin, Julie Medero, Nicole Nichols, Sining Sun, Jingyong Hou and Shobhit Hathi for their company throughout my six years at UW. I would also like to thank my friends Ting-hao Huang, Shoou-i Yu, Zuxuan Wu, Yun-Nung Chen, for their company when I was interning at Pittsburgh.

I wish to acknowledge my parents Yunhui Li and Houdi Luan for being greatest listeners and life mentors. Their love and understanding help me survive countless gloomy days and back me

up with great confidence to conquer new life challenges. I would also like to thank Chiyu Zhang, for being my personal driver, my personal chef and my best friend. This dissertation is dedicated to all of you.

DEDICATION

to my family.

Chapter 1

INTRODUCTION

There exists a vast amount of unstructured electronic text in the world, including newswire, scientific papers, technical documents, chat logs, and so on. One challenge of Natural Language Processing (NLP) is how to understand and automatically extract structured information from large number of unstructured texts. Information Extraction (IE) is the widely studied task of retrieving such structured information. There are many subtasks under the broad field of IE that focus on extracting different types of information such as entity extraction, relation extraction, coreference resolution and entity linking [1, 2, 3, 4, 5, 6]. Figure 1.1 shows an example, where entities such as person (PER) and location (LOC), relations such as located-at (PHYS) as well as coreference links (COREF) are extracted. The goal of my thesis work is to develop a general high performance IE system that can work across many different domains and tasks. In particular, we explore the less well-studied domain of scientific literature, creating a dataset with expert annotation and propose a way to automatically construct scientific knowledge graph.

Due to the large variety of domains and annotation difficulties, most IE tasks suffer from limited annotated resources. Specifically, some domains such as science papers or technical reports require domain expertise, which makes annotation extremely expensive. Therefore, methods that can work on small amount of training data are in great need. This thesis provides a efficient way of improving supervised system performance through semi-supervised learning and unsupervised pre-training.

In addition to data limitations, an important challenge of IE is how to design a general model that can capture the interdependence across different IE tasks, as well as interactions across multiple sentences. Even though each subtask has its own annotation scheme, there are strong interactions between different tasks. For example, in Figure 1.1, it is impossible to predict the entity labels for *This thing* and *it* from within-sentence context alone. However, the antecedent *car* strongly sug-

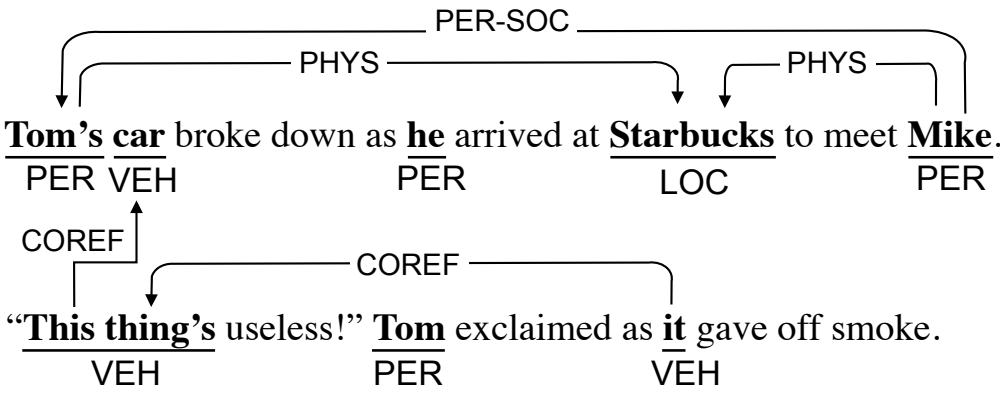


Figure 1.1: A text passage illustrating interactions between entities, relations and coreference links. Some relation and coreference links are omitted.

gests that these two entities have a VEH type. Similarly, the fact that *Tom* is located at *Starbucks* and *Mike* has a relation to *Tom* provides support for the fact that *Mike* is located at *Starbucks*.

In this thesis, we develop a general IE framework through a series of three systems: 1) We introduce a method of integrating a graph-based semi-supervised algorithm together with a confidence based instance selection scheme to leverage unannotated articles, building on a neural sequential tagging framework. 2) We introduce a general IE framework, Span-based IE (SPANIE) which is a multi-task learning model for extracting entities, relations and coreference resolution leveraging shared span representations across all three tasks. 3) We further propose Dynamic Graph IE (DYGIE) which improves SPANIE by leveraging rich contextual span representations across multiple sentences and propagating information through coreference and relation links. Our framework is effective in several domains, demonstrating a benefit from leveraging large unannotated data as well as incorporating broader context learned from relation and coreference annotations. Our model achieves state of the art in 5 different datasets ranging from News, Science to Biomedical and Wetlab Reports. We further apply the approach to construct knowledge graph for scientific papers. We create a dataset SciERC for scientific information extraction, which includes expert annotations of scientific terms, relation categories and co-reference links. The resulting knowledge

graph is used for paper generation and academic trend analysis.

The remaining of the chapter is organized as follows. We first provide an overview of previous IE works. The general approach and main contributions are presented in §1.2. Finally, we provide an overview of the dissertation in §1.3.

1.1 Overview of IE tasks

There are several different IE approaches to construct structured database or knowledge graph from large collection of raw text. Among them, supervised IE is the most accurate option, where a supervised model is trained on labeled texts annotated based on a pre-defined ontology. The performance of supervised systems are usually very sensitive to quantity and quality of the annotated data. In general, human annotation can be expensive and time consuming, especially for domains like scientific literature which requires expert annotations. Therefore, supervised IE systems often suffer from the lack of data problem. Distant learning [7] provides a way of getting cheap annotations through string matching with a high precision, high coverage database. However, the approach is not practical for domains with no such database existed. As an alternative, open domain relation extraction (e.g. OpenIE [8]) is a fully unsupervised approach, which uses processed text strings between the two entities as relations and results in a completely unconstrained knowledge graph. Since the relations are not specified, there is no generalization of these relations and can be hard to use for downstream tasks. Due to the poor generalization ability of OpenIE and requirement of high quality database for distant supervision, in this thesis, we focus on improving the performance of low resource supervised system by leveraging large amount of unannotated texts.

Supervised methods rely on a training set where examples have been labeled based on a domain-specific ontology. Most commonly studied IE tasks including entity extraction, relation extraction and coreference resolution focuses on extracting information from different aspects. Entity Extraction is mainly targeted at extracting spans within a sentence, and classify them into pre-defined categories such as a type of keyphrases or named entities. Relation Extraction is a step further in analyzing information in the texts that connects pairs of entities with a relation type. Coreference resolution is a task to identify when two or more mentions in a text refer to the same

person or thing.

A pre-defined entity and relation is in the form of (s_i, e_i) or (s_i, r_{ij}, s_j) where s_i and s_j are entity spans in pre-defined entity type e_i and e_j respectively. The two entities are connected by a pre-defined relation type r_{ij} or by coreference link. For example, in Figure 1.1, entity instance *(Mike, PER)* and relation triplet *(Mike, PHYS, Starbucks)* would be extracted where *PER* indicates *person* and *PHYS* indicates *located at*. Entities *this thing* and *it* would be connected with coreference links since both referring to *car* in the first sentence.

In previous studies, designed domain specific features and knowledge resources such as gazetteers are widely used [9, 10, 11]. These traditional feature-based methods rely on carefully designed features to learn good models. In addition, due to limited training data, many of those neural models still rely upon domain-specific external syntactic tools to construct dependency paths between the entities in a sentence [12, 13, 1, 14]. These systems suffer from parsing errors from these tools and are hard to generalize to different domains. In addition, in most previous work, each of IE tasks are placed in a pipeline system, with the best output of entity extraction feeding as input to downstream modules such as relation extraction and coreference resolution which may introduce cascading errors.

Recently, supervised models based on neural networks have advanced the state of the art by automatically learning powerful feature representations [13, 15, 16, 2, 17]. With the introduction of LSTM (Long Short Term Memory) [18] units, neural approaches has gain increasing power in modeling sequences for various NLP tasks. In order to improve the performance of neural systems on small amount of training data, transfer learning [19, 20, 21] or initializing the model with pre-trained word embeddings [22, 23, 24, 25, 26, 27] has been widely used. Most recently, contextualized pre-trained embeddings have shown great impact on improving the performance of NLP tasks [28, 29]. In our work, we provide a different way of leveraging unannotated data on top of neural sequential tagging framework, including graph-based semi-supervision [30, 31, 32, 33, 34] and a method for leveraging partially labeled data [35]. We show that the combination of these techniques gives better results than any one alone [36].

In order to avoid the fact of cascading errors, previous studies have explored joint model-

ing [1, 14, 37, 38]) and multi-task learning [39, 3, 40, 41] across all IE subtasks as methods to share representational strength across related information extraction tasks. A span-ranking approach [42] is introduced which reasons over a larger space by jointly detecting mentions and predicting coreference. This thesis extends the span-based approach to a more general multitask learning framework for multiple IE subtasks and further improve the system by incorporating global contexts. We also apply the state-of-the-art pre-trained contextualized embeddings like ELMo [28] and BERT [29] to further improve the performance.

1.2 Our approach and Contributions

In this thesis, I will address the above mentioned limitations (limited training data, pipeline processing and cascading errors) by describing new methods for leveraging large amount of unannotated data through semi-supervised training, a framework to more effectively use data that combines IE tasks using a unified multi-task learning framework and a method to incorporate rich contextual information into the span representations through dynamic graph propagation.

1.2.1 Semi-supervised Neural Tagging

We introduce semi-supervised methods to a neural tagging model for keyphrase extraction tasks in order to take advantage of the large quantity of unlabeled texts. Our semi-supervised learning algorithm uses a graph-based label propagation scheme to estimate the posterior probabilities of unlabeled data. It additionally extends the training objective to leverage the confidence of the estimated posteriors. The new training objective treats low confidence tokens as missing labels and computes the sentence-level score by marginalizing over them. Our experiments show that our neural tagging model achieves state-of-the-art results in the SemEval Science IE task [43].

1.2.2 Multi-task learning across different IE tasks

We develop a unified span-based IE model SPANIE for extracting entities, relations, and coreference resolution, that can reduce the cascading error problem and can model the interaction be-

tween different tasks. Different from a standard tagging system, SPANIE enumerates all possible spans during decoding and can effectively detect overlapped spans. It avoids cascading errors between tasks by jointly modeling all spans and span-span relations. This is different from previous work [36, 44, 45, 46] which often addresses these tasks as independent components of a pipeline. Our unified model is a multi-task setup that shares parameters of first layer LSTMs across low-level tasks. Specifically, we extend prior work for learning span representations and coreference resolution [42, 47]. To make the model more general, we combine the multitask learning framework with ELMo embeddings [28] without relying on external syntactic tools and risking the cascading errors that accompany them.

Most systems based on sequence labeling suffer from an inability to extract entities with overlapping spans. Recently [48] and [49] have presented methods enabling neural models to extract overlapping entities, applying hypergraph-based representations on top of sequence labeling systems. Our framework offers an alternative approach, forgoing sequence labeling entirely and simply considering all possible spans as candidate entities.

1.2.3 Incorporating global context

Even though SPANIE is able to improve performance across different tasks through multi-task learning, there is a obvious limitation on the architecture of SPANIE. SPANIE relies on the first layer LSTM to capture interactions between different tasks where the broader context are incorporated indirectly to span representations via the gradients passed back to the LSTM layer. However, for some IE tasks, the information flow can go across very long distances, sometimes beyond sentence boundaries (e.g. coreference resolution). Therefore, the power of SPANIE to model interactions between tasks is not fully exploited due to the limitation of LSTM on capturing long distance context.

Based on this observation, we proposed DYGIE (Dynamic span Graph IE) to explicitly incorporate rich contextual information into the span representations through dynamic graph propagation. The nodes in the 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.

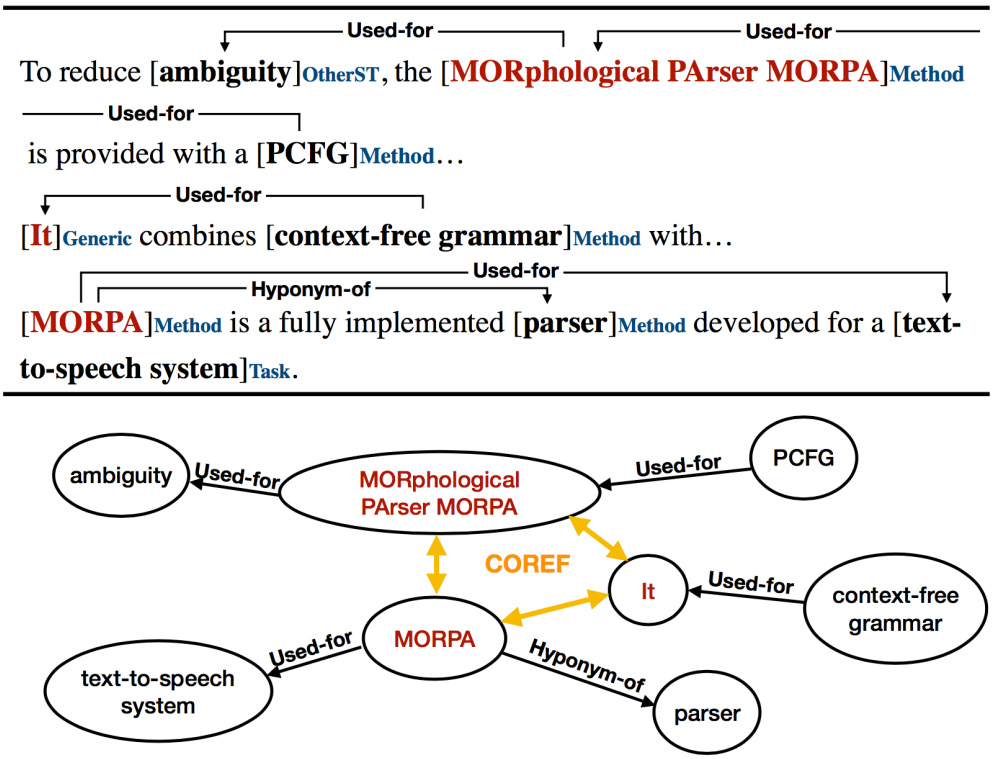


Figure 1.2: Example annotation: phrases that refer to the same scientific concept are annotated into the same coreference cluster, such as *MORphological PAser MORPA*, *it* and *MORPA* (marked as red).

Unlike SPANIE that only shares span representations from the local context [40], DYGIE leverages rich contextual span representations by propagating information through coreference and relation links.

1.2.4 Application on scientific domain

As scientific communities grow and evolve, new tasks, methods, materials and datasets are introduced and different methods are compared with each other. Despite advances in search engines, it is still hard to identify new technologies and their relationships with what existed before. To help researchers more quickly identify opportunities for new combinations of tasks, methods and data,

it is important to design intelligent algorithms that can extract and organize scientific information from a large collection of documents.

Therefore, we apply SPANIE and DYGIE to the application of scientific knowledge graph construction. However, the challenges associated with scientific IE are greater than for a general domain. Extracting information from scientific articles requires extracting relations across sentences.

Figure 1.2 illustrates this problem. The cross-sentence relations between some entities can only be connected by entities that refer to the same scientific concept, including generic terms (such as the pronoun *it*, or phrases like *our method*) that are not informative by themselves. With co-reference, *context-free grammar* can be connected to *MORPA* through the intermediate co-referred pronoun *it*. Since most relation extraction systems are designed for within-sentence relations, applying existing IE systems to this data without co-reference, will result in much lower relation coverage (and a sparse knowledge base).

To explore this problem, we create a dataset SCIERC for scientific information extraction, which includes annotations of scientific terms, relation categories and co-reference links.

1.3 Dissertation Overview

In Chapter 2, we will provide a brief overview of the background of this thesis. We will present some of the most widely used neural-based structured prediction approaches in IE literature. In addition, we will review the literature of semi-supervised learning approaches that have broadly benefit NLP tasks.

In Chapter 3, we will improve the performance of a scientific neural tagging system by leveraging large unannotated data through semi-supervised learning. We show that both inductive and transductive semi-supervised strategies significantly improve the performance. We provide in-depth analysis of domain differences as well as analysis of failure cases.

In Chapter 4, we will present the technical details of our span-based IE models: SPANIE. SPANIEis a unified learning model for extracting scientific entities, relations,and coreference resolution that can model the interaction between different tasks and avoid cascading errors. Our

experiments show that the unified model is better at predicting span boundaries, and it outperforms previous state-of-the-art scientific IE systems on entity and relation extraction.

In Chaper 5, we propose DYGIE by improve SPANIE through directly modeling the information flow across different tasks through dynamic span graphs. We evaluate DYGIE on several datasets spanning many domains (including news, scientific articles, and wet lab experimental protocols), achieving state-of-the-art performance across all tasks and domains and demonstrating the value of coupling related tasks to learn richer span representations.

In Chapter 6, we apply SPANIE to a specific application scenario: scientific knowledge graph construction. To achieve this goal, we also develop the SciERC dataset which has expert annotation for entity, relation and coreference for scientific papers. The automatically constructed knowledge graph is then used for scientific trend analysis and paper abstract generation.

Finally, we conclude and discuss on future directions in Chapter 7. Our work is among the earliest group for scientific knowledge graph construction. We will also conclude the impact of this paper for the NLP community in Chapter 7.

Chapter 2
BACKGROUND

In this chapter, we will provide a brief overview of the background information for this thesis. §2.1 presents an overview of some of the most widely used neural-based structured prediction approaches to IE modeling, since the work in this thesis builds on and advances these state-of-the-art algorithms, we also apply the approach to a real task of a less studied field - knowledge graph construction on scientific papers in §2.3. §2.2 reviews the literature of semi-supervised learning approaches as well as pre-trained contextualized embeddings such as ELMo [28] or BERT [29] which are used to improve the performance of a large range of NLP tasks, and which this work also take advantage of.

2.1 Structured Prediction for IE

The major task of Information Extraction (IE) is to turn unstructured text into structured information. Usually IE can be regarded as a set of process with several different types of information that can be extracted: entity extraction, relations and coreference links between the entities. In the past few years, all the state-of-the-art IE systems are based on neural methods. The main approaches are described below.

2.1.1 Entity extraction through Neural Sequential Tagging

Sequence tagging involves assigning a label to each element of a sequence. Sequence tagging has been a classic NLP task which includes part-of-speech tagging (POS), chunking, and named entity recognition (NER). Tasks like chunking and NER which require detecting the exact span of a multi-token term, can often be casted as sequence tagging problem. In order to be able to distinguish spans of two consecutive terms of the same type, sentences are usually represented in

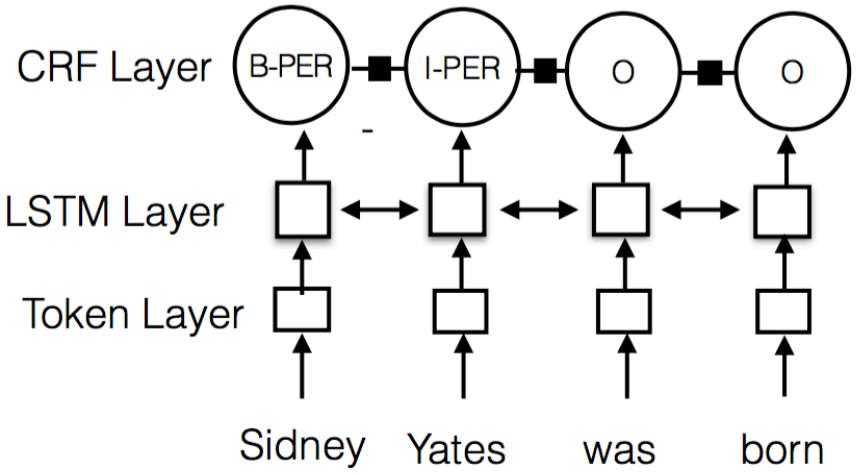


Figure 2.1: Neural Network Structure

the BIO format (Beginning, Inside, Outside) where every token is labeled as B-label if the token is the beginning of a named entity, I-label if it is inside a named entity but not the first token, or O otherwise.

Sequence tagging models generally assumes that the optimal label for a given element depends on the choices of nearby elements. Common models for sequence tagging are statistical models which include the Hidden Markov Model (HMM) and the Conditional Random Field (CRF) [50]. With the introduction of neural approaches, Recurrent Neural Network (RNN) models have been developed to tackle the sequence tagging problem [9, 51]. Long-Short-Term Memory (LSTM) [18] models are a variant of RNNs that combat the RNN vanishing gradient problem by using a series of gates to avoid amplifying or suppressing gradients during backpropagation. LSTMs have proved to outperform other architectures in many NLP applications and have gained a lot of attention recently. Another popular variant is GRU (Gated Recurrent Unit) [**?**], in this thesis, we use LSTM to capture the sequential information of text.

There is also great interest in NNs that use character-based representations [52, 53, 54] to reduce the effect of out-of-vocabulary (OOV) words. Some of the work [54, 52] apply a convolutional neural network (CNN) to the raw character sequence that detects character patterns and

represents them as a fixed-sized embedding vector. Another popular option is to use LSTM [55] to obtain a representation of the word according to the sequential order of characters, which is applied to the model developed in this thesis.

Recent work [55] uses a CRF objective function on top of hybrid word-character LSTM structure and get state-of-the-art result in a NER task. The sequence is tagged using a hierarchical multi-stage model that consists of 3 layers (Figure 2.1):

1. The **Token Representation Layer** is the representation of each token, which can be a word embedding or a character representation or a concaternation of both.

2. The **Token LSTM Layer** uses a bidirectional LSTM to incorporate contextual cues from surrounding tokens to derive intermediate token embeddings.

3. The **CRF Layer** models token-level tagging decisions jointly using a CRF objective function.

In Chapter 3, we extend this BIO tagging framework to the task of scientific keyphrase extraction and introduced semi-supervised learning builds on this method, which significantly boosts the performance.

2.1.2 Relation Extraction

Most neural approaches for relation extraction are built on top of a sequential tagging framework, such as the system described in the previous section. There is usually a core representation learner that takes word embeddings as input and produces contextual entity representations through LSTMs. Based on span boundaries predicted by IBO tagging framework, a representation can be learned for each entity. Such representations are then taken by relation classifiers to produce the final predictions [56, 17, 57].

It is also very common for state-of-the-art RE models to use domain-specific external syntactic tools to construct dependency paths between the entities in a sentence. For example, a neural architecture using a LSTM sequence layer is proposed [1] , followed by a dependency-tree sequence

layer and advance the state-of-the-art performance in multiple single sentence relation extraction tasks. Since these systems suffer from cascading errors from these syntactic tools and are hard to generalize to different domains, in this thesis, our goal is to develop models for IE directly from text, in an end-to-end manner.

2.1.3 Span-based Coreference Resolution

The information extraction systems described above are pipeline systems that take predicted entity boundaries as input for downstream tasks, and therefore could introduce cascading errors. A new framework (E2E-COREF) is proposed [42] for identifying spans, which forgoes sequence labeling entirely and reasons over the space of all spans up to a maximum length. E2E-COREF is the first state-of-the-art neural coreference resolution model that is learned end-to-end given only gold mention clusters. All recent coreference models, including neural approaches that achieved impressive performance gains [58, 59], rely on syntactic parsers, both for head-word features and as the input to carefully hand-engineered mention proposal algorithms. E2E-COREF demonstrate for the first time that these resources are not required, and in fact performance can be improved significantly without them, by training an end-to-end neural model that jointly learns which spans are entity mentions and how to best cluster them.

The model directly optimizes the marginal likelihood of antecedent spans from gold coreference clusters. It includes a span-ranking model that decides, for each span, which of the previous spans is a good antecedent. At the core of the model are vector embeddings representing spans of text in the document, which combine the LSTM representation of the first word and last word in the span with a head-weighted attention mechanism over the span. Since the complexity would be quartic in the document length, enumerating all span pairs in the end-to-end model is impractical. Therefore the model introduces a beam pruning strategy where only the spans with top scores are kept in the beam. The scores indicates how likely is that a span is involved in a coreference cluster. The beam pruning strategy can aggressively reduce the number of pairwise computations.

Inspired by E2E-COREF, in this thesis we extend the framework to a multi-task learning framework for entity, relation and coreference. Our unified model shares parameters across low-level

tasks and makes predictions by leveraging context across the document through coreference links. Different from a standard tagging system, our system enumerates all possible spans during decoding and can effectively detect overlapped spans. It avoids cascading errors between tasks by jointly modeling all spans and span-span relations.

2.2 Leveraging unannotated data

The bottleneck of the supervised methods in information extraction is usually the lack of training data. Therefore, leveraging large unlabeled text sources is very important. Previous work has mainly focused on transfer learning [19, 20], multi-task learning [60, 39, 41] or initializing the model with pre-trained word embeddings [22, 23, 24, 27, 29, 28]. Here we especially focus on two ways of leveraging external resources: semi-supervised learning (SSL) and pre-trained contextualized word embeddings.

2.2.1 Semi-supervised Learning

A common SSL algorithm is self-training, where one makes use of a previously trained model to annotate unlabeled data which is then used to re-train the model. Self-training has been successfully applied to several natural language processing tasks such as word sense disambiguation [61], parsing and machine translation [62]. The EM approach can be viewed as a special case of soft self-training. EM based semi-supervised methods have been successfully applied to many applications such as text classification [61] and face orientation discrimination [63]. One can imagine that a classification mistake can reinforce itself. Some algorithms try to avoid this by ignoring unlabeled points if the prediction confidence is below a threshold [31, 64].

Graph-based SSL algorithms [65, 66] are another important subclass of semi-supervised techniques that have received much attention in the recent past. Graph-based semi-supervised methods define a graph where the nodes are represented as the labeled and unlabeled samples in the dataset, and the edges are the similarity between the samples. Thus constructing the graph requires a function for characterizing the similarity of the two unlabeled samples. Graph-based methods are

based on the assumption that neighboring nodes on the graph tend to have similar labels (smoothness over the graph). Graph-based methods are non-parametric, discriminative, and transductive in nature. Graph-based methods have also been widely used in NLP applications, but mostly focus on unstructured problems such as text classification [67, 68] and machine translation [69]. It can be difficult to construct a graph for structured NLP problems such as POS tagging and NER, since it is hard to use whole sequence similarity to constrain whole tag sequences assigned to linked examples. Subramanya et. al. [70] construct the graph based on subsequences, i.e. word trigrams, where the similarity is based on some hand-designed features. They then use graph-based SSL on top of a CRF structure to improve POS tagging performance. Following [70], similar methods have also been applied to NER [71] and slot filling tasks [72].

In this thesis we use graph-based SSL and a method for leveraging partially labeled data [35] to improve the performance of the neural tagging system described above. We show that the combination of these techniques gives better results than any one alone.

2.2.2 Contextualized word embeddings

Recently, contextualized word embeddings [19, 73, 28, 29] have shown to be effective for improving many natural language processing tasks, such as paraphrasing [74], named entity recognition [75] and SQuAD question answering [76]. Contextualized word embeddings provide an alternative to improve the performance of supervised learning through leveraging unannotated data and are usually used as additional input features to neural networks.

Contextualized word embeddings differ from traditional context-independent word embeddings such as Word2Vec [22] or GLoVe [23] in that for contextualized embeddings, each token is assigned a representation based on the context of the entire input sentence, whereas traditional context-independent embeddings can be stored in a lookup table. Peters et al. [28] proposes ELMo (Embeddings from Language Models) which uses vectors derived from a bidirectional LSTM that is trained with a coupled language model objective on a large text corpus. ELMo embeddings are usually used as an additional feature vector input to task-specific architectures. As an alternative, Devlin et al. [29] proposed a fine-tuning based approach called BERT (Bidirectional Encoder

Representations from Transformers) which further pushes state-of-the-art in multiple NLP tasks. BERT proposes a new pre-training objective, referred to as a masked language models, which is based on the cross entropy of the predicted probability of randomly masked tokens given the left and right word context. Unlike ELMo which pretrains language model from left to right with single direction predictions, the new objective allows pre-training a deep bidirectional transformer and can better capture contexts. In this thesis, we use both ELMo and BERT as input feature vectors to our system, we will compare the performance between the two type of contextualized embeddings.

2.3 IE for scientific Literature

There has been growing interest in research on automatic methods to help researchers search and extract useful information from scientific literature. Past research on this field mainly focused on citation analysis, information extraction and summarization. We will briefly introduce the related work on citation analysis and summarization, and then discuss information extraction in depth which is the focus of this thesis.

Some research investigated citation function, for example analyzing citation function, and predicting the reason for whether citing a paper is positive or negative [77, 78]. Some research focused on citation network and community [79, 80], where the main research problems are about exploring key authors in a field [81, 82], observation of temporal trends in these networks [83], and detecting scientific breakthroughs using text content [84].

Research on summarizing scientific papers has also been extensively explored [85, 80, 86]. Due to scarce hand-annotated data resources, previous work on IE for scientific literature is very limited. Gupta and Manning [44] first proposed a keyphrase extraction task that defines scientific terms into three aspects: *domain*, *technique* and *focus* and apply template-based bootstrapping to tackle the problem. Based on this study, [45] improve the performance by introducing hand-designed features from named entity recognition [87] to the bootstrapping framework.

For relation extraction in scientific literature, most work has been done in the biomedical domain under a distant learning framework e.g. using the Gene Drug Knowledge Database [4, 3]. The main challenge for relation extraction in scientific domains is the long context window that

the relations can embed in. Relations between scientific terms cross-sentence can be chained with coreference [88, 89] and discourse relations [4, 3]. However the performance of coreference and discourse parsers is not sufficiently accurate on scientific domains. There has been various proposed schema in scientific discourse analysis such as [86, 90], but mostly on limited hand annotated data.

More recently, two datasets in SemEval 2017 and 2018 have been introduced, which facilitate research on supervised and semi-supervised learning for scientific information extraction. SemEval 17 includes 500 paragraphs from articles in the domains of computer science, physics, and material science [43]. It includes three types of entities, called keyphrases (Tasks, Methods, and Materials) and two relation types (hyponym-of and synonym-of). SemEval 18 is focused on predicting relations between entities within a sentence annotated on 500 NLP paper abstracts. It consists of six relation types [46] (Topic, Result, Usage, Compare, Part-Whole, Model-Feature).

Using these datasets, neural models [91, 92, 93] are introduced for extracting scientific information. For this thesis, we create a new dataset SciERC by extending both datasets by increasing relation coverage, adding cross-sentence coreference linking [94, 40], and removing some annotation constraints as described in Chapter 3.

2.4 Summary

This chapter presented a brief overview of structured prediction methods for IE, some semi-supervised approaches that has been widely used in NLP, and a survey of NLP on scientific literature. In Chapter 3, we will present more detailed discussions on how to better incorporate semi-supervised approaches with neural structured prediction models to advance the state of the art in IE on scientific literature. We advanced prior work on graph-based SSL for sequence by integrating graph-based label propagation and confidence-aware data selection to a neural sequential tagging model. In particular, we focus on keyphrase extraction task in scientific domain. In Chapters 4 and 5, we will introduce two unified neural IE models, based on span enumeration and multi-task learning for extracting entity, relation and coreference. We also combine our model with pre-trained contextualized embeddings such as ELMo and BERT. Both models are trained

in an end-to-end manner, without relying on any syntactic tools or pipelined process. Chapter 5 improves the multi-task model in Chapter 4 by incorporating global context through dynamic span graphs.

Chapter 3

SEMI-SUPERVISED LEARNING ON SEQUENTIAL TAGGING FOR SCIENTIFIC IE

As a research community grows, more and more papers are published each year. As a result there is increasing demand for improved methods for finding relevant papers and automatically understanding the key ideas in those papers. The purpose of this work is to extract phrases that can answer questions that researchers usually face when reading a paper: What TASK has the paper addressed? What PROCESS or method has the paper used or compared to? What MATERIALS has the paper utilized in experiments? While these fundamental concepts are important in a wide variety of scientific disciplines, the terms that are used in specific disciplines can be substantially different. For example, MATERIALS in computer science might be a text corpus, while they would be physical materials in physics or materials science.

However, due to the large variety of domains and extremely limited annotated resources, there has been relatively little work on scientific information extraction. As described in previous chapter, early work [44, 45] relied on bootstrapping with hand-designed templates to extract scientific keyphrase. The opportunity to use supervised IE models opened with the availability of the SemEval Task 10.

The SemEval Task 10 dataset consisting of 500 scientific paragraphs with keyphrase annotations for three categories (TASK, PROCESS, MATERIAL) across three scientific domains, Computer Science (CS), Material Science (MS), and Physics (Phy) [43][1] as in Figure 3.1, has been released recently. This dataset enables the use of more advanced approaches such as neural network (NN) models. To that end, we cast the keyphrase extraction task as a sequence tagging problem, and build on an LSTM-CRF model used for CRF [55, 39]. Like named entities, keyphrases can be

[1] SemEval (Task 10)`https://scienceie.github.io/index.html`

Computer Science:

This paper addresses the task of **[named entity recognition]**$_{\text{Task}}$, using **[conditional random fields]**$_{\text{Process}}$. Our method is evlauated on the **[ConLL NER Corpus]**$_{\text{Material}}$.

Physics:

[Local field effects] $_{\text{Process}}$ on spontaneous emission rates within **[nanostructure photonics material]**$_{\text{Material}}$ for example are familiar, and have been well used.

Material Science:

The **[Kelvin probe force microscopy technique]** $_{\text{Process}}$ allows **[detection of local EWF]**$_{\text{Task}}$ between an **[atomic force micorscopy]**$_{\text{Material}}$ and **[metal surface]**$_{\text{Material}}$.

Figure 3.1: Annotated ScienceIE examples.

identified by their linguistic context, e.g. researchers ”use” methods. In addition, keyphrases can be associated with different categories in different contexts. For example, ‘semantic parsing’ can be labeled as a TASK in one article and as a PROCESS in another. However, scientific keyphrases differ from named entities in traditional news IE tasks in that they can include both noun phrases and verb phrases and in that non-standard “words” (equations, chemical compounds, references) can provide important cues.

Since the scale of the data in SemEval is still small for supervised training of neural systems, we introduce semi-supervised methods to the neural tagging model in order to take advantage of the large quantity of unlabeled scientific articles. This is particularly important because of the differences in keyphrases across domains. Our semi-supervised learning algorithm uses a graph-based label propagation scheme to estimate the posterior probabilities of unlabeled data. It additionally extends the training objective to leverage the confidence of the estimated posteriors. The new training treats low confidence tokens as missing labels and computes the sentence-level score by marginalizing over them.

The key contributions of our work include: i) achieving state of the art in scientific information extraction SEMEVAL Task 10 by extending recent advances in neural tagging models;

ii) introducing a semi-supervised learning algorithm that uses graph-based label propagation and confidence-aware data selection which significantly improves performance, iii) exploring different alternatives for taking advantage of large, multi-domain unannotated data including both unsupervised embedding initialization and semi-supervised model training. [2]

3.1 Problem Definition and Data

Data We use the SemEval 2017 Task 10 ScienceIE dataset. Fig. 3.1 provides examples that illustrate the variation in domains, but also show that there are common cues such as "the task of", "using", "technique," etc.

The SemEval ScienceIE (SE) corpus consists of 500 journal articles; one paragraph of each article is randomly selected and annotated. The complete unlabeled articles and their metadata are provided together with the labeled data. The training data consists of 350 documents; 50 are kept for development and 100 for testing. The 500 articles come from 82 different journals evenly distributed in three domains. We manually labeled 82 journal names in the dataset into the three domains and do analysis based on the domain partitions. The 500 full articles contains 2M words and is 30 times the size of the annotated data.

A challenge with this dataset is that the size of the training data is very small. It is built from ScienceDirect open access publications and consists of 500 journal articles, but only one paragraph of each article is manually labeled. Therefore, we use a large amount of external data to leverage the continuous-space representation of language of a neural network model. We explore the effect of pre-training word embeddings with two different external resources: i) a data set of Wikipedia articles as a general English resource, and ii) a data set of 50k Computer Science papers from ACM.[3]

Tagging Problem Formulation The task requires detecting the exact span of a keyphrase. In order to be able to distinguish spans of two consecutive keyphrases of the same type, we assign

[2]This chapter is primarily describing work that has been published in [36].

[3]Due to the difficulty of data collection, experiments with external data from the other two domains is left to future work.

labels to every word in a sentence, indicating position in the phrase and the type of phrase. We formulate the problem as an IOBES (Inside, Outside, Beginning, End and Singleton) tagging problem where every token is labeled either as: B, if it is at the beginning of a keyphrase; E, if it ends the phrase; I, if it is inside a keyphrase but not the first or last token; S, if it is a single-word keyphrase; or O, otherwise. For example, "named entity recognition" in first sentence of Fig. 3.1 is labeled as "*B-Task I-task E-task*".

3.2 Neural Architecture Model

We introduce an end-to-end model to categorize scientific keyphrases, building on a neural named entity recognition model [55] and adding a feature-based embedding.

3.2.1 Model

We develop a 3-layer hierarchical neural model to tag tokens of the documents (details of the tokenization is in Sec. 4.3). (1) The token representation layer concatenates three components for each token: a bi-directional character-based embedding, a word embedding, and an embedding associated with orthographic and part-of-speech features. (2) The token LSTM layer uses a bidirectional LSTM to incorporate contextual cues from surrounding tokens to derive intermediate token embeddings. (3) The CRF tagging layer models token-level tagging decisions jointly using a CRF objective function to incorporate dependencies between tags.

Character-Based Embedding. The embedding for a token is derived from its characters as the concatenation of forward and backward representations from a bidirectional LSTM. The character lookup table is initialized at random. The advantage of building a character-based embedding layer is that it can handle out-of-vocabulary words and equations, which are frequent in this data, all of which are mapped to "UNK" tokens in the Word Embedding Layer.

Word Embedding. Words from a fixed vocabulary (plus the unknown word token) are mapped to a vector space, initialized using Word2vec pre-training with different combinations of corpora.

Feature Embedding. We map features to a vector space: capitalization (all capital, first capital, all lower, any capital but first letter) and Part-of-Speech tags.[4] We randomly initialize feature vectors and train them together as other parameters.

Token LSTM Layer We apply a bidirectional LSTM at the token level taking the concatenated character-word-feature embedding as input. The token representation obtained by stacking the forward and backward LSTM hidden states is passed as input to a linear layer that project the dimension to the size of label type space and is used as input to CRF layer.

CRF Layer Keyphrase categorization is a task where there is strong dependencies across output labels (e.g., I-TASK cannot follow B-Process). Therefore, instead of making independent tagging decisions for each output, we model them jointly using conditional random field [50]. For an input sentence $\boldsymbol{x} = (x_1, x_2, x_3, \ldots, x_n)$, we consider P to be the matrix of scores output by the bidirectional LSTM network. P is of size $n \times m$, where n is the number of tokens in a sentence, and m is the number of distinct tags. $P_{t,i}$ corresponds to the score of the i-th tag of the t-th word in a sentence. We use a first-order Markov Model and define a transition matrix $\boldsymbol{T}$ where $\boldsymbol{T}_{i,j}$ represents the score from tag i to tag j. We also add y_0 and y_n as the *start* and *end* tags of a sentence. Therefore $\boldsymbol{T}$ becomes a square matrix of dimension $m + 2$.

Given one possible output $\boldsymbol{y}$, and neural network parameters θ we define the score as

$$\phi(\boldsymbol{y}; \boldsymbol{x}, \theta) = \sum_{t=0}^{n} T_{y_t, y_{t+1}} + \sum_{t=1}^{n} P_{t, y_t} \tag{3.1}$$

The probability of sequence $\boldsymbol{y}$ is obtained by applying a softmax over all possible tag sequences

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) = \frac{\exp(\phi(\boldsymbol{y}; \boldsymbol{x}, \theta))}{\sum_{\boldsymbol{y}' \in Y} \exp(\phi(\boldsymbol{y}'; \boldsymbol{x}, \theta))} \tag{3.2}$$

where Y denotes all possible tag sequences. The normalization term is efficiently computed using the forward algorithm.

[4]Dependency features were investigated but did not lead to performance gains.

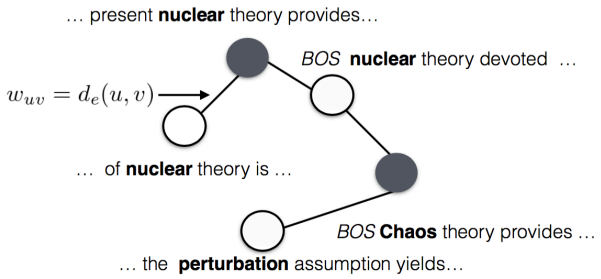


Figure 3.2: Label propagation. Gray nodes indicates labeled data while white nodes are unlabeled. Bold font word indicates the current token. The assumption is if two instances are similar according to the graph, the output labels should be similar.

Supervised Training During training, we maximize the log-probability $\mathcal{L}(\boldsymbol{Y}; \boldsymbol{X}, \theta)$ of the correct tag sequence given the corpus $\{\boldsymbol{X}, \boldsymbol{Y}\}$. Back-propagation is done based on a gradient computed using sentence-level scores.

3.3 Semi-supervised Learning

We develop a semi-supervised algorithm that extends self-training by estimating the labels of unlabeled data and then using those labels for re-training. Specifically, we use a graph-based algorithm to estimate the posterior probabilities of unlabeled data and develop a new CRF training to take the uncertainty of the estimated labels into account while optimizing the objective function.

We develop a semi-supervised algorithm that extends self-training by estimating the labels of unlabeled data and then using those labels for re-training. Specifically, we use a graph-based algorithm to estimate the posterior probabilities of unlabeled data and develop a new CRF training to take the uncertainty of the estimated labels into account while optimizing the objective function.

3.3.1 Graph-based Posterior Estimates

Our semi-supervised algorithm uses the following steps to estimate the posterior. It first constructs a graph of tokens based on their semantic similarity, then uses the CRF marginal as a regularization

term to do label propagation on the graph. The smoothed posterior is then used to either interpolate with the CRF marginal or as an additional feature to the neural network.

Graph Construction Vertices in the graph correspond to tokens, and edges are associated with the distance between token features which capture semantic similarity. The total size of the graph is equal to the number of tokens in both labeled data V_l and unlabeled data V_u. The tokens are modelled with a concatenation of pre-trained word embeddings (with dimension d) of the 5-gram centered by the current token, the word embedding of the closest verb, and a set of discrete features including part-of-speech tags and capitalization (43 and 4 dimension one-hot features). The resulting feature vector with dimension of $5d + d + 43 + 4$ is then projected down to 100 dimensions using PCA. We define the weight w_{uv} of the edge between nodes u and v as follows: $w_{uv} = d_e(u, v)$ if $v \in \mathcal{K}(u)$ or $u \in \mathcal{K}(v)$, where $\mathcal{K}(u)$ is the set of *k*-nearest neighbors of u and $d_e(u, v)$ is the Euclidean distance between any two nodes u and v in the graph. An example part of our graph is in Fig. 3.2, using 4-gram for brevity.

Label Propagation We use prior-regularized measure propagation [64, 30] to propagate labels from the annotated data to their neighbors in the graph. The algorithm aims for the label distribution between neighboring nodes to be as similar to each other as possible by optimizing an objective function that minimizes the Kullback-Leibler distances between: i) the empirical distribution $\boldsymbol{r}_u$ of labeled data and the predicted label distribution $\boldsymbol{q}_u$ for all labeled nodes in the graph; ii) the distributions $\boldsymbol{q}_u$ and $\boldsymbol{q}_v$ for all nodes u in the graph and their neighbors v; iii) the distributions $\boldsymbol{q}_u$ and the CRF marginals $\tilde{\boldsymbol{p}}_u$ for all nodes. The third term regularizes the predicted distribution toward the CRF prediction if the node is not connected to a labeled vertex, ensuring the algorithm performs at least as well as standard self-training.

Posterior Estimates We develop two strategies to estimate the new posteriors $\hat{p}(y_t|\boldsymbol{x}; \theta)$, which can then be used in the CRF training. In the inductive setting, we only use the unlabeled data from the development set for the semi-supervision. In the transductive setting we also use the unlabeled data of the test set to construct the graph. In both cases, the parameters are tuned only on the dev

set.

The first strategy (called GRAPHINTERP) is the commonly used approach [70, 95] that interpolates the smoothed posterior $\{\boldsymbol{q}\}$ with CRF marginals p:

$$\hat{p}(y_t|\boldsymbol{x};\theta) = \alpha p(y_t|\boldsymbol{x};\theta) + (1-\alpha)q(y) \tag{3.3}$$

where α is a mixing coefficient.

A second strategy introduced here (called GRAPHFEAT) uses the smoothed posterior $\{\boldsymbol{q}\}$ as features and learns it with other parameters in the neural network. Given a sentence $\{x_1, \ldots, x_n\}$, let $Q = \{\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n\}$ be the predicted label distribution from the graph. We then use the $m \times n$ Q as a feature input to neural network as $\tilde{P} = P + MQ$ where P is the $m \times n$ matrix output by the bidirectional LSTM network as in Eq. 3.1, and M is an $m \times m$ matrix that is learned together with other parameters of the neural network. In CRF layer, we modify Eq. 3.1 by replacing P_{t,y_t} with $\tilde{P}_{t,y_t}$. Note that GRAPHFEAT can only be done in a transductive way since it requires output Q from the graph at test time.

3.3.2 CRF SSL with Uncertain Labels

A standard approach to self-training is to make hard decisions for labeling tokens based on the estimated posteriors and retrain the model, or use the posteriors as soft labels in an EM algorithm. However, the estimated posteriors in our task are noisy due to the difficulty and variety of the ScienceIE task. We want to leverage only the most confident labels, but since this is a sequence modeling task, we can not ignore other elements in the sequence. New semi-supervised CRF training algorithm (called Uncertain Label Marginalizing (ULM)) that treats low confidence tokens as missing labels and computes the sentence-level score by marginalizing over them. A similar idea has been previously used in treating partially labeled data [35].

Specifically, given a sentence $\boldsymbol{x}$ we define a constrained *lattice* $\mathcal{Y}_n(\boldsymbol{x})$, where at each position t the allowed label types $\mathcal{Y}(x_t)$ are:

$$\mathcal{Y}_n(x_t) = \begin{cases} \{y_t\}, & \text{if} \quad p(y_t|\boldsymbol{x};\theta^n) > \eta \\ \text{All label types}, & \text{otherwise} \end{cases} \tag{3.4}$$

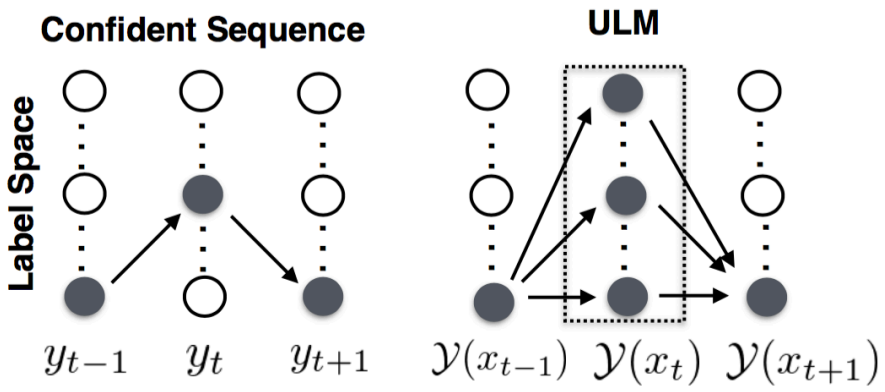


Figure 3.3: Lattice representation of ULM. Dashed box is the uncertain token which is going to be marginalized over. Arrows and grey nodes are paths to be summed over during training. When all tokens are confident, the score of only one path is calculated.

where η is the confidence threshold, y_t is the prediction of posterior decoding and $p(y_t|\boldsymbol{x};\theta)$ is its CRF token-level posterior. The new neural network parameters $\theta^{(n+1)}$ are estimated by maximizing the log-likelihood of $p(\mathcal{Y}(\boldsymbol{x}^j)|\boldsymbol{x}^j,\theta)$ for every input sentence $\boldsymbol{x}^j$, where

$$p(\mathcal{Y}(\boldsymbol{x}^j)|\boldsymbol{x}^j,\theta) = \frac{\sum_{\boldsymbol{y}^j \in \mathcal{Y}(\boldsymbol{x}^j)} \exp(\phi(\boldsymbol{y}^j;\boldsymbol{x}^j,\theta))}{\sum_{\boldsymbol{y}' \in Y} \exp(\phi(\boldsymbol{y}';\boldsymbol{x},\theta))}$$

where $\boldsymbol{y}^j$ is an instance sequence of lattice $\mathcal{Y}(\boldsymbol{x})$, and j is the sentence index in the training set. Extreme cases are when all tokens are uncertain then the likelihood would be equal to 1, when all tokens of a sequence are confident, it would be equal to Eq. 3.2 where only one possible sequence, as in the left figure in Fig. 3.3.

3.3.3 Combining Graph-based SSL and ULM

The integrated semi-supervised training process is summarized as follows:

- Initialize: Estimate CRF parameters from labeled data.
- Iterate:

  (i) Compute label posteriors over the unlabeled data using the CRF.

(ii) Use the CRF posteriors in a regularization term for label propagation.

(iii) Combine the CRF posteriors $\tilde{p}^n$ and graph posteriors q^n to add labels to the training set.

(iv) Estimate new CRF parameters.

3.4 Experimental Setup

Implementation details All parameters are tuned on the dev set performance, the best parameters are selected and fixed for model switching and semi-supervised systems. The word embedding dimension is 250; the token-level hidden dimension is 100; the character-level hidden dimension is 25; and the optimization algorithm is SGD with a learning rate of 0.05. Two special tokens *BOS* and *EOS* are added when pre-training, indicating the begin and end of a sentence. The number of the graph vertices is 2M in tranductive setting and 1.4M in inductive setting. The ULM parameter η in Eq. 3.4 is tuned from 0.1 to 0.9, the best η is 0.4. The best parameters of label propagation are $\mu = 10^{-6}$ and $\nu = 10^{-5}$. The interpolation parameter α in Eq. 3.3 is tuned from 0.1 to 0.9, the best α is 0.3. We do iteration of semi-supervised learning until we obtain the best result on the dev set, which is mostly achieved in the second round.

We use Stanford CoreNLP [96] tokenizer to tokenize words. The tokenizer is augmented with a few hand-designed rules to handle equations (e.g. "fs(B,t)=Spel(t)S" is a single token) and other non-standard word phenomena (Cu40Zn, 20MW/m2) in scientific literature. We use Approximate Nearest Neighbor Searching (ANN)[5] to calculate the *k*-nearest neighbors. For all experiments in this thesis, $k = 10$.

Setup We evaluate SSL algorithms in both inductive and transductive settings. The inductive setting uses 400 full articles in ScienceIE training and dev sets, while the transductive setting uses 500 full articles including the test set. In both settings parameters are tuned over the dev set.

Comparisons We compare our system with two template matching baselines and the state-of-the-art on the SemEval Science IE task. The first baseline [44] is an unsupervised method to extract keyphrases by initially using seed patterns in a dependency tree, and then adding to seed patterns

[5] `https://www.cs.umd.edu/~mount/ANN/`

Span Level	Classification (dev)	Classification (test)	Identification
Gupta et.al.(unsupervised)	-	9.8	6.4
Tsai et.al. (unsupervised)	-	11.9	8.0
MULTITASK	45.5	-	-
Best Non-Neural SemEval^{+}	-	38	51
Best Neural SemEval^{+}	-	44	56
DYGIE(supervised)	48.1	40.2	52.1
DYGIE(semi)	51.9	45.3	56.9
DYGIE(semi)*	**52.1**	**46.6**	**57.6**

Table 3.1: Overall span-level F1 results for keyphrase identification (SemEval Subtask A) and classification (SemEval Subtask B). * indicates tranductive setting. $^{+}$ indicates not documented as either transductive or inductive. - indicates score not reported or not applied.

through bootstrapping. The second baseline [45] improves the work of [44] by adding Named Entity Features and use different set of seed patterns.

3.5 Experimental Results

We evaluate our NN-CRF model in both supervised and semi-supervised settings. We also perform ablations and try different variants to best understand different learning strategies.

3.5.1 Best Case System Performance

Table 3.1 reports the results of our neural sequence tagging model NN-CRF in both supervised and semi-supervised learning (ULM and graph-based), and compares them with the baselines and the state-of-the-art (best SemEval System [43]).

A multi-task learning strategy [93] is use to improve the performance of supervised keyphrase classification, but they only report dev set performance on SemEval Task 10, we also include their result here and refer to it as MULTITASK. We report results for both span identification (SemEval

Model	P	R	F1
DyGIE(supervised)	46.2	48.2	47.2
No features	44.2	46.1	45.1
No bi-LSTM	45.2	44.7	44.9
No CRF	36.7	38.2	37.4
No char	45.7	46.2	45.9

Table 3.2: Ablation study showing impact of neural network configurations of our DyGIE(supervised) model on the dev set.

SubTask A) and span classification into Task, Process and Material (SemEval Subtask B).[6]

The results show that our neural sequence tagging models significantly outperform the state of the art and both baselines. It confirms that our neural tagging model outperforms other non-neural and neural models for the SemEval ScienceIE challenge.[7] It further shows that our system achieves significant boost from semi-supervised learning using unlabeled data.

3.5.2 Supervised Learning

Impact of Neural Model Components Table 4.3 provides the results of an ablation study on the dev set showing the impact of different components of our NN-CRF on the Scientific IE task. For the basic model, the word embeddings are initialized by word2vec trained on the 350 full journal articles in the SE training set together with Wikipedia and ScienceIE data. The feature layer, character layer, and bi-LSTM word layers all improves the performance. Moreover, we observe a large improvement (20.6% relative) in the scientific IE task by adding the CRF layer.

Initialization Table 3.3 reports our NN-CRF performance when pretrained on different domains. We explore different word embedding pre-training with ScienceIE training set alone (SE),

[6]The evaluation script is provided by the challenge, with a modification to report 3 decimal precision results.

[7]Best SemEval Numbers from https://scienceie.github.io/

	Dev			Test		
Initialization	MS	Phy	CS	MS	Phy	CS
SE	49.4	39.4	45.0	42.9	33.0	30.5
+wiki	**52.9**	**40.5**	47.9	**46.1**	**39.2**	31.0
+ACM	50.3	39.8	**49.5**	42.2	37.8	34.2
+wiki+ACM	50.5	40.3	48.9	43.1	37.9	**34.4**

Table 3.3: F1 score on the dev and test sets for using different sources of data for pretraining.

and adding other external resources including Wikipedia (wiki) and Computer Science articles (ACM). All alternatives use word2vec. Compared with using SE alone, introduction of all external data sources improve performance. Moreover, we observe that with the introduction of the ACM dataset, the performance on the CS domain is increased significantly in both the dev and test sets. Adding Wikipedia data benefits all three domains, with more significant improvement on the MS and Physics domains.

Based on these observations, we select the best model on each domain according to the dev set and use the combined result as our best suprevised system (called NN-CRF(supervised)). The F1 score improves from 39.4 to 40.2 when applying model switching strategy. The best model on the dev set is used for each domain: for MS and physics domain, we pretrain word embeddings with the SE and Wiki, and for the CS domain, we pretrain with the SE and ACM.

3.5.3 Semi-Supervision Learning

Table 3.4 reports the results of the semi-supervised learning algorithms in different settings. In particular we ablate incorporating the graph-based methods of computing the posteriors and CRF training (ULM vs. hard decision). The table shows incorporating graph-based methods for computing posterior and ULM for CRF training outperforms their counterparts.

For computing the posteriors, we explore two different strategies of the graph-based methods GRAPHINTERP and GRAPHFEAT. GRAPHINTERP interpolates the smoothed posterior from label

Posterior	Training	Dev	Test
-	-	50.2	42.9
-	ULM	51.3	44.4
GRAPHINTERP	-	50.9	43.3
GRAPHINTERP	ULM	**51.9**	**45.3**
GRAPHINTERP*	-	50.7	44.0
GRAPHINTERP*	ULM	51.8	45.7
GRAPHFEAT*	-	51.4	44.9
GRAPHFEAT*	ULM	**52.1**	**46.6**

Table 3.4: F1 scores of semi-supervised Learning approaches; * shows transductive models.

propagation with CRF posteriors; In the inductive setting, GRAPHINTERP only uses un-annotated data from the dev set and uses the best model for decoding at test time. In the transductive setting, GRAPHINTERP* uses un-annoated data from the test set to build the graph as well, tuning the parameters on the dev set. GRAPHFEAT uses the smoothed posterior from label propagation as an additional input to the neural network and only applies in the transductive setting.

As expected, the transductive approaches consistently outperform inductive approaches on the test set. With around the same performance on the dev set, GRAPHINTERP* seems to generalize better on the test set with 1.6% relative improvement over GRAPHINTERP. We observe higher improvement with GRAPHFEAT* compared to GRAPHINTERP. This is mainly because automatically learning the weight matrix M between neural network scores and graph outputs adds more flexibility compared to tuning an interpolation weight α. The performance is further improved by applying data selection through modifying the objective to ULM. The best inductive system is ULM+GRAPHINTERP with 5.6% relative improvement over pure Self-Training that makes hard decisions, and the best transductive system is ULM+GRAPHFEAT* with 8.6% relative improvement.

Span Level	T	P	M	K
Best SemEval	19	44	48	55
supervised	13.3	40.5	43.7	52.1
ULM+GRAPHINTERP	17.0	45.4	49.4	56.9
ULM+GRAPHFEAT*	17.2	46.5	50.7	57.6
Token Level	**T**	**P**	**M**	**K**
supervised	29.6	56.0	59.3	70.8
ULM+GRAPHINTERP	40.0	60.7	61.2	77.0
ULM+GRAPHFEAT*	40.1	62.8	63.4	78.1

Table 3.5: F1 score results on the test set for different categories: T indicates TASK, P indicates PROCESS, M is MATERIAL and K is Keyword identification (SubTask A). * indicates a transductive model.

3.5.4 Error Analysis

Table 3.5 details the performance of our method on the three categories at the span and token level. We observe significant improvement by using ULM+GRAPHINTERP and ULM+GRAPHFEAT over the best SemEval result and our best supervised system on all three categories at both token and span levels. We further observe that all systems have much lower performance on TASK classification than PROCESS and MATERIAL. This is in part because TASK is much less frequent than the other types. In addition, TASK keyphrases often include verb phrases, while the other two domains mainly consists of noun phrases. An analysis of confusion patterns show that the most frequent type confusions are between PROCESS and MATERIAL. However, we observe that ULM+GRAPHFEAT* can greatly reduce the confusion, with 3.5% relative improvement of PROCESS and 3.6% relative improvement of PROCESS over ULM+GRAPHINTERP on token level.

We provide examples of typical errors that our system makes in Table 3.6. As described above, TASK is the hardest type to identify with our system. Row 1 shows a failure to detect the verb phrase

Error types	Annotation and System Output
Verb phrases	A key requirement in aiming to [achieve [enantiopure products]$_{\text{Material}}$]$_{\text{Task}}$ is therefore a means to [quantitate [the enantiometric excess]$_{\text{Process}}$]$_{\text{Task}}$.
General terms	Since the [receptors]$_{\text{Material}}$ in human biology mostly consist of [chiral molecules]$_{\text{Material}}$, [drug action]$_{\text{Process}}$ mostly involves a specified enantiometric form.
Falsely predicted adjectives	It has been shown that the most efficient forms of energy transfer between the two occurs when there is a [neighbouring carotenoid species]$_{\text{Material}}$.
Lack of context	Other models use [SWEs]$^{\text{Material}}_{\text{Process}}$ but focus on the use of multi resolution grids or irregular mesh.

Table 3.6: Common errors, where blue means golden label our system misses, red means falsely predicted results, and green means correctly predicted spans.

following 'to' as part of the TASK, and instead detects 'enantiopure products' as MATERIAL. Row 2 illustrates the problem of identifying general terms as keyphrases due to similar context, such as 'receptors' and 'drug action'. A third common error involves incorrectly labeling adjectives, such as 'neighbouring' in Row 3, which leads to span errors. Another common cause of error is insufficient context: in the last example, a larger context is needed to determine whether 'SWE' is a PROCESS or MATERIAL.

3.6 Conclusion

In this chapter, we cast the scientific information extraction task as a sequence tagging problem and introduce a hierarchical LSTM-CRF neural tagging model for this task, building on recent results in NER. We introduced a semi-supervised learning algorithm that incorporates graph-based label propagation and confidence-aware self-training for sequential models. We show the introduction of semi-supervision significantly outperforms the performance of the supervised LSTM-CRF tagging model. We additionally show that external resources are useful for initializing word embeddings.

Both inductive and transductive semi-supervised strategies achieve state-of-the-art performance in SemEval 2017 ScienceIE task. We also conducted a detailed analysis of the system and point out common error cases.

In our experiments, we observe that including only domain-matched data for semi-supervised learning has slightly better performance than using cross-domain data. Reducing the amount of in-domain data hurts performance. It would be useful to assess the impact of matched unlabeled data for the physics and material science domain. Other future work includes leveraging global context, such as information from the citation network.

Chapter 4

SPAN-BASED MULTI-TASK IDENTIFICATION OF ENTITIES, RELATIONS, AND COREFERENCE

In this chapter, we develop a unified learning model for extracting scientific entities, relations, and coreference resolution that can model the interaction between different tasks and avoid cascading errors. This is different from previous work [36, 44, 45, 46] which often addresses these tasks as independent components of a pipeline. Our unified model is a multi-task setup that shares parameters across low-level tasks, making predictions by leveraging context across the document through coreference links. Specifically, we extend prior work for learning span representations and coreference resolution [42, 47]. Different from a standard tagging system, our system enumerates all possible spans during decoding and can effectively detect overlapped spans. It avoids cascading errors between tasks by jointly modeling all spans and span-span relations.

By extending the previous end-to-end coreference resolution system, we develop the multi-task learning framework that can detect scientific entities, relations, and coreference clusters without hand-engineered features. Our experiments show that the unified model is better at predicting span boundaries, and it outperforms previous state-of-the-art scientific IE systems on entity and relation extraction [36, 43].[1]

4.1 SciERC Dataset

Our dataset (called SciERC) includes annotations for scientific entities, their relations, and coreference clusters for 500 scientific abstracts. These abstracts are taken from 12 AI conference/workshop

[1]This chapter is primarily describing work that has been published in [40]. My contribution involves model design, running experiments and writing.

proceedings in four AI communities from the Semantic Scholar Corpus[2]. SciERC extends previous datasets in scientific articles SemEval 2017 Task 10 (SemEval 17) [43] and SemEval 2018 Task 7 (SemEval 18) [46] by extending entity types, relation types, relation coverage, and adding cross-sentence relations using coreference links. Our dataset is publicly available at: `http://nlp.cs.washington.edu/sciIE/`. Table 4.1 shows the statistics of SciERC.

Annotation Scheme We define six types for annotating scientific entities (Task, Method, Metric, Material, Other-ScientificTerm and Generic) and seven relation types (Compare, Part-of, Conjunction, Evaluate-for, Feature-of, Used-for, Hyponym-Of). Directionality is taken into account except for the two symmetric relation types (Conjunction and Compare). Coreference links are annotated between identical scientific entities. A Generic entity is annotated only when the entity is involved in a relation or is coreferred with another entity. Annotation guidelines can be found in Appendix A.

Following annotation guidelines from [97] and using the BRAT interface [98], our annotators perform a greedy annotation for spans and always prefer the longer span whenever ambiguity occurs. Nested spans are allowed when a subspan has a relation/coreference link with another term outside the span.

Human Agreements One domain expert annotated all the documents in the dataset; 12% of the data is dually annotated by 4 other domain experts to evaluate the user agreements. Each annotator was shown with the span that has been annotated by the main annotator, and was asked to correct some obvious span errors as well as annotating keyphrase types, links and types (if any) of relation and coreference. The kappa score for annotating entities is 0.77, relation extraction is 0.68 and coreference is 0.64.

---

[2]These conferences include general AI (AAAI, IJCAI), NLP (ACL, EMNLP, IJCNLP), speech (ICASSP, Interspeech), machine learning (NIPS, ICML), and computer vision (CVPR, ICCV, ECCV) at `http://labs.semanticscholar.org/corpus/`

Statistics	SciERC	SemEval 17	SemEval 18
#Entities	8089	9946	7483
#Relations	4716	672	1595
#Relations/Doc	9.4	1.3	3.2
#Relation types	7	2	6
#Coref links	2752	-	-
#Coref clusters	1023	-	-

Table 4.1: Dataset statistics for our dataset SciERC and two previous datasets on scientific information extraction. All datasets annotate 500 documents.

Comparison with previous datasets SciERC is focused on annotating cross-sentence relations and has more relation coverage than SemEval 17 and SemEval 18, as shown in Table 4.1. SemEval 17 is mostly designed for entity recognition and only covers two relation types (Hyponym-of and Synonym-of). The task in SemEval 18 is to classify a relation between a pair of entities given entity boundaries, but only intra-sentence relations are annotated and each entity only appears in one relation, resulting in sparser relation coverage than our dataset (3.2 vs. 9.4 relations per abstract). SciERC extends these datasets by adding more relation types and coreference clusters, which allows representing cross-sentence relations, and removing annotation constraints. In addition, SciERC aims at including broader coverage of general AI communities.

4.2 Model

We develop a unified framework (called SpanIE) to identify and classify scientific entities, relations, and coreference resolution across sentences. SpanIE is a multi-task learning setup that extends previous span-based models for coreference resolution [42] and semantic role labeling [47]. All three tasks of entity recognition, relation extraction, and coreference resolution are treated as multinomial classification problems with shared span representations. SpanIE benefits from expressive contextualized span representations as classifier features. By sharing span representations, sentence-level tasks can benefit from information propagated from coreference resolution across

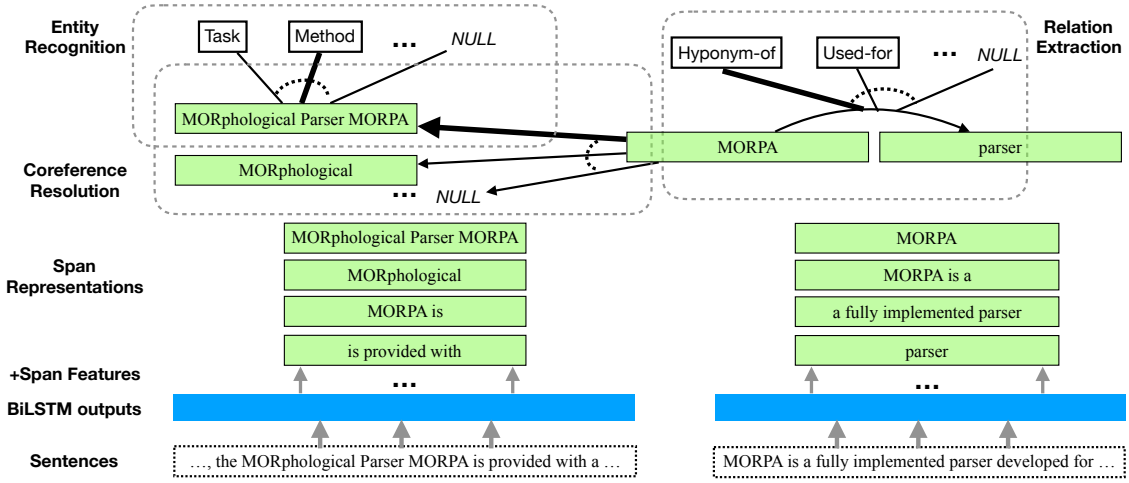


Figure 4.1: Overview of the multitask setup, where all three tasks are treated as classification problems on top of shared span representations. Dotted arcs indicate the normalization space for each task.

sentences, without increasing the complexity of inference.

4.2.1 Problem Definition

The input is a document represented as a sequence of words $D = \{w_1, \ldots, w_n\}$, from which we derive $S = \{s_1, \ldots, s_N\}$, the set of all possible within-sentence word sequence spans (up to a reasonable length) in the document. The output contains three structures: the entity types E for all spans S, the relations R for all pair of spans $S \times S$, and the coreference links C for all spans in S. The output structures are represented with a set of discrete random variables indexed by spans or pairs of spans. Specifically, the output structures are defined as follows.

Entity recognition is to predict the best entity type for every candidate span. Let L_{E} represent the set of all possible entity types including the null-type ϵ. The output structure E is a set of random variables indexed by spans: $e_i \in L_{\text{E}}$ for $i = 1, \ldots, N$.

Relation extraction is to predict the best relation type given an ordered pair of spans (s_i, s_j). Let L_{R} be the set of all possible relation types including the null-type ϵ. The output structure R is a set

of random variables indexed over pairs of spans (i, j) that belong to the same sentence: $r_{ij} \in L_{\text{R}}$ for $i, j = 1, \ldots, N$.

Coreference resolution is to predict the best antecedent (including a special null antecedent) given a span, which is the same mention-ranking model used in [42]. The output structure C is a set of random variables defined as: $c_i \in \{1, \ldots, i-1, \epsilon\}$ for $i = 1, \ldots, N$.

4.2.2 Model Definition

We formulate the multi-task learning setup as learning the conditional probability distribution $P(E, R, C|D)$. Figure 5.1 shows a high-level overview of the SPANIE multi-task framework. For efficient training and inference, we decompose $P(E, R, C|D)$ assuming spans are conditionally independent given D:

$$\begin{aligned} P(E, R, C \mid D) &= P(E, R, C, S \mid D) \\ &= \prod_{i=1}^{N} P(e_i \mid D) P(c_i \mid D) \prod_{j=1}^{N} P(r_{ij} \mid D), \end{aligned} \tag{4.1}$$

where the first equality holds since the spans are known given D. The conditional probabilities of each random variable are independently normalized:

$$\begin{aligned} P(e_i = e \mid D) &= \frac{\exp(\Phi_{\text{E}}(e, s_i))}{\sum_{e' \in L_{\text{E}}} \exp(\Phi_{\text{E}}(e', s_i))} \\ P(r_{ij} = r \mid D) &= \frac{\exp(\Phi_{\text{R}}(r, s_i, s_j))}{\sum_{r' \in L_{\text{R}}} \exp(\Phi_{\text{R}}(r', s_i, s_j))} \\ P(c_i = j \mid D) &= \frac{\exp(\Phi_{\text{C}}(s_i, s_j))}{\sum_{j' \in \{1, \ldots, i-1, \epsilon\}} \exp(\Phi_{\text{C}}(s_i, s_{j'}))}, \end{aligned} \tag{4.2}$$

where Φ_{E} denotes the unnormalized model score for an entity type e and a span s_i, Φ_{R} denotes the score for a relation type r and span pairs s_i, s_j, and Φ_{C} denotes the score for a binary coreference link between s_i and s_j. These Φ scores are further decomposed into span and pairwise span scores computed from feedforward networks, as will be explained in Section 4.2.3.

Objective Given a set of all documents $\mathcal{D}$, the model loss function is defined as a weighted sum of the negative log-likelihood loss of all three tasks:

$$-\sum_{(D,R^*,E^*,C^*)\in\mathcal{D}} \Big\{\lambda_{\text{E}} \log P(E^* \mid D) + \lambda_{\text{R}} \log P(R^* \mid D) + \lambda_{\text{C}} \log P(C^* \mid D)\Big\} \tag{4.3}$$

where E^*, R^*, and C^* are gold structures of the entity types, relations, and coreference, respectively. The task weights λ_{E}, λ_{R}, and λ_{C} are introduced as hyper-parameters to control the importance of each task.

For entity recognition and relation extraction, $P(E^* \mid D)$ and $P(R^* \mid D)$ are computed with the definition in Equation (4.2). For coreference resolution, we use the marginalized loss following [42] since each mention can have multiple correct antecedents. Let C_i^* be the set of all correct antecedents for span i, we have: $\log P(C^* \mid D) = \sum_{i=1..N} \log \sum_{c\in C_i^*} P(c \mid D)$.

4.2.3 Scoring Architecture

We use feedforward neural networks (FFNNs) over *shared span representations* $\mathbf{g}$ to compute a set of span and pairwise span scores. For the span scores, $\phi_e(s_i)$ measures how likely a span s_i has an entity type e, and $\phi_{\text{mr}}(s_i)$ and $\phi_{\text{mc}}(s_i)$ measure how likely a span s_i is a mention in a relation or a coreference link, respectively. The pairwise scores $\phi_r(s_i, s_j)$ and $\phi_{\text{c}}(s_i, s_j)$ measure how likely two spans are associated in a relation r or a coreference link, respectively. Let $\mathbf{g}_i$ be the fixed-length vector representation for span s_i. For different tasks, the span scores $\phi_{\text{x}}(s_i)$ for $\text{x} \in \{e, \text{mc}, \text{mr}\}$ and pairwise span scores $\phi_{\text{y}}(s_i, s_j)$ for $\text{y} \in \{r, \text{c}\}$ are computed as follows:

$$\begin{aligned}\phi_{\text{x}}(s_i) =& \mathbf{w}_{\text{x}} \cdot \text{FFNN}_{\text{x}}(\mathbf{g}_i)\\ \phi_{\text{y}}(s_i, s_j) =& \mathbf{w}_{\text{y}} \cdot \text{FFNN}_{\text{y}}([\mathbf{g}_i, \mathbf{g}_j, \mathbf{g}_i \odot \mathbf{g}_j]),\end{aligned}$$

where $\odot$ is element-wise multiplication, and $\{\mathbf{w}_{\text{x}}, \mathbf{w}_{\text{y}}\}$ are neural network parameters to be learned.

We use these scores to compute the different Φ:

$$\begin{aligned} \Phi_{\text{E}}(e, s_i) &= \phi_e(s_i) \\ \Phi_{\text{R}}(r, s_i, s_j) &= \phi_{\text{mr}}(s_i) + \phi_{\text{mr}}(s_j) + \phi_r(s_i, s_j) \\ \Phi_{\text{C}}(s_i, s_j) &= \phi_{\text{mc}}(s_i) + \phi_{\text{mc}}(s_j) + \phi_{\text{c}}(s_i, s_j) \end{aligned} \tag{4.4}$$

The scores in Equation (4.4) are defined for entity types, relations, and antecedents that are not the null-type ϵ. Scores involving the null label are set to a constant 0: $\Phi_{\text{E}}(\epsilon, s_i) = \Phi_{\text{R}}(\epsilon, s_i, s_j) = \Phi_{\text{C}}(s_i, \epsilon) = 0$.

We use the same span representations $\mathbf{g}$ from [42] and share them across the three tasks. We start by building bi-directional LSTMs [18] from word, character and ELMo [28] embeddings.

For a span s_i, its vector representation $\mathbf{g}_i$ is constructed by concatenating s_i's left and right end points from the BiLSTM outputs, an attention-based soft "headword," and embedded span width features. Hyperparameters and other implementation details will be described in Section 4.3.

4.2.4 Inference and Pruning

Following previous work, we use beam pruning to reduce the number of pairwise span factors from $O(n^4)$ to $O(n^2)$ at both training and test time, where n is the number of words in the document. We define two separate beams: B_{C} to prune spans for the coreference resolution task, and B_{R} for relation extraction. The spans in the beams are sorted by their span scores ϕ_{mc} and ϕ_{mr} respectively, and the sizes of the beams are limited by $\lambda_{\text{C}} n$ and $\lambda_{\text{R}} n$. We also limit the maximum width of spans to a fixed number W, which further reduces the number of span factors to $O(n)$.

4.3 Experimental Setup

We evaluate our unified framework SPANIE on SCIERC and SemEval 17.

4.3.1 Baselines

We compare our model with the following baselines on SCIERCdataset:

- **LSTM+CRF**: The state-of-the-art NER system [55], which applies CRF on top of LSTM for named entity tagging, the approach has also been used in scientific term extraction [36].

- **LSTM+CRF+ELMo**: LSTM+CRF with ELMO as an additional input feature.

- **E2E Rel**: State-of-the-art joint entity and relation extraction system [1] that has also been used in scientific literature [73, 43]. This system uses syntactic features such as part-of-speech tagging and dependency parsing.

- **E2E Rel(Pipeline)**: Pipeline setting of E2E Rel. Extract entities first and use entity results as input to relation extraction task.

- **E2E Rel+ELMo**: E2E Rel with ELMO as an additional input feature.

- **E2E Coref**: State-of-the-art coreference system [42] combined with ELMO. Our system SPANIE extends E2E Coref with multi-task learning.

In the SemEval task, we compare our model SPANIE with the best reported system in the SemEval leaderboard [73], which extends E2E Rel with several in-domain features such as gazetteers extracted from existing knowledge bases and model ensembles. We also compare with the state of the art on keyphrase extraction NN-CRF from Chapter 3, which applies semi-supervised methods to a neural tagging model.[3]

4.3.2 Implementation details

Our system extends the implementation and hyper-parameters from [42] with the following adjustments. We use a 1 layer BiLSTM with 200-dimensional hidden layers. All the FFNNs have 2 hidden layers of 150 dimensions each. We use 0.4 variational dropout [99] for the LSTMs, 0.4 dropout for the FFNNs, and 0.5 dropout for the input embeddings. We model spans up to 8 words. For beam pruning, we use $\lambda_{\mathrm{C}} = 0.3$ for coreference resolution and $\lambda_{\mathrm{R}} = 0.4$ for relation extraction.

[3] We compare with the inductive setting results.

	Dev			Test		
Model	P	R	F1	P	R	F1
LSTM+CRF	67.2	65.8	66.5	62.9	61.1	62.0
LSTM+CRF+ELMo	68.1	66.3	67.2	63.8	63.2	63.5
E2E Rel(Pipeline)	66.7	65.9	66.3	60.8	61.2	61.0
E2E Rel	64.3	68.6	66.4	60.6	61.9	61.2
E2E Rel+ELMo	67.5	66.3	66.9	63.5	63.9	63.7
SpanIE	70.0	66.3	**68.1**	67.2	61.5	**64.2**

(a) Entity recognition.

	Dev			Test		
Model	P	R	F1	P	R	F1
E2E Rel(Pipeline)	34.2	33.7	33.9	37.8	34.2	35.9
E2E Rel	37.3	33.5	35.3	37.1	32.2	34.1
E2E Rel+ELMo	38.5	36.4	37.4	38.4	34.9	36.6
SpanIE	45.4	34.9	**39.5**	47.6	33.5	**39.3**

(b) Relation extraction.

	Dev			Test		
Model	P	R	F1	P	R	F1
E2E Coref	59.4	52.0	55.4	60.9	37.3	46.2
SpanIE	61.5	54.8	**58.0**	52.0	44.9	**48.2**

(c) Coreference resolution.

Table 4.2: Comparison with previous systems on the development and test set for our three tasks. For coreference resolution, we report the average P/R/F1 of MUC, B^3, and $CEAF_{\phi_4}$ scores.

Task	Entity Rec.	Relation	Coref.
Multi Task (SPANIE)	68.1	39.5	58.0
Single Task	65.7	37.9	55.3
+Entity Rec.	-	38.9	57.1
+Relation	66.8	-	57.6
+Coreference	67.5	39.5	-

Table 4.3: Ablation study for multitask learning on SCIERC development set. Each column shows results for the target task.

4.4 Experimental Results

We evaluate SPANIE on SCIERC and SemEval 17 datasets.

4.4.1 IE Results

Results on SciERC Table 4.2 compares the result of our model with baselines on the three tasks: entity recognition (Table 4.2a), relation extraction (Table 4.2b), and coreference resolution (Table 4.2c). As evidenced by the table, our unified multi-task setup SPANIE outperforms all the baselines. For entity recognition, our model achieves 1.3% and 2.4% relative improvement over LSTM+CRF with and without ELMO, respectively. Moreover, it achieves 1.8% and 2.7% relative improvement over E2E Rel with and without ELMO, respectively. For relation extraction, we observe more significant improvement with 13.1% relative improvement over E2E Rel and 7.4% improvement over E2E Rel with ELMO. For coreference resolution, SPANIE outperforms E2E Coref with 4.5% relative improvement. We still observe a large gap between human-level performance and a machine learning system.

Ablations We evaluate the effect of multi-task learning in each of the three tasks defined in our dataset. Table 4.3 reports the results for individual tasks when additional tasks are included in the

	Span Indentification			Keyphrase Extraction			Relation Extraction			Overall		
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
NN-CRF	-	-	56.9	-	-	45.3	-	-	-	-	-	-
Best SemEval	55	54	55	44	43	44	36	23	**28**	44	41	43
SPANIE	62.2	55.4	**58.6**	48.5	43.8	**46.0**	40.4	21.2	27.8	48.1	41.8	**44.7**

Table 4.4: Results for scientific keyphrase extraction and extraction on SemEval 2017 Task 10, comparing with previous best systems.

learning objective function. We observe that performance improves with each added task in the objective. For example, entity recognition (65.7) benefits from both coreference resolution (67.5) and relation extraction (66.8). Relation extraction (37.9) significantly benefits when multi-tasked with coreference resolution (7.1% relative improvement). Coreference resolution benefits when multi-tasked with relation extraction, with a 4.9% relative improvement.

Results on SemEval 17 Table 4.4 compares the results of our model with the state of the art on the SemEval 17 dataset for tasks of span identification, keyphrase extraction and relation extraction as well as the overall score. Span identification aims at identifying spans of entities. Keyphrase classification and relation extraction has the same setting with the entity and relation extraction in SCIERC. Our model outperforms all the previous models that use hand-designed features. We observe more significant improvement in span identification than keyphrase classification. This confirms the benefit of our model in enumerating spans (rather than BIO tagging in state-of-the-art systems). Moreover, we have competitive results compared to the previous state of the art in relation extraction. We observe less gain compared to the SCIERC dataset mainly because there are no coference links, and the relation types are not comprehensive.

4.5 Related Work

Different from most previous IE systems for scientific literature and general domains [1, 2, 3, 4, 5, 6], which use preprocessed syntactic, discourse or coreference features as input, our unified framework does not rely on any pipeline processing and is able to model overlapping spans.

While [37] show improvements by jointly modeling entities, relations, and coreference links, most recent neural models for these tasks focus on single tasks [59, 100, 42, 55, 3] or joint entity and relation extraction [101, 14, 6, 102]. Among those studies, many papers assume the entity boundaries are given, such as [59], [6] and [3]. Our work relaxes this constraint and predicts entity boundaries by optimizing over all possible spans. Our model draws from recent end-to-end span-based models for coreference resolution [42, 103] and semantic role labeling [47] and extends them for the multi-task framework involving the three tasks of identification of entity, relation and coreference.

Neural multi-task learning has been applied to a range of NLP tasks. Most of these models share word-level representations [60, 104, 41, 105], while [3] uses high-order cross-task factors. Our model instead propagates cross-task information via span representations, which is related to [106].

4.6 Conclusion

In this chapter, we develop a multi-task model for identifying entities, relations, and coreference clusters in scientific articles. By sharing span representations and leveraging cross-sentence information, our multi-task setup effectively improves performance across all tasks. Moreover, we show that our multi-task model is better at predicting span boundaries and outperforms previous state-of-the-art scientific IE systems on entity and relation extraction, without using any hand-engineered features or pipeline processing.

We still observe a large gap between the performance of our model and human performance, confirming the challenges of scientific IE. Future work includes improving the performance using semi-supervised techniques and providing in-domain features.

Chapter 5

A GENERAL FRAMEWORK FOR INFORMATION EXTRACTION USING DYNAMIC SPAN GRAPHS

In the previous chapter, we introduced SPANIE, a unified learning model for extracting scientific entities, relations, and coreference resolution. SPANIE is able to model the interaction between different tasks and avoid cascading errors by span-based modeling. However, it mostly relies on the first layer LSTM to share span representations between different tasks; there is limited sharing of document context information. In this chapter, we improve SPANIE by directly modeling the information flow across different tasks through dynamic span graphs (DYGIE). DYGIE couples multiple information extraction tasks through shared span representations which are refined leveraging contextualized information from relations and coreferences. The graphs are constructed by selecting the most confident entity spans and linking these nodes with confidence-weighted relation types and coreferences. The nodes in the 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 the multi-task method in Chapter 4 that only shares span representations from the local context [40], our framework leverages rich contextual span representations by propagating information through coreference and relation links. The dynamic span graph allows coreference and relation type confidences to propagate through the graph to iteratively refine the span representations. DYGIE significantly outperforms the state-of-the-art on multiple information extraction tasks across multiple datasets reflecting different domains.

We evaluate DYGIE on several datasets spanning many domains (including news, scientific articles, and wet lab experimental protocols), achieving state-of-the-art performance across all tasks and domains and demonstrating the value of coupling related tasks to learn richer span representations. For example, DYGIE achieves relative improvements of 5.7% and 9.9% over state of the

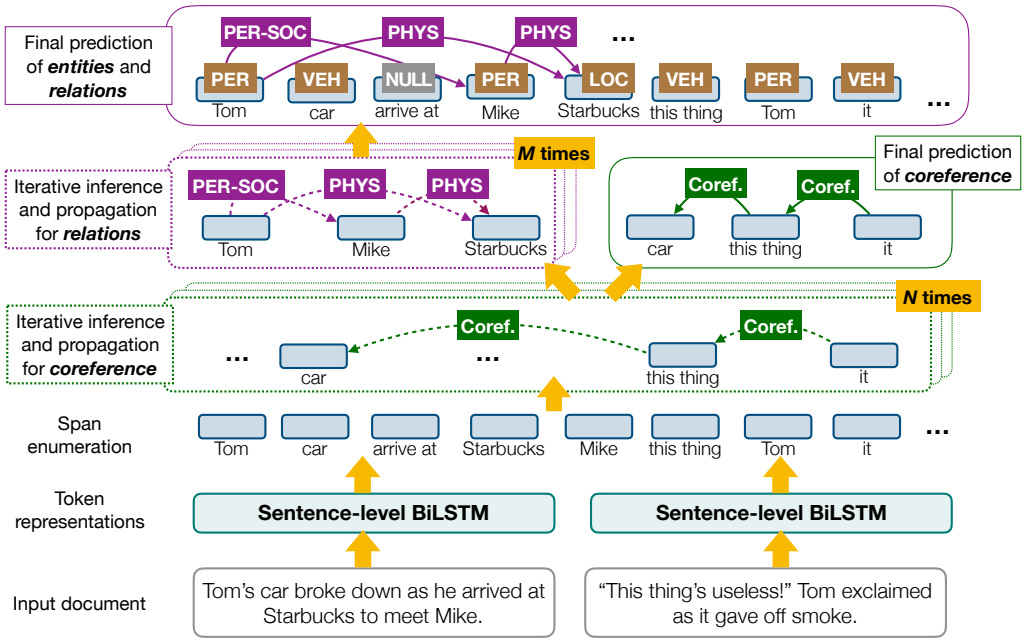


Figure 5.1: Overview of our DYGIE model. Dotted arcs indicate confidence weighted graph edges. Solid lines indicate the final predictions.

art on the ACE05 entity and relation extraction tasks, and an 11.3% relative improvement on the ACE05 overlapping entity extraction task. [1]

5.1 Model

Problem Definition The basic problem definition is the same as Chapter 4. The input is a document represented as a sequence of words D, from which we derive $S = \{s_1, \ldots, s_T\}$, the set of all possible within-sentence word sequence spans (up to length L) in the document. The output contains three structures: the entity types E for all spans S, the relations R for all span pairs $S \times S$ within the same sentence, and the coreference links C for all spans in S across sentences. However in this chapter Entity Recognition and Relation Recognition are the primary tasks. Coreference resolution is an auxiliary task where we predict the best antecedent c_i for each span s_i.

[1] This chapter is primarily describing work that has been published in [107]. My contribution involves model design, running experiments and writing.

Model We develop a general information extraction framework (DyGIE) to identify and classify entities, relations, and coreference in a multi-task setup. DyGIE first enumerates all text spans in each sentence, and computes a locally-contextualized vector space representation of each span. The model then employs a *dynamic span graph* to incorporate global information into its span representations, as follows. At each iteration, the model identifies the text spans that are most likely to represent entities, and treats these spans as nodes in a graph structure. It constructs confidence-weighted arcs for each node according to its predicted coreference and relation links with the other nodes in the graph. Then, the span representations are refined using broader context from gated updates propagated from neighboring relation types and co-referred entities. These refined span representations are used in a multi-task framework to predict entity types, relation types, and coreference links.

5.1.1 Model Architecture

In this section, we give an overview of the main components and layers of the DyGIE framework, as illustrated in Figure 5.1. Details of the graph construction and refinement process will be presented in the next section.

Token Representation Layer We apply a bidirectional LSTM over the input tokens. The input for each token is a concatenation of the character reprensetation, GLoVe [23] word embeddings, and ELMo embeddings [28]. The output token representations are obtained by stacking the forward and backward LSTM hidden states.

Span Representation Layer For each span s_i, its initial vector representation $\mathbf{g}_i^0$ is obtained by concatenating BiLSTM outputs at the left and right end points of s_i, an attention-based soft "headword," and an embedded span width feature, following [42].

Coreference Propagation Layer The propagation process starts from the span representations $\mathbf{g}_i^0$. At each iteration t, we first compute an *update vector* $\mathbf{u}_C^t(i)$ for each span s_i. Then we use

$\mathbf{u}_C^t(i)$ to update the current representation $\mathbf{g}_i^t$, producing the next span representation $\mathbf{g}_i^{t+1}$. By repeating this process N times, the final span representations $\mathbf{g}_i^N$ share contextual information across spans that are likely to be antecedents in the coreference graph, similar to the process in [103].

Relation Propagation Layer The outputs $\mathbf{g}_i^N$ from the coreference propagation layer are passed as inputs to the relation propagation layer. Similar to the coreference propagation process, at each iteration t, we first compute the update vectors $\mathbf{u}_R^t$ for each span s_i, then use it to compute $\mathbf{g}_i^{t+1}$. Information can be integrated from multiple relation paths by repeating this process M times.

Final Prediction Layer We use the outputs of the relation graph layer $\mathbf{g}_i^{N+M}$ to predict the entity labels E and relation labels R. For entities, we pass $\mathbf{g}_i^{N+M}$ to a feed-forward network (FFNN) to produce per-class scores $\mathbf{P}_E(i)$ for span s_i. For relations, we pass the concatenation of $\mathbf{g}_i^{N+M}$ and $\mathbf{g}_j^{N+M}$ to a FFNN to produce per-class relation scores $\mathbf{P}_R(i,j)$ between spans s_i and s_j. Entity and relation scores are normalized across the label space, as in Chapter 4. For coreference, the scores between span pairs (s_i, s_j) are computed from the coreference graph layer outputs $(\mathbf{g}_i^N, \mathbf{g}_j^N)$, and then normalized across all possible antecedents, similar to [103].

5.1.2 Dynamic Graph Construction and Span Refinement

The dynamic span graph facilitates propagating broader contexts through soft coreference and relation links to refine span representations. The nodes in the graph are spans s_i with vector representations $\mathbf{g}_i^t \in \mathbb{R}^d$ for the t-th iteration. The edges are weighted by the coreference and relation scores, which are iteratively updated with the span representations. In this section, we explain how coreference and relation links can update span representations and link scores.

Coreference Propagation Similar to Chapter 4, we define a beam B_C consisting of b_c spans that are most likely to be in a coreference chain. We consider $\mathbf{P}_C^t$ to be a matrix of real values that indicate coreference confidence scores between these spans at the t-th iteration. $\mathbf{P}_C^t$ is of size

$b_c \times K$, where K is the maximum number of antecedents considered. For the coreference graph, an edge in the graph is single directional, connecting the current span s_i with all its potential antecedents s_j in the coreference beam, where $j < i$. The edge between s_i and s_j is weighted by coreference confidence score at the current iteration $P_C^t(i,j)$. The span update vector $\mathbf{u}_C^t(i) \in \mathbb{R}^d$ is computed by aggregating the neighboring span representations $\mathbf{g}_j^t$, weighted by their coreference scores $P_C^t(i,j)$:

$$\mathbf{u}_C^t(i) = \sum_{j \in B_C(i)} P_C^t(i,j)\mathbf{g}_j^t \tag{5.1}$$

where $B_C(i)$ is the set of K spans that are antecedents of s_i,

$$P_C^t(i,j) = \frac{\exp(V_C^t(i,j))}{\sum_{j' \in B_C(i)} \exp(V_C^t(i,j'))} \tag{5.2}$$

$V_C^t(i,j)$ is a scalar score computed by concatenating the span representations $[\mathbf{g}_i^t, \mathbf{g}_j^t, \mathbf{g}_i^t \odot \mathbf{g}_j^t]$, where $\odot$ is element-wise multiplication. The concatenated vector is then fed as input to a FFNN, similar to [103].

Relation Propagation For each sentence, we define a beam B_R consisting of b_r entity spans that are mostly likely to be involved in a relation. Unlike the coreference graph, the relation edges are associated with a vector of weights that capture different relation types. Therefore, for the t-th iteration, we use a tensor $\mathbf{V}_R^t \in \mathbb{R}^{b_R \times b_R \times L_R}$ to capture scores of each of the L_R relation types. In other words, each edge in the relation graph connects two entity spans s_i and s_j in the relation beam B_R. $\mathbf{V}_R^t(i,j)$ is a L_R-length vector of relation scores, computed with a FFNN with $[\mathbf{g}_i^t, \mathbf{g}_j^t]$ as the input. The relation update vector $\mathbf{u}_R^t(i) \in \mathbb{R}^d$ is computed by aggregating neighboring span representations on the relation graph:

$$\mathbf{u}_R^t(i) = \sum_{j \in B_R} f(\mathbf{V}_R^t(i,j))\mathbf{A}_R \odot \mathbf{g}_j^t, \tag{5.3}$$

where $\mathbf{A}_R \in \mathbb{R}^{L_R \times d}$ is a trainable linear projection matrix, f is a non-linear function to select the most important relations. Because only a small number of entities in the relation beam are actually linked to the target span, propagation among all possible span pairs would introduce too

much noise to the new representation. Therefore, we choose f to be the ReLU function to remove the effect of unlikely relations by setting the all negative relation scores to 0. Unlike coreference connections, two spans linked via a relation are not expected to have similar representations, so the matrix $\mathbf{A}_R$ helps to transform the embedding $\mathbf{g}_j^t$ according to each relation type.

Updating Span Representations with Gating To compute the span representations for the next iteration $t \in \{1, \ldots, N+M\}$, we define a gating vector $\mathbf{f}_x^t(i) \in \mathbb{R}^d$, where $x \in \{C, R\}$, to determine whether to keep the previous span representation $\mathbf{g}_i^t$ or to integrate new information from the coreference or relation update vectors $\mathbf{u}_x^t(i)$. Formally,

$$\begin{aligned} \mathbf{f}_x^t(i) &= g(\mathbf{W}_x^{\mathrm{f}}[\mathbf{g}_i^t, \mathbf{u}_x^t(i)]) \\ \mathbf{g}_i^{t+1} &= \mathbf{f}_x^t(i) \odot \mathbf{g}_i^t + (1 - \mathbf{f}_x^t(i)) \odot \mathbf{u}_x^t(i), \end{aligned} \tag{5.4}$$

where $\mathbf{W}_x^{\mathrm{f}} \in \mathbb{R}^{d \times 2d}$ are trainable parameters, and g is an element-wise sigmoid function. The update of $\mathbf{g}_i$ can involve different uses of coreference and relations including only $x = C$, only $x = R$ or sequential application of $x = C$ for N iteration followed by $X = R$ for M iterations (or vice versa).

5.1.3 Training

The loss function is defined as a weighted sum of the log-likelihood of all three tasks:

$$\begin{aligned} \sum_{(D, R^*, E^*, C^*) \in \mathcal{D}} \Big\{ & \lambda_{\mathrm{E}} \log P(E^* \mid C, R, D) \\ & + \lambda_{\mathrm{R}} \log P(R^* \mid C, D) + \lambda_{\mathrm{C}} \log P(C^* \mid D) \Big\} \end{aligned} \tag{5.5}$$

where E^*, R^* and C^* are gold structures of the entity types, relations and coreference, respectively. $\mathcal{D}$ is the collection of all training documents D. The task weights λ_{E}, λ_{R}, and λ_{C} are hyper-parameters to control the importance of each task.

We use a 1 layer BiLSTM with 200-dimensional hidden layers. All the FFNN functions have 2 hidden layers of 150 dimensions each. We use 0.4 variational dropout [99] for the LSTMs, 0.4

	Domain	Docs	Ent	Rel	Coref
ACE04	News	348	7	7	✓
ACE05	News	511	7	6	✗
SciERC	AI	500	6	7	✓
WLP	Bio lab	622	18	13	✗

Table 5.1: Datasets for joint entity and relation extraction and their statistics. *Ent*: Number of entity categories. *Rel*: Number of relation categories.

dropout for the FFNNs, and 0.5 dropout for the input embeddings. The hidden layer dimensions and dropout rates are chosen based on the development set performance in multiple domains. The task weights, learning rate, maximum span length, number of propagation iterations and beam size are tuned specifically for each dataset using development data.

5.2 Experiments

DyGIE is a general IE framework that can be applied to multiple tasks. We evaluate the performance of DyGIE against models from two lines of work: combined entity and relation extraction, and overlapping entity extraction.

5.2.1 Entity and relation extraction

For the entity and relation extraction task, we test the performance of DyGIE on four different datasets: ACE2004, ACE2005, SciERC and the Wet Lab Protocol Corpus. We include the relation graph propagation layer in our models for all datasets. We include the coreference graph propagation layer on the data sets that have coreference annotations available.

Data All four data sets are annotated with entity and relation labels. Only a small fraction of entities ($< 3\%$ of total) in these data sets have a text span that overlaps the span of another entity. Statistics on all four data sets are displayed in Table 5.1.

The **ACE2004** and **ACE2005** corpora provide entity and relation labels for a collection of documents from a variety of domains, such as newswire and online forums. We use the same entity and relation types, data splits, and preprocessing as [1] and [12]. Following the convention established in this line of work, an entity prediction is considered correct if its type label and head region match those of a gold entity. We will refer to this version of the ACE2004 and ACE2005 data as ACE04 and ACE05. Since the domain and mention span annotations in the ACE datasets are very similar to those of OntoNotes [108], and OntoNotes contains significantly more documents with coreference annotations, we use OntoNotes to train the parameters for the auxiliary coreference task. The OntoNotes corpus contains 3493 documents, averaging roughly 450 words in length.

The **SciERC** corpus [40] provides entity, coreference and relation annotations for a collection of documents from 500 AI paper abstracts. The dataset defines scientific term types and relation types specially designed for AI domain knowledge graph construction. An entity prediction is considered correct if its label and span match with a gold entity.

The **Wet Lab Protocol Corpus (WLPC)** provides entity, relation, and event annotations for 622 wet lab protocols [109]. A wet lab protocol is a series of instructions specifying how to perform a biological experiment. Following the procedure in [109], we perform entity recognition on the union of entity tags and event trigger tags, and relation extraction on the union of entity-entity relations and entity-trigger event roles. Coreference annotations are not available for this dataset.

Baselines We compare DyGIE with current state of the art methods in different datasets. Miwa & Bansal [1] (abbreviated M&B in the table) provide the current state of the art on ACE04. They construct a Tree LSTM using dependency parse information, and use the representations learned by the tree structure as features for relation classification. Bekoulis et al. [110] (Bekoulis18) use adversarial training as regularization for a neural model. Zhang et al. [14] (Zhang17) cast joint entity and relation extraction as a table filling problem and build a globally optimized neural model incorporating syntactic representations from a dependency parser. Similar to DyGIE, Sanh et al. [111] (Sanh18) and SpanIE use a multi-task learning framework for extracting entity, relation and coreference labels. Sanh18 improved the state of the art on ACE05 using multi-task, hierarchical

Dataset	System	Entity	Relation
ACE04	Bekoulis18	81.6	47.5
	M&B	81.8	48.4
	DyGIE	**87.4**	**59.7**
ACE05	M&B	83.4	55.6
	Zhang17	83.6	57.5
	Sanh18	87.5	62.7
	DyGIE	**88.4**	**63.2**
SciERC	Luan18	64.2	39.3
	DyGIE	**65.2**	**41.6**
WLPC	Kulkarni18	78.0	*54.9
	DyGIE	**79.5**	**64.1**

Table 5.2: F1 scores on the joint entity and relation extraction task on each test set, compared against the previous best systems. * indicates relation extraction system that takes gold entity boundary as input.

supervised training with a set of low level tasks at the bottom layers of the model and more complex tasks at the top layers of the model. SpanIE previously achieved the state of the art on SciERC and use a span-based neural model like our DyGIE. Kulkarni et al. [109] (Kulkarni18) provide a baseline for the WLPC data set. They employ an LSTM-CRF for entity recognition, following [55]. For relation extraction, they assume the presence of gold entities and train a maximum-entropy classifier using features from the labeled entities.

Results Table 5.2 shows test set F1 on the joint entity and relation extraction task. We observe that DyGIE achieves substantial improvements on both entity recognition and relation extraction across the four data sets and three domains, all in the realistic setting where no "gold" entity labels are supplied at test time. DyGIE achieves 7.1% and 7.0% relative improvements over the state of the art on NER for ACE04 and ACE05, respectively. For the relation extraction task, DyGIE

attains 25.8% relative improvement over SOTA on ACE04 and 13.7% relative improvement on ACE05. For ACE05, the best entity extraction performance is obtained by switching the order between `CorefProp` and `RelProp` (`RelProp` first then `CorefProp`).

On SciERC, DyGIE advances the state of the art by 5.9% and 1.9% for relation extraction and NER, respectively. The improvement of DyGIE over the previous SciERC model underscores the ability of coreference and relation propagation to construct rich contextualized representations.

The results from [109] establish a baseline for IE on the WLPC data. In that work, relation extraction is performed using gold entity boundaries as input. Without using any gold entity information, DyGIE improves on the baselines by 16.8% for relation extraction and 2.2% for NER. The gain for NER maybe smaller than on other tasks because `CorefProp` is not used in this dataset (no suitable coreference annotation can be found as auxiliary task).

On the OntoNotes data set used for the auxiliary coreference task with ACE05, our model achieves coreference test set performance of 70.4 F1, which is competitive with the state-of-the-art performance reported in [42].

5.2.2 Overlapping Entity Extraction

There are many applications where the correct identification of overlapping entities is crucial for correct document understanding. For instance, in the biomedical domain, a *BRCA1 mutation carrier* could refer to a patient taking part in a clinical trial, while *BRCA1* is the name of a gene.

We evaluate the performance of DyGIE on overlapping entity extraction in three datasets: ACE2004, ACE2005 and GENIA. Since relation annotations are not available for these datasets, we include the coreference propagation layer in our models but not the relation layer.[2]

Data Statistics on our three datasets are listed in Table 5.3. All three have a substantial number ($> 20\%$ of total) of overlapping entities, making them appropriate for this task.

As in the joint case, we evaluate our model on **ACE2004** and **ACE2005**, but here we follow the same data preprocessing and evaluation scheme in Wang et al. [49]. We refer to these data sets

[2]We use the pre-processed ACE dataset from previous work, and relation annotation is not available.

	Domain	Docs	Ent	Overlap	Coref
ACE04-O	News	443	7	42%	✓
ACE05-O	News	437	7	32%	✗
GENIA	Biomed	1999	5	24%	✓

Table 5.3: Datasets for overlapping entity extraction and their statistics. *Ent*: Number of entity categories. *Overlap*: Percentage of sentences that contain overlapping entities.

as ACE04-O and ACE05-O. Unlike the combined joint entity and relation task in Sec. 5.2.1, where only the entity head span need be predicted, an entity prediction is considered correct in these experiments if both its entity label and its full text span match a gold prediction. This is a more stringent evaluation criterion than the one used in Section 5.2.1. As before, we use the OntoNotes annotations to train the parameters of the coreference layer.

The **GENIA** corpus [112] provides entity tags and coreferences for 1999 abstracts from the biomedical research literature. We only use the IDENT label to extract coreference clusters. We use the same data set split and preprocessing procedure as [49] for overlapping entity recognition.

Baselines The current state-of-the-art approach on all three data sets is Wang et al. [49](Wang18), which uses a segmental hypergraph coupled with neural networks for feature learning. Katiyar et al. [48](Katiyar18) also propose a hypergraph approach using a recurrent neural network as a feature extractor.

Results Table 5.4 presents the results of our overlapping entity extraction experiments on the different datsets. DyGIE improves 11.6% absolute on the state of the art for ACE04-O and 11.3% for ACE05-O. DyGIE also advances the state of the art on GENIA, albeit by a more modest 1.5%. Together these results suggest that DyGIE can be utilized fruitfully for information extraction across different domains with overlapped entities, such as bio-medicine.

Dataset	System	Entity F1
ACE04-O	Katiyar18	72.7
	Wang18	75.1
	DyGIE	**84.7**
ACE05-O	Katiyar18	70.5
	Wang18	74.5
	DyGIE	**82.9**
GENIA	Katiyar18	73.8
	Wang18	75.1
	DyGIE	**76.2**

Table 5.4: Performance on the overlapping entity extraction task, compared to previous best systems. We report F1 of extracted entities on the test sets.

5.3 Analysis of Graph Propagation

We use the dev sets of ACE2005 and SciERC to analyze the effect of different model components.

5.3.1 Coreference and Relation Graph Layers

Tables 5.5 and 5.6 show the effects of graph propagation on entity and relation prediction accuracy, where $-$`CorefProp` and $-$`RelProp` denote analyzing the propagation process by setting $N = 0$ or $M = 0$, respectively. `Base` is the base model without any propagation. For ACE05, we observe that coreference propagation is mainly helpful for entities; it appears to hurt relation extraction. On SciIE, coreference propagation gives a small benefit on both tasks. Relation propagation significantly benefits both entity and relation extraction in both domains. In particular, there are a large portion of sentences with multiple relation instances across different entities in both ACE05 and SciERC, which is the scenario in which we expect relation propagation to help.

Since coreference propagation has more effect on entity extraction and relation propagation has more effect on relation extraction, we mainly focus on ablating the effect of coreference propaga-

	Entity			Relation		
Model	P	R	F1	P	R	F1
DyGIE	87.4	86.7	**87.1**	56.2	60.9	58.4
−CorefProp	86.2	85.2	85.7	64.3	56.7	**60.2**
−RelProp	87.0	86.7	86.9	60.4	55.8	58.0
Base	86.1	85.7	85.9	59.5	55.7	57.6

Table 5.5: Ablations on the ACE05 development set with different graph propagation setups. −CorefProp ablates the coreference propagation layers, while −RelProp ablates the relation propagation layers. Base is the system without any propagation.

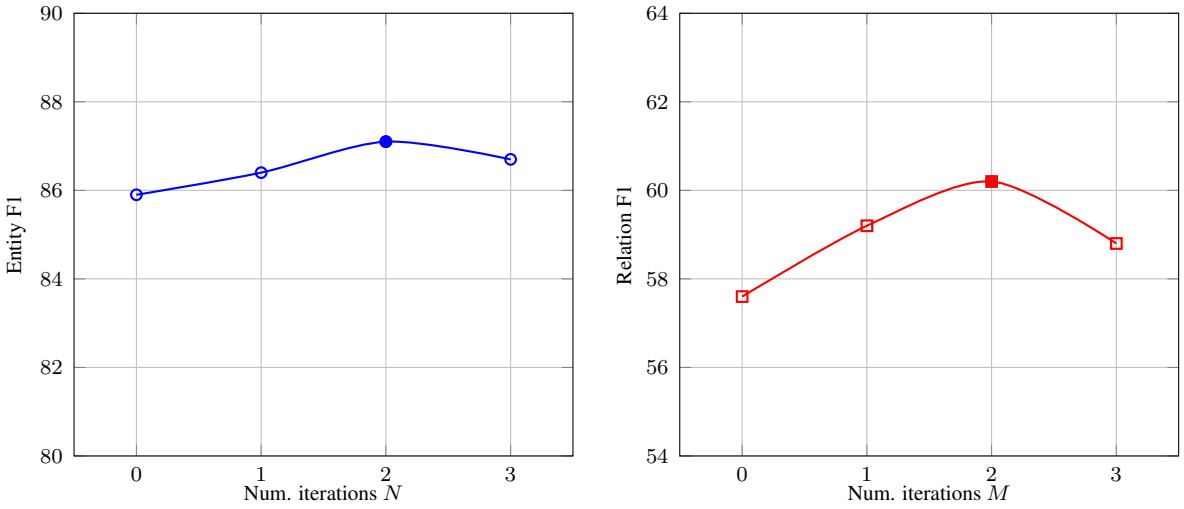
	Entity			Relation		
Model	P	R	F1	P	R	F1
DyGIE	68.6	67.8	**68.2**	46.2	38.5	**42.0**
−CorefProp	69.2	66.9	68.0	42.0	40.5	41.2
−RelProp	69.1	66.0	67.5	43.6	37.6	40.4
Base	70.0	66.3	68.1	45.4	34.9	39.5

Table 5.6: Ablations on the SciERC development set on different graph progation setups. CorefProp has a much smaller effect on entity F1 compared to ACE05.

tion on entity extraction and relation propagation on relation extraction in the following subsections.

5.3.2 Coreference Propagation and Entities

A major challenge of ACE05 is to disambiguate the entity class for pronominal mentions, which requires reasoning with cross-sentence contexts. For example, in a sentence from ACE05 dataset, "One of **[them]**$_{\text{PER}}$, from a very close friend of **[ours]**$_{\text{ORG}}$." It is impossible to identity whether *them* and *ours* is a person (*PER*) or organization (*ORG*) unless we have read previous sentences.



(a) Entity F1 with different number of `CorefProp` iterations N.

(b) Relation F1 with different number of `RelProp` iterations M.

Figure 5.2: F1 score of each layer on ACE development set for different number of iterations. $N = 0$ or $M = 0$ indicates no propagation is made for the layer.

We hypothesize that this is a context where coreference propagation can help. Table 5.7 shows the effect of the coreference layer for entity categorization of pronouns.[3] DyGIE has 6.6% improvement on pronoun performance, confirming our hypothesis.

Looking further, Table 5.8 shows the impact on all entity categories, giving the difference between the confusion matrix entries with and without `CorefProp`. The frequent confusions associated with pronouns (*GPE/PER* and *PER/ORG*, where *GPE* is a geopolitical entity) greatly improve, but the benefit of `CorefProp` extends to most categories.

Of course, there are a few instances where `CorefProp` causes errors in entity extraction. For example, in the sentence "[They]$^{\text{ORG}}_{\text{PER}}$ might have been using Northshore...", DyGIE predicted *They* to be of *ORG* type because the most confident antecedent is *those companies* in the previous sentence: "The money was invested in *those companies*." However, *They* is actually referring to

[3]Pronouns included: `anyone, everyone, it, itself, one, our, ours, their, theirs, them, themselves, they, us, we, who`

Entity Perf. on Pronouns	P	R	F1
DyGIE	79.0	77.1	**78.0**
DyGIE−CorefProp	73.8	72.6	73.2

Table 5.7: Entity extraction performance on pronouns in ACE05. CorefProp significantly increases entity extraction F1 on hard-to-disambiguate pronouns by allowing the model to leverage cross-sentence contexts.

these fund managers earlier in the document, which belongs to *PER* category.

In the SciERC dataset, the pronouns are uniformly assigned with a *Generic* label, which explains why CorefProp does not have much effect on entity extraction performance.

The Figure 5.2a shows the effect of number of iterations for coreference propagation in the entity extraction task. The figure shows that coreference layer obtains the best performance on the second iteration ($N = 2$).

5.3.3 Relation Propagation Impact

Figure 5.3 shows relation scores as a function of number of entities in sentence for DyGIE and DyGIE without relation propagation on ACE05. The figure indicates that relation propagation achieves significant improvement in sentences with more entities, where one might expect that using broader context could have more impact.

Figure 5.2b shows the effect of number of iterations for relation propagation in the relation extraction task. Our model achieves the best performance on the second iteration ($M = 2$).

5.4 Related Work

Neural graph-based models have achieved significant improvements over traditional feature-based approaches on several graph modeling tasks. Knowledge graph completion [113, 114] is one prominent example. For relation extraction tasks, graphs have been used primarily as a means

	LOC	WEA	GPE	PER	FAC	ORG	VEH
LOC	5	0	-2	-1	2	-1	0
WEA	0	3	0	0	1	-3	-1
GPE	-3	0	31	**-26**	3	-7	0
PER	0	-2	-3	18	-1	**-26**	4
FAC	4	-1	2	-3	2	-5	1
ORG	0	0	0	-8	-1	6	0
VEH	0	-2	-1	2	5	-1	1

Table 5.8: Difference in the confusion matrix counts for ACE05 entity extraction associated with adding `CorefProp`.

to incorporate pipelined features such as syntactic or discourse relations [3, 115, 116]. [117] models all possible paths between entities as a graph, and refines pair-wise embeddings by performing a walk on the graph structure. All these previous works assume that the nodes of the graph (i.e. the entity candidates to be considered during relation extraction) are predefined and fixed throughout the learning process. On the other hand, our framework does not require a fixed set of entity boundaries as an input for graph construction. Motivated by state-of-the-art span-based approaches to coreference resolution [42, 103] and semantic role labeling [47], the model uses a beam pruning strategy to dynamically select high-quality spans, and constructs a graph using the selected spans as nodes.

To make the model more general, we combine the multitask learning framework with ELMo embeddings [28] without relying on external syntactic tools and risking the cascading errors that accompany them, and improve the interaction between tasks through dynamic graph propagation. While the performance of DyGIE benefits from ELMo, it advances over some systems [40, 111] that also incorporate ELMo. The analyses presented here give insights into the benefits of joint

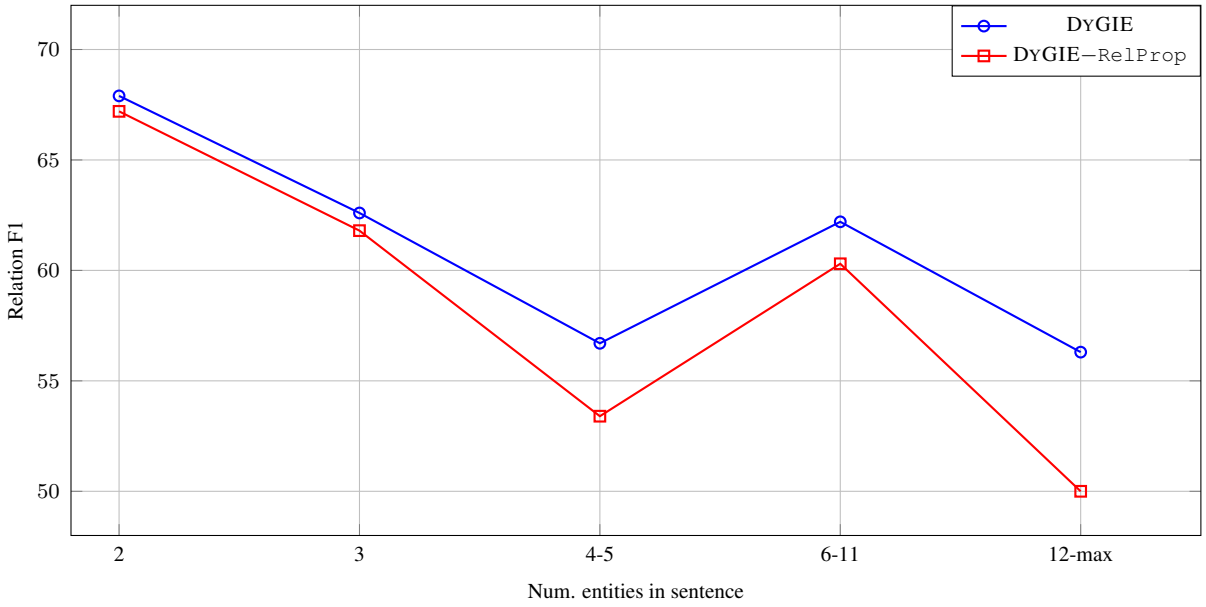


Figure 5.3: Relation F1 broken down by number of entities in each sentence. The performance of relation extraction degrades on sentences containing more entities. Adding relation propagation alleviates this problem.

modeling.

5.5 Conclusion

We have introduced DyGIE as a general information extraction framework, and have demonstrated that our system achieves state-of-the art results on entity recognition and relation extraction tasks across a diverse range of domains. The key contribution of this chapter is the dynamic span graph approach, which enhance interaction across tasks that allows the model to learn useful information from broader context. Unlike many IE frameworks, SpanIE does not require any preprocessing using syntactic tools, and has significant improvement across different IE tasks including entity, relation extraction and overlapping entity extraction. The addition of co-reference and relation propagation across sentences adds only a small computation cost to inference; the memory cost is controlled by beam search. These added costs are small relative to those of the baseline span-based model.

Chapter 6

SCIENTIFIC KNOWLEDGE GRAPH CONSTRUCTION

As scientific communities grow and evolve, new tasks, methods, and datasets are introduced and different methods are compared with each other. Despite advances in search engines, it is still hard to identify new technologies and their relationships with what existed before. To help researchers more quickly identify opportunities for new combinations of tasks, methods and data, it is important to design intelligent algorithms that can both extract and organize scientific information from a large collection of documents.

Organizing scientific information into structured knowledge bases requires IE about scientific entities and their relationships. However, the challenges associated with scientific IE are greater than for a general domain. First, annotation of scientific text requires domain expertise which makes annotation costly and limits resources. In addition, most relation extraction systems are designed for within-sentence relations. However, extracting information from scientific articles requires extracting relations across sentences. In Chapter 3 - 5, we develop IE systems that advance state of the art in extracting entity, relation and coreference, leveraging cross-sentence context. In this chapter, we apply SPANIE and DYGIE to the task of organizing extracted information, specifically constructing a knowledge graph for scientific papers.

To explore this problem, we used our dataset SCIERC, which includes annotations of scientific terms, relation categories and co-reference links. We then build a scientific knowledge graph integrating terms and relations extracted from each article, where the extracted entities and relations are propagated through coreference links. A good portion of cross-sentence relations can be extracted in this way. Human evaluation shows that propagating relations through coreference links can significantly improve the quality of the automatic constructed knowledge graph.

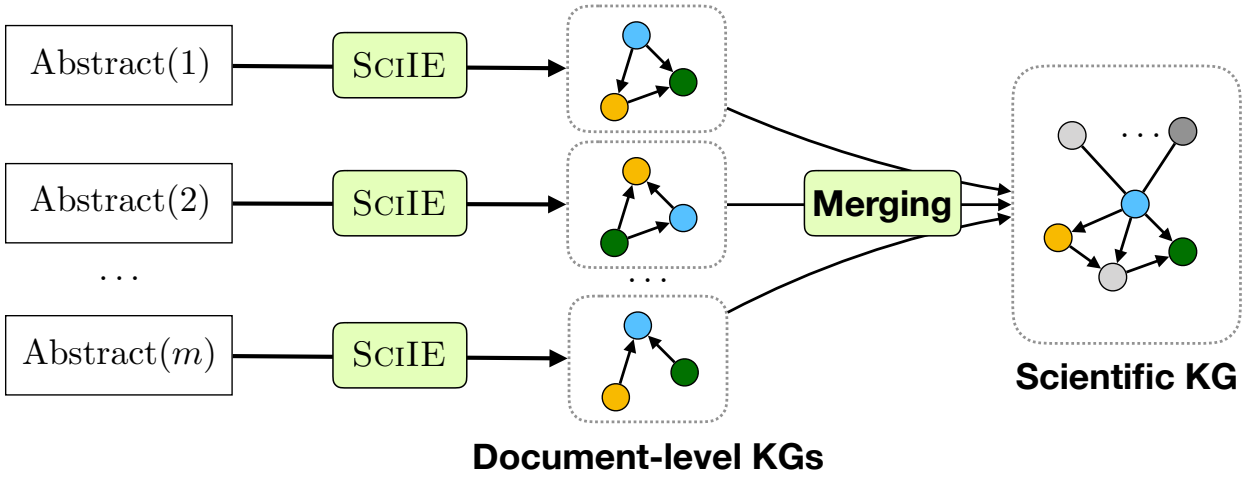


Figure 6.1: Knowledge graph construction process.

6.1 Knowledge Graph Construction

We construct a scientific knowledge graph from a large corpus of scientific articles. The corpus includes all abstracts (110k in total) from 12 AI conference proceedings from the Semantic Scholar Corpus and are automatically annotated by SpanIE trained on SciERC dataset. Nodes in the knowledge graph correspond to scientific entity clusters. Edges correspond to scientific relations between pairs of entities. The edges are typed according to the relation types in SciERC. In order to construct the knowledge graph for the whole corpus, we first apply the DyGIE model over single documents and then integrate the entities and relations across multiple documents (Figure 6.1). Figure 6.2 shows a part of a knowledge graph created by our method. For example, *Statistical Machine Translation (SMT)* and *grammatical error correction* are nodes in the graph, and they are connected through a *Used-for* relation type.

Extracting nodes (entities) The SpanIE model extracts entities, their relations, and coreference clusters within a document for each document in the collection. Extracted phrases are heuristically normalized using entities and coreference links. We replace all acronyms with their corresponding full name and normalize all the plural terms with their singular counterparts. Next, we link all enti-

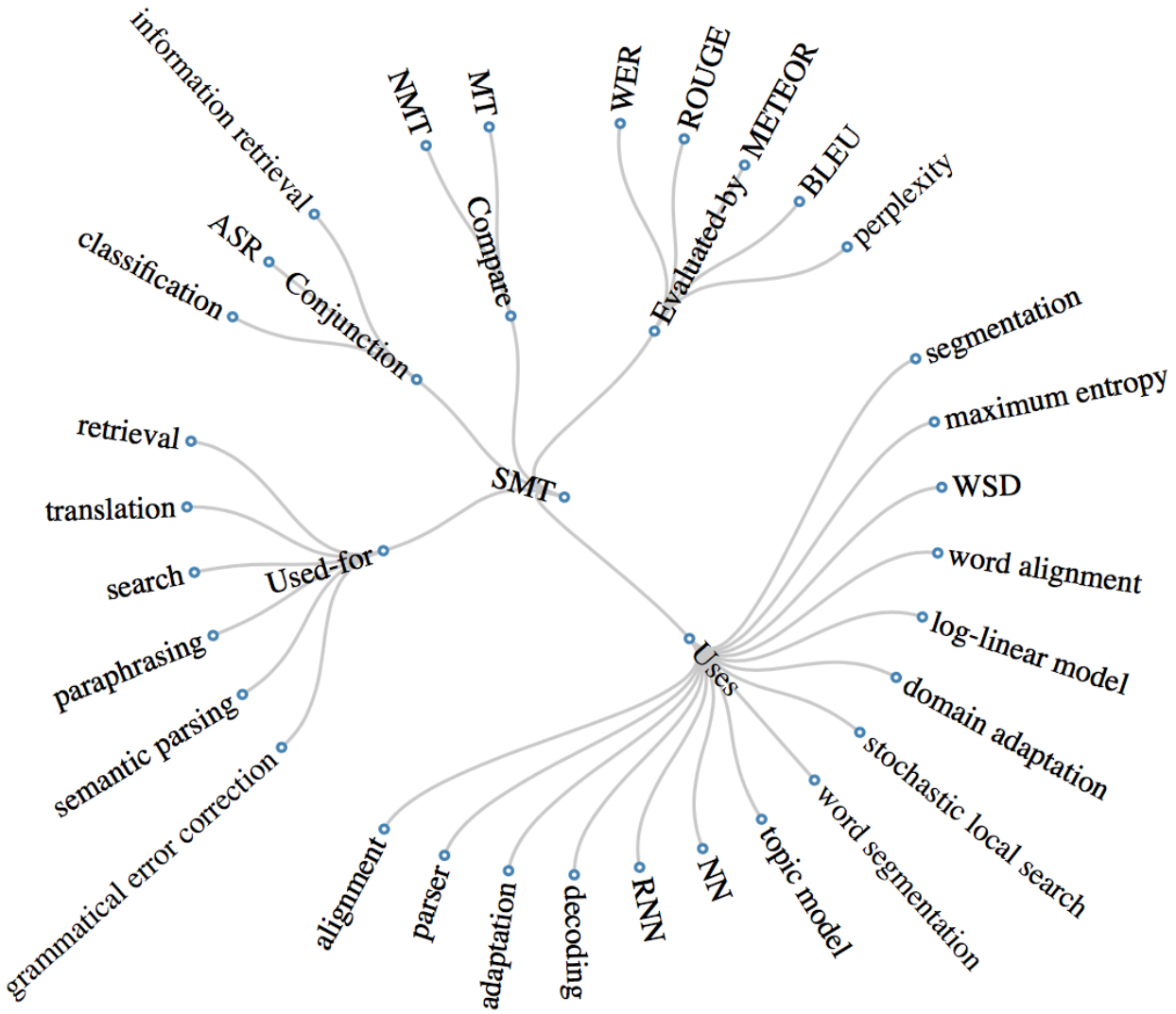


Figure 6.2: A part of an automatically constructed scientific knowledge graph with the most frequent neighbors of the scientific term *statistical machine translation (SMT)* on the graph. For simplicity we denote *Used-for (Reverse)* as *Uses*, *Evaluated-for (Reverse)* as *Evaluated-by*, and replace common terms with their acronyms. The original graph is given in Appendix B.

ties that belong to the same coreference cluster to replace generic terms with any other non-generic term in the cluster, we replace all the entities in the cluster with the entity that has the longest

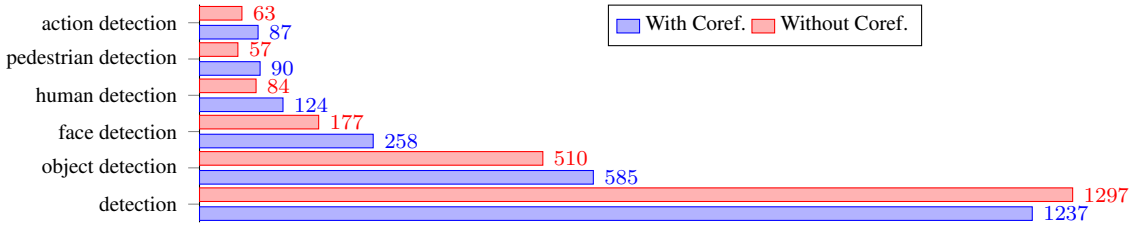


Figure 6.3: Frequency of detected entities with and without coreferece resolution: using coreference reduces the frequency of the generic phrase *detection* while significantly increasing the frequency of specific phrases. Linking entities through coreference helps disambiguate phrases when generating the knowledge graph.

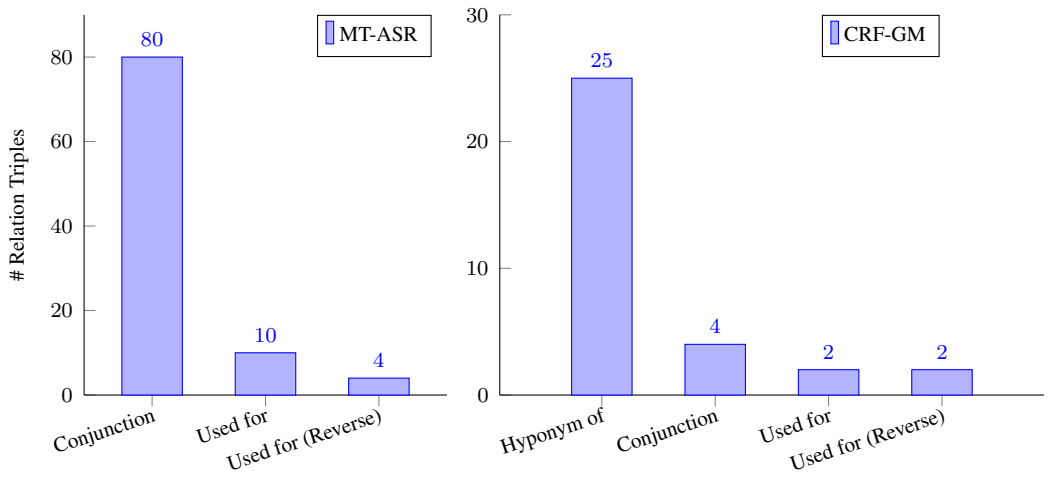


Figure 6.4: Frequency of relation types between pairs of entities: (*left*) automatic speech recognition (ASR) and machine translation (MT), (*right*) conditional random field (CRF) and graphical model (GM). We use the most frequent relation between pairs of entities in the knowledge graph.

string. Our qualitative analysis shows that there are fewer ambiguous phrases using coreference links (Figure 6.3). We calculate the frequency counts of all entities that appear in the whole corpus. We assign nodes in the knowledge graph by selecting the most frequent entities (with counts $> k$) in the corpus, and merge in any remaining entities for which a frequent entity is a substring. Even

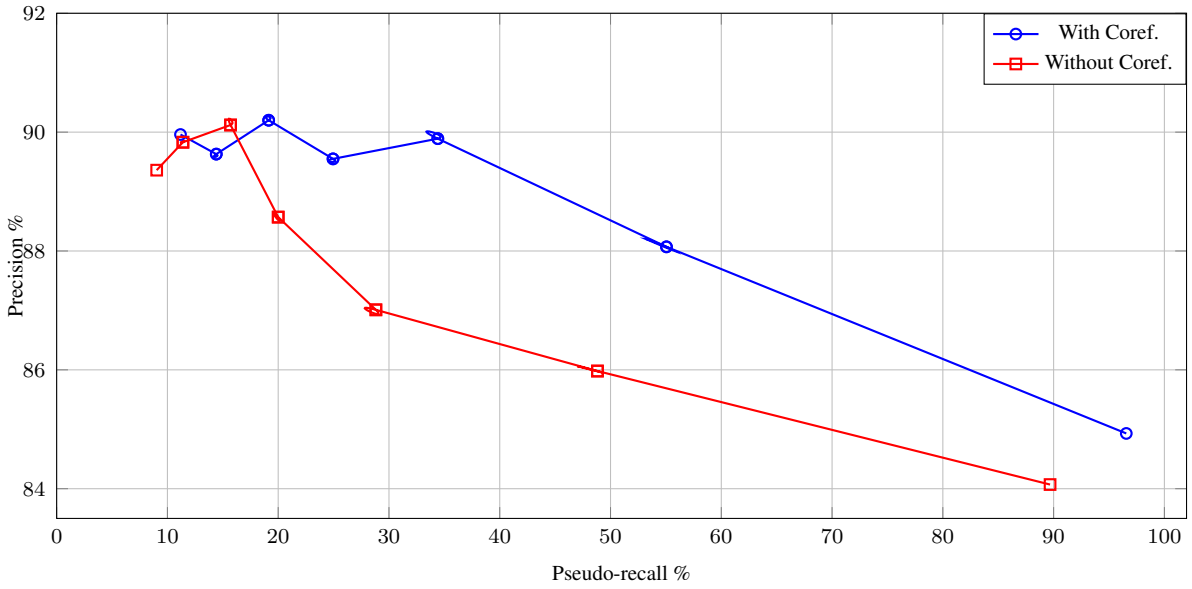


Figure 6.5: Precision/pseudo-recall curves for human evaluation by varying cut-off thresholds. The AUC is 0.751 with coreference, and 0.695 without.

though we do not disambiguate entity types when constructing the knowledge graph in Figure 6.2, the entity type can be obtained and used for other applications from our automatically extracted triples.

Assigning edges (relations) A pair of entities may appear in different contexts, resulting in different relation types between those entities (Figure 6.4). Even though it is obvious that the low frequent relations in Figure 6.4 are errors, it is possible that multiple relations hold between two entities. For every pair of entities in the graph, we calculate the frequency of different relation types across the whole corpus.We assign edges between entities by selecting the most frequent relation types to reduce noise.

6.1.1 Knowledge Graph Analysis

The knowledge graph for scientific community analysis is built using the Semantic Scholar Corpus (110k abstracts in total). We provide qualitative analysis and human evaluations on the constructed knowledge graph.

Knowledge Graph Evaluation and Analysis Figure 6.5 shows the human evaluation of the constructed knowledge graph, comparing the quality of automatically generated knowledge graphs with and without the coreference links. We randomly select 10 frequent scientific entities and extract all the relation triples that include one of the selected entities leading to 1.5k relation triples from both systems. We ask four domain experts to annotate each of these extracted relations to define ground truth labels. Each domain expert is assigned 2 or 3 entities and all of the corresponding relations. This is a harsher evaluation since entity types are not separated in graph. Figure 6.5 shows precision/recall curves for both systems. Since it is not feasible to compute the actual recall of the systems, we compute the pseudo-recall [118] based on the output of both systems. We observe that the knowledge graph curve with coreference linking is mostly above the curve without coreference linking. The precision of both systems is high (above 84% for both systems), but the system with coreference links has significantly higher pseudo recall.

Scientific trend analysis Figure 6.6 shows the historical trend analysis (from 1996 to 2016) of the most popular applications of the phrase *neural network*, selected according to the statistics of the extracted relation triples with the 'Used-for' relation type from speech, computer vision, and NLP conference papers. We observe that, before 2000, *neural network* has been applied to a greater percentage of speech applications compared to the NLP and computer vision papers. In NLP, neural networks first gain popularity in language modeling and then extend to other tasks such as POS Tagging and Machine Translation. In computer vision, the application of neural networks gains popularity in *object recognition* earlier (around 2010) than the other two more complex tasks of *object detection* and *image segmentation* (hardest and also the latest).

6.2 Summary

In this chapter, we create a new dataset SciERC for identifying entities, relations, and coreference clusters in scientific articles. Using our model SPANIE, we are able to automatically organize the extracted information from a large collection of scientific articles into a knowledge graph. Our analysis shows the importance of coreference links in making a dense, useful graph. Human evaluation shows that propagating coreference can significantly improve the quality of the automatic constructed knowledge graph. This work is done before DYGIE has been developed, therefore we expect to observe better performance if the knowledge graph is constructed through DYGIE .

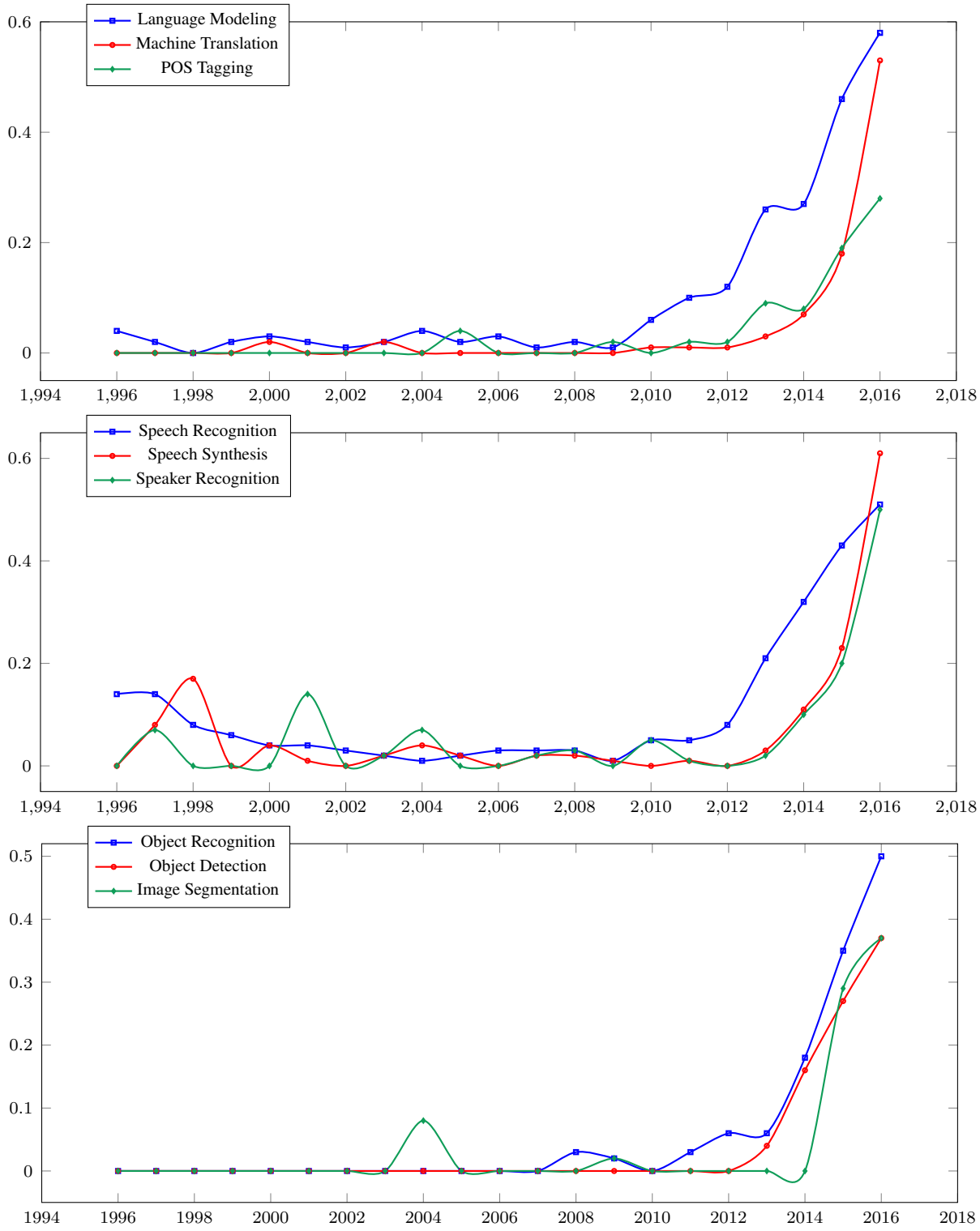


Figure 6.6: Historical trend for top applications of the keyphrase *neural network* in NLP, speech, and CV conference papers we collected. y-axis indicates the ratio of papers that use *neural network* in the task to the number of papers that is about the task.

Chapter 7

CONCLUSION

7.1 Summary

This thesis presented a series of techniques to improve the performance of low-resource IE tasks which involved approaches leveraging unannotated data as well as enhancing interactions across different IE tasks. In chapter 3, we developed an efficient way of improving the performance of supervised neural tagging systems through a new semi-supervised learning strategy for domain-matched leveraging unannotated data. We showed that through integrating a graph-based semi-supervised algorithm together with a confidence based self-training scheme, we are able to significantly boost the performance of the supervised neural model on a scientific keyphrase extraction task. In chapter 4, we introduced a span-based multi-task learning model SPANIE, for extracting entities, relations and coreference resolution. SPANIE leverages unannotated data through pre-trained contextualized embeddings. To assess performance gains for scientific information extraction, we create a dataset SciERC for scientific information extraction. Our experiments showed that the unified model is better at predicting span boundaries, and it outperforms previous state-of-the-art scientific IE systems on entity and relation extraction. In Chapter 5, we generalize and improve on SPANIE by leveraging rich contextual span representations and propagating information through coreference and relation links in SPANIE. DYGIE achieves state of the art in five different datasets ranging from News, Science to Biomedical and Wetlab Reports. In Chapter 6, we apply the span IE approach to construct a knowledge graph for scientific papers and demonstrate that the use of coreference leads to a high quality knowledge graph.

7.2 Impact

As scientific literature is increasingly available online via open access initiatives, there is growing interest in applying NLP tools to this genre, as for all online text. This thesis includes the earliest work on scientific information extraction and scientific knowledge graph construction. Research on IE for scientific literature has since gain a lot of attention. Many workshops in scientific literature have came out such as Scientific Literature Knowledge Bases (SLKB) Workshop and Workshop on Extracting Structured knowledge from Scientific Publications (ESSP). Our work has also impacted other studies more directly. The SciERC dataset provides a good testbed for evaluating IE systems on scientific domains and has been used in other studies [119]. In addition, the scientifc IE system we developed has been used for creating a large dataset for scientific abstract generation [120, 121], where DyGIE is used to construct a document level knowledge graph for each paper abstract. The abstract - knowledge graph pairs are then used for training an abstract generation system with the knowledge graph as input.

More generally, this thesis has had impact on IE in that, DyGIE is currently one of the best performing supervised IE models in terms of achieving high performance in different domains and tasks. DyGIE significantly outperforms the state-of-the-art on joint entity and relation detection tasks across four datasets: ACE 2004, ACE 2005, SciERC and the Wet Lab Protocol Corpus. Moreover, DyGIE excels at detecting entities with overlapping spans, achieving an improvement of up to 8 F1 points on three benchmarks annotated with overlapped spans: ACE 2004, ACE 2005 and GENIA.

7.3 Future directions

In this thesis, we have made the first step toward capturing long distance context by modeling the information flow between three IE tasks through a dynamically updated graph. There is great potential for modeling document-level semantic representations through building a internal dynamic span graph on a document, which may benefit a broader range of NLP tasks (beyond IE) such as Question Answering (QA) or generation. In addition, the dynamic span graph idea can also be

used for building the connection between raw text and an existing knowledge base. We consider our dynamic span graphs has the potential to be extented in the following ways:

Document-level dynamic span graphs The research in this thesis represents only abstracts, but the dynamic span graphs can be extended to whole documents. The document level dynamic span graphs have the potential to be extended to any NLP tasks that requires grounding and reasoning. For example, in the Squad QA [76] task, the model needs to reason over a given document and predict an answer, where attention-based approaches have been proved to be effective in recent literature [122]. However, since the attention model is learned in a completely unconstrained fashion where all tokens within the context window are connected to each other, the reasoning space over a long context can be hard to track [123]. Dynamic span graphs have the potential to reduce the problem by constructing a limited space in a graph consisting of high-confidence spans. The edge scores that connect the graph can be obtained from supervised tasks such as coreference, or from unconstrained self-attention weights. It is easier to reason on the constrained space over the constructed graph. For example, in order to answer the question "*What can be used to treat tumors with L858E mutation in EGFR gene?*" given the text "*The deletion mutation on exon-19 of EGFR gene was present in 16 patients, while the L858E point mutation on exon-21 was noted in 10. All patients were treated with gefitinib...*", we need to figure out multiple relations between different genes and proteins. We can construct a dynamic graph that directly connects *EGFR gene*, *L858E point mutation*, *gefitinib*, *exon-19* and *exon-21* with different edge scores learned from dynamic graph parameters, which makes it easier for reasoning over the space.

Connecting with existing knowledge bases There have been many large scale, high quality knowledge bases built in recent years, including Freebase [124] and YAGO2 [125] that have greatly contribute to the performance of NLP tasks such as QA and search. For many NLP tasks, knowledge bases provide an more easily accessible resource of external knowledge through structured data. The connection between raw text and knowledge bases can be built through the entity linking task, which determines the identity of entities mentioned in text where the entities are linked to the

corresponding entry in the knowledge base.

The dynamic span graph approach can be extended to the entity linking problem. Take the same example of *EGFR gene* and *L858E point mutation*, a mapping between the span candidates extracted from the raw text can be linked to an external knowledge base through an additional entity linking task. Specifically, a beam can be applied to keep the most confident entities that are likely to be referring to an entry in the knowledge base. Then an end-to-end scoring function can be used to predict the most likely span-entry pairs. Once this connection is built, information from knowledge base can be used to help downstream tasks on raw text such as QA or IE.

The dynamic graph constructed from the raw text can also be used for knowledge graph completion tasks, where we can predict a missing entity or a missing relation edge by leveraging the confidence from both raw text and from the knowledge graph structure.

Temporal dynamic span graphs In this thesis, the dynamics of the graph is associated with iterative refinement over extended contexts. However, it is also possible for these dynamics to track changes over time associated with news updates, an evolving discussion or a long dialogue that changes topics from time to time. In particular, the dynamic span graph can be used to track the state change over time, with parameters associated with temporal variants that can be learned from the data.

Scientific recommendation system In this thesis we have explored knowledge graph construction and trend analysis for scientific papers. Future work involves scientific recommendation for related works that can further benefit academic search for researchers. For example, a recommendation system can be designed to provide methods such as CRF, LSTM etc, that have been used for the NER task in the literature. The problem can be cast as knowledge base completion where the recommendation process can be formulated as a ranking problem. The relation scores between the queried scientific term and all other scientific terms in the knowledge graph can be calculated and ranked in descending order, where the recommendation system provides the user with the top k term candidates that have the highest *Used-for* relation score. The system can further predict

facts that have not been observed in the existing collection of papers through link prediction as in [126].

BIBLIOGRAPHY

[1] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 1105–1116, 2016.

[2] Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. Improved relation classification by deep recurrent neural networks with data augmentation. In *Proc. Int. Conf. Computational Linguistics (COLING)*, pages 1461–1470, 2016.

[3] Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. Cross-sentence n-ary relation extraction with graph lstms. *Trans. Assoc. for Computational Linguistics (TACL)*, 5:101–115, 2017.

[4] Chris Quirk and Hoifung Poon. Distant supervision for relation extraction beyond the sentence boundary. In *Proc. European Chapter Assoc. for Computational Linguistics (EACL)*, pages 1171–1182, 2017.

[5] Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. The uwnlp system at semeval-2018 task 7: Neural relation extraction model with selectively incorporated concept embeddings. In *Proc. Int. Workshop on Semantic Evaluation (SemEval)*, pages 788–792, 2018.

[6] Heike Adel and Hinrich Schütze. Global normalization of convolutional neural networks for joint entity and relation classification. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1723–1729, 2017.

[7] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009.

[8] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.

[9] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Machine Learning Research*, 12(Aug):2493–2537, 2011.

[10] Sujan Kumar Saha, Sudeshna Sarkar, and Pabitra Mitra. Gazetteer preparation for named entity recognition in indian languages. In *IJCNLp*, pages 9–16, 2008.

[11] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications*, pages 104–107. Association for Computational Linguistics, 2004.

[12] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, volume 1, pages 402–412, 2014.

[13] Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 536–540, 2015.

[14] Meishan Zhang, Yue Zhang, and Guohong Fu. End-to-end neural relation extraction with global optimization. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1730–1740, 2017.

[15] Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. In *Proc. Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 626–634, 2015.

[16] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1785–1794, 2015.

[17] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 1201–1211. Association for Computational Linguistics, 2012.

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

[19] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, pages 3079–3087, 2015.

[20] Yi Luan, Daisuke Saito, Yosuke Kashiwagi, Nobuaki Minematsu, and Keikichi Hirose. Semi-supervised noise dictionary adaptation for exemplar-based noise robust speech recognition. In *Proc. Int. Conf. Acoustic, Speech, and Signal Process. (ICASSP)*, pages 1745–1748. IEEE, 2014.

[21] Bret Harsham, Shinji Watanabe, Alan Esenther, John Hershey, Jonathan Le Roux, Yi Luan, Daniel Nikovski, and Vamsi Potluru. Driver prediction to improve interaction with in-vehicle hmi. In *Proc. Workshop on Digital Signal Processing for In-Vehicle Systems*, 2015.

[22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *arXiv preprint arXiv:1301.3781*, 2013.

[23] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, volume 14, pages 1532–1543, 2014.

[24] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 302–308, 2014.

[25] Yi Luan, Yangfeng Ji, and Mari Ostendorf. LSTM based conversation models. In *arXiv preprint arXiv:1603.09457*, 2016.

[26] Yi Luan, Shinji Watanabe, and Bret Harsham. Efficient learning for spoken language understanding tasks with word embedding based pre-training. In *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, pages 1398–1402. Citeseer, 2015.

[27] Yi Luan, Yangfeng Ji, Hannaneh Hajishirzi, and Boyang Li. Multiplicative representations for unsupervised semantic role induction. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, page 118, 2016.

[28] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018.

[29] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

[30] Amarnag Subramanya and Jeff Bilmes. Semi-supervised learning with measure propagation. *J. Machine Learning Research*, 12(Nov):3311–3370, 2011.

[31] Yuzong Liu and Katrin Kirchhoff. Graph-based semi-supervised learning for phone and segment classification. In *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2013.

[32] Yuzong Liu and Katrin Kirchhoff. Acoustic modeling with neural graph embeddings. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.

[33] Yuzong Liu and Katrin Kirchhoff. Graph-based semisupervised learning for acoustic modeling in automatic speech recognition. *IEEE/ACM Trans. Audio, Speech, and Language Process.*, 24(11):1946–1956, 2016.

[34] Yuzong Liu and Katrin Kirchhoff. Novel front-end features based on neural graph embeddings for DNN-HMM and LSTM-CTC acoustic modeling. In *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*. ISCA, 2016.

[35] Young-Bum Kim, Minwoo Jeong, Karl Stratos, and Ruhi Sarikaya. Weakly supervised slot tagging with partially labeled sequences from web search click logs. In *Proc. Conf. Human Language Technology and North American Assoc. for Computational Linguistics (HLT-NAACL)*, pages 84–92, 2015.

[36] Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. Scientific information extraction with semi-supervised neural tagging. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2017.

[37] Sameer Singh, Sebastian Riedel, Brian Martin, Jiaping Zheng, and Andrew McCallum. Joint inference of entities, relations, and coreference. In *Proc. of the 2013 workshop on Automated knowledge base construction*, pages 1–6. ACM, 2013.

[38] Bishan Yang and Tom M Mitchell. 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*, pages 289–299, 2016.

[39] Nanyun Peng and Mark Dredze. Named entity recognition for chinese social media with jointly trained embeddings. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 548–554, 2015.

[40] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2018.

[41] Yi Luan, Chris Brockett, Bill Dolan, Jianfeng Gao, and Michel Galley. Multi-task learning for speaker-role adaptation in neural conversation models. In *Proc. IJCNLP*, 2017.

[42] Kenton Lee, Luheng He, Mike Lewis, and Luke S. Zettlemoyer. End-to-end neural coreference resolution. In *EMNLP*, 2017.

[43] Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. Semeval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proc. Int. Workshop on Semantic Evaluation (SemEval)*, 2017.

[44] Sonal Gupta and Christopher D Manning. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proc. IJCNLP*, pages 1–9, 2011.

[45] Chen-Tse Tsai, Gourab Kundu, and Dan Roth. Concept-based analysis of scientific literature. In *Proc. ACM Int. Conference on Information & Knowledge Management*, pages 1733–1738. ACM, 2013.

[46] Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. Semeval-2018 Task 7: Semantic relation extraction and classification in scientific papers. In *Proc. Int. Workshop on Semantic Evaluation (SemEval)*, 2018.

[47] Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. Jointly predicting predicates and arguments in neural semantic role labeling. In *ACL*, 2018.

[48] Arzoo Katiyar and Claire Cardie. Nested named entity recognition revisited. In *Proc. Conf. North American Assoc. for Computational Linguistics (NAACL)*, 2018.

[49] Bailin Wang and Wei Lu. Neural segmental hypergraphs for overlapping mention recognition. In *EMNLP*, 2018.

[50] John Lafferty, Andrew McCallum, Fernando Pereira, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. Int. Conf. Machine Learning (ICML)*, volume 1, pages 282–289, 2001.

[51] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. In *arXiv preprint arXiv:1508.01991*, 2015.

[52] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns. In *Trans. Assoc. for Computational Linguistics (TACL)*, 2016.

[53] Miguel Ballesteros, Chris Dyer, and Noah A Smith. Improved transition-based parsing by modeling characters instead of words with LSTMs. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2015.

[54] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, 2016.

[55] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proc. Conf. North American Assoc. for Computational Linguistics (NAACL)*, 2016.

[56] Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. Bidirectional long short-term memory networks for relation classification. In *PACLIC*, 2015.

[57] Rui Cai, Xiaodong Zhang, and Houfeng Wang. Bidirectional recurrent convolutional neural network for relation classification. 2016.

[58] Sam Wiseman, Alexander M Rush, and Stuart M Shieber. Learning global features for coreference resolution. *arXiv preprint arXiv:1604.03035*, 2016.

[59] Kevin Clark and Christopher D. Manning. Improving coreference resolution by learning entity-level distributed representations. *CoRR*, abs/1606.01323, 2016.

[60] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. Int. Conf. Machine Learning (ICML)*, pages 160–167, 2008.

[61] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Machine learning*, 39(2):103–134, 2000.

[62] Ellen Riloff, Janyce Wiebe, and Theresa Wilson. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 25–32. Association for Computational Linguistics, 2003.

[63] Shumeet Baluja. Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. In *NIPS*, pages 854–860, 1998.

[64] Yuzong Liu and Katrin Kirchhoff. Graph-based semi-supervised acoustic modeling in DNN-based speech recognition. In *Proc. IEEE Workshop on Spoken Language Technology (SLT)*, 2014.

[65] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proc. Int. Conf. Machine Learning (ICML)*, volume 3, pages 912–919, 2003.

[66] Adrian Corduneanu and Tommi Jaakkola. On information regularization. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 151–158. Morgan Kaufmann Publishers Inc., 2002.

[67] Amarnag Subramanya and Jeff Bilmes. Soft-supervised learning for text classification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1090–1099. Association for Computational Linguistics, 2008.

[68] Kohei Ozaki, Masashi Shimbo, Mamoru Komachi, and Yuji Matsumoto. Using the mutual k-nearest neighbor graphs for semi-supervised classification of natural language data. In *Proceedings of the fifteenth conference on computational natural language learning*, pages 154–162. Association for Computational Linguistics, 2011.

[69] Andrei Alexandrescu and Katrin Kirchhoff. Graph-based learning for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 119–127. Association for Computational Linguistics, 2009.

[70] Amarnag Subramanya, Slav Petrov, and Fernando Pereira. Efficient graph-based semi-supervised learning of structured tagging models. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 167–176, 2010.

[71] Sherzod Hakimov, Salih Atilay Oto, and Erdogan Dogdu. Named entity recognition and disambiguation using linked data and graph-based centrality scoring. In *Proceedings of the 4th international workshop on semantic web information management*, page 4. ACM, 2012.

[72] Mohammad Aliannejadi, Masoud Kiaeeha, Shahram Khadivi, and Saeed Shiry Ghidary. Graph-based semi-supervised conditional random fields for spoken language understanding using unaligned data. *arXiv preprint arXiv:1701.08533*, 2017.

[73] Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, volume 1, pages 1756–1765, 2017.

[74] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005.

[75] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. 2003.

[76] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. 2016.

[77] Awais Athar and Simone Teufel. Detection of implicit citations for sentiment detection. In *Proc. ACL Workshop on Detecting Structure in Scholarly Discourse*, pages 18–26, 2012.

[78] Awais Athar and Simone Teufel. Context-enhanced citation sentiment detection. In *Proc. Conf. North American Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 597–601, 2012.

[79] Huy Hoang Nhat Do, Muthu Kumar Chandrasekaran, Philip S Cho, and Min Yen Kan. Extracting and matching authors and affiliations in scholarly documents. In *Proc. ACM/IEEE-CS Joint Conference on Digital libraries*, pages 219–228, 2013.

[80] Kokil Jaidka, Muthu Kumar Chandrasekaran, Beatriz Fisas Elizalde, Rahul Jha, Christopher Jones, Min-Yen Kan, Ankur Khanna, Diego Molla-Aliod, Dragomir R Radev, Francesco Ronzano, et al. The computational linguistics summarization pilot task. In *Proc. Text Analysis Conference*, 2014.

[81] Miray Kas. Structures and statistics of citation networks. Technical report, DTIC Document, 2011.

[82] Yanchuan Sim, Noah A Smith, and David A Smith. Discovering factions in the computational linguistics community. In *Proc. ACL Special Workshop on Rediscovering 50 Years of Discoveries*, pages 22–32, 2012.

[83] Adam Vogel and Dan Jurafsky. He said, she said: Gender in the ACL anthology. In *Proc. ACL Special Workshop on Rediscovering 50 Years of Discoveries*, pages 33–41, 2012.

[84] Ashton Anderson, Dan McFarland, and Dan Jurafsky. Towards a computational history of the ACL: 1980-2008. In *Proc. ACL Special Workshop on Rediscovering 50 Years of Discoveries*, pages 13–21, 2012.

[85] Amjad Abu-Jbara and Dragomir Radev. Coherent citation-based summarization of scientific papers. In *Proc. Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 500–509, 2011.

[86] Simone Teufel and Marc Moens. Summarizing scientific articles: experiments with relevance and rhetorical status. *Computational linguistics*, 28(4):409–445, 2002.

[87] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Proc. Joint SIGDAT conference on empirical methods in natural language processing and very large corpora*, pages 100–110, 1999.

[88] Matthew Gerber and Joyce Y Chai. Beyond nombank: A study of implicit arguments for nominal predicates. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1583–1592. Association for Computational Linguistics, 2010.

[89] Katsumasa Yoshikawa, Sebastian Riedel, Tsutomu Hirao, Masayuki Asahara, and Yuji Matsumoto. Coreference based event-argument relation extraction on biomedical text. *Journal of Biomedical Semantics*, 2(5):S6, 2011.

[90] Anita de Waard and Henk Pander Maat. Verb form indicates discourse segment type in biological research papers: Experimental evidence. *Journal of English for academic purposes*, 11(4):357–366, 2012.

[91] Waleed Ammar, Matthew Peters, Chandra Bhagavatula, and Russell Power. The ai2 system at semeval-2017 task 10 (scienceie): semi-supervised end-to-end entity and relation extraction. In *Proc. Int. Workshop on Semantic Evaluation (SemEval)*, pages 592–596, 2017.

[92] Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, et al. Construction of the literature graph in semantic scholar. In *Proc. Conf. North American Assoc. for Computational Linguistics: Human Language Technologies (NAACL-HLT), (Industry Papers)*, pages 84–91, 2018.

[93] Isabelle Augenstein and Anders Søgaard. Multi-task learning of keyphrase boundary classification. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 341–346, 2017.

[94] Yi Luan. Information extraction from scientific literature for method recommendation. *arXiv preprint arXiv:1901.00401*, 2018.

[95] Mohammad Aliannejadi, Masoud Kiaeeha, Shahram Khadivi, and Saeed Shiry Ghidary. Graph-based semi-supervised conditional random fields for spoken language understanding using unaligned data. In *Proc. Australasian Language Technology Association Workshop*, page 98, 2014.

[96] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford CoreNLP natural language processing toolkit. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 55–60, 2014.

[97] Behrang QasemiZadeh and Anne-Kathrin Schumann. The ACL RD-TEC 2.0: A language resource for evaluating term extraction and entity recognition methods. In *LREC*, 2016.

[98] Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. Brat: a web-based tool for nlp-assisted text annotation. In *Proc. European Chapter Assoc. for Computational Linguistics (EACL)*, pages 102–107, 2012.

[99] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Proc. Annu. Conf. Neural Inform. Process. Syst. (NIPS)*, 2016.

[100] Sam Wiseman, Alexander M. Rush, and Stuart M. Shieber. Learning global features for coreference resolution. In *HLT-NAACL*, 2016.

[101] Arzoo Katiyar and Claire Cardie. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, volume 1, pages 917–928, 2017.

[102] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, volume 1, pages 1227–1236, 2017.

[103] Kenton Lee, Luheng He, and Luke Zettlemoyer. Higher-order coreference resolution with coarse-to-fine inference. In *NAACL*, 2018.

[104] Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. Improving sentence compression by learning to predict gaze. In *HLT-NAACL*, 2016.

[105] Marek Rei. Semi-supervised multitask learning for sequence labeling. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, 2017.

[106] Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. *CoRR*, abs/1706.09528, 2017.

[107] Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. A general framework for information extraction using dynamic span graphs. In *Proc. Conf. North American Assoc. for Computational Linguistics (NAACL)*, 2019.

[108] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL-Shared Task*, pages 1–40. Association for Computational Linguistics, 2012.

[109] Chaitanya Kulkarni, Wei Xu, Alan Ritter, and Raghu Machiraju. An annotated corpus for machine reading of instructions in wet lab protocols. In *NAACL-HLT*, 2018.

[110] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. Adversarial training for multi-context joint entity and relation extraction. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2830–2836, 2018.

[111] Victor Sanh, Thomas Wolf, and Sebastian Ruder. A hierarchical multi-task approach for learning embeddings from semantic tasks. *AAAI*, 2019.

[112] Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. Genia corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19 Suppl 1:i180–2, 2003.

[113] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.

[114] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, 2013.

[115] Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. N-ary relation extraction using graph-state lstm. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2226–2235, 2018.

[116] Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2018.

[117] Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. A walk-based model on entity graphs for relation extraction. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, volume 2, pages 81–88, 2018.

[118] Congle Zhang, Stephen Soderland, and Daniel S. Weld. Exploiting parallel news streams for unsupervised event extraction. *TACL*, 3:117–129, 2015.

[119] Iz Beltagy, Arman Cohan, and Kyle Lo. Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

[120] Rik Koncel-Kedziorski, Dhanush Bekal, Yi Luan, Mirella Lapata, and Hannaneh Hajishirzi. Text generation from knowledge graphs with graph transformers. In *Proc. Conf. North American Assoc. for Computational Linguistics (NAACL)*, 2019.

[121] Qingyun Wang, Lifu Huang, Zhiying Jiang, Kevin Knight, Heng Ji, Mohit Bansal, and Yi Luan. Paperrobot: Incremental draft generation of scientific ideas. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, 2019.

[122] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2016.

[123] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 2369–2380, 2018.

[124] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.

[125] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[126] Kristina Toutanova, Xi Victoria Lin, Wen-tau Yih, Hoifung Poon, and Chris Quirk. Compositional learning of embeddings for relation paths in knowledge bases and text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1434–1444, 2016.

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

[128] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[129] Yee Seng Chan and Dan Roth. Exploiting syntactico-semantic structures for relation extraction. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, 2011.

[130] Greg Durrett and Dan Klein. A joint model for entity analysis: Coreference, typing, and linking. *Trans. Assoc. for Computational Linguistics (TACL)*, 2:477–490, 2014.

[131] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, 2014.

[132] Aron Culotta, Andrew McCallum, and Jonathan Betz. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Proc. Conf. North American Assoc. for Computational Linguistics (NAACL)*, pages 296–303. Association for Computational Linguistics, 2006.

[133] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[134] Scott Miller, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. A novel use of statistical parsing to extract information from text. In *Proceedings of the 1st North American chapter of the Association for Computational Linguistics conference*, pages 226–233. Association for Computational Linguistics, 2000.

[135] Jie Wang, Jiayu Zhou, Peter Wonka, and Jieping Ye. Advances in neural information processing systems. In *Neural information processing systems foundation*, 2013.

[136] Vincent Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1396–1411. Association for Computational Linguistics, 2010.

[137] Shubin Zhao and Ralph Grishman. Extracting relations with integrated information using kernel methods. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 419–426. Association for Computational Linguistics, 2005.

[138] Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 492–501. Association for Computational Linguistics, 2010.

[139] Hannaneh Hajishirzi, Leila Zilles, Daniel S Weld, and Luke Zettlemoyer. Joint coreference resolution and named-entity linking with multi-pass sieves. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 289–299, 2013.

[140] Jie Zhou and Wei Xu. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 1127–1137, 2015.

[141] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In *Proc. Conf. Computational Natural Language Learning (CoNLL)*, pages 147–155, 2009.

[142] Kata Gabor, Haifa Zargayouna, Davide Buscaldi, Isabelle Tellier, and Thierry Charnois. Semantic annotation of the ACL anthology corpus for the automatic analysis of scientific literature. In *Proc. Language Resources and Evaluation Conference (LREC)*, 2016.

[143] Kata Gábor, Haïfa Zargayouna, Isabelle Tellier, Davide Buscaldi, and Thierry Charnois. Unsupervised relation extraction in specialized corpora using sequence mining. In *International Symposium on Intelligent Data Analysis*, pages 237–248. Springer, 2016.

[144] Kokil Jaidka, Muthu Kumar Chandrasekaran, Sajal Rustagi, and Min-Yen Kan. Overview of the cl-scisumm 2016 shared task. In *Proc. Joint Workshop on Bibliometric-enhanced Information Retrieval and NLP for Digital Libraries (BIRNDL 2016)*, 2016.

[145] Yi Luan, Masayuki Suzuki, Yutaka Yamauchi, Nobuaki Minematsu, Shuhei Kato, and Keikichi Hirose. Performance improvement of automatic pronunciation assessment in a noisy classroom. In *Proc. IEEE Workshop on Spoken Language Technology (SLT)*, pages 428–431. IEEE, 2012.

[146] Yanjun Qi, Pavel Kuksa, Ronan Collobert, Kunihiko Sadamasa, Koray Kavukcuoglu, and Jason Weston. Semi-supervised sequence labeling with self-learned features. In *Proc. IEEE Int. Conference on Data Mining*, pages 428–437, 2009.

[147] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *Proc. IJCAI*, volume 7, pages 2670–2676, 2007.

[148] Anne-Kathrin Schumann and Behrang QasemiZadeh. The acl rd-tec annotation guideline.

[149] Behrang QasemiZadeh and Anne-Kathrin Schumann. The acl rd-tec 2.0: A language resource for evaluating term extraction and entity recognition methods. In *Proc. Language Resources and Evaluation Conference (LREC)*, 2012.

[150] Steven Bird, Robert Dale, Bonnie J Dorr, Bryan R Gibson, Mark Thomas Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir R Radev, Yee Fan Tan, et al. The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proc. Language Resources and Evaluation Conference (LREC)*, 2008.

[151] Alan Ritter, Colin Cherry, and William B Dolan. Data-driven response generation in social media. In *Proc. Conf. Empirical Methods Natural Language Process. (EMNLP)*, pages 583–593, 2011.

[152] Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*, 2015.

[153] Thorsten Joachims et al. Transductive learning via spectral graph partitioning. In *Proc. Int. Conf. Machine Learning (ICML)*, volume 3, pages 290–297, 2003.

[154] Oriol Vinyals and Quoc Le. A neural conversational model. *arXiv preprint arXiv:1506.05869*, 2015.

[155] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015.

[156] Hongyuan Mei, Mohit Bansal, and Matthew R Walter. Coherent dialogue with attention-based language models. *arXiv preprint arXiv:1611.06997*, 2016.

[157] Yi Luan, Richard Wright, Mari Ostendorf, and Gina-Anne Levow. Relating automatic vowel space estimates to talker intelligibility. In *Proc. Conf. Int. Speech Communication Assoc. (INTERSPEECH)*, 2014.

[158] Gina-Anne Levow, Valerie Freeman, Alena Hrynkevich, Mari Ostendorf, Richard Wright, Julian Chan, Yi Luan, and Trang Tran. Recognition of stance strength and polarity in spontaneous speech. In *Proc. IEEE Workshop on Spoken Language Technology (SLT)*, pages 236–241, 2014.

[159] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*, 2016.

[160] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

[161] Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. A neural network approach to context-sensitive generation of conversational responses. *arXiv preprint arXiv:1506.06714*, 2015.

[162] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 311–318, 2002.

[163] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015.

[164] Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proc. Annu. Meeting Assoc. for Computational Linguistics (ACL)*, pages 160–167, 2003.

[165] Kenton Lee, Mike Lewis, and Luke Zettlemoyer. Global neural ccg parsing with optimality guarantees. In *EMNLP*, 2016.

[166] Phong Le and Ivan Titov. Improving entity linking by modeling latent relations between mentions. 2018.

[167] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[168] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: A Simple and General Method for Semi-supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010.

[169] Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *ICLR*, 2015.

[170] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint*, 2014.

[171] Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass. Highway long short-term memory rnns for distant speech recognition. In *ICASSP*, 2016.

[172] Guillaume Lample, Miguel Ballesteros, Sandeep K Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *HLT-NAACL*, 2016.

[173] Émile Enguehard, Yoav Goldberg, and Tal Linzen. Exploring the syntactic abilities of rnns with multi-task learning. In *CoNLL*, 2017.

[174] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*, 2008.

[175] Hao Peng, Sam Thomson, and Noah A. Smith. Deep multitask learning for semantic dependency parsing. In *ACL*, 2017.

[176] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. Conll-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *EMNLP-CoNLL Shared Task*, 2012.

[177] Wei Lu and Dan Roth. Joint mention extraction and classification with mention hypergraphs. In *EMNLP*, 2015.

[178] Jenny Rose Finkel and Christopher D. Manning. Nested named entity recognition. In *EMNLP*, 2009.

Appendix A

ANNOTATION GUIDELINE FOR SCIERC DATASET

A.1 Annotation Guideline

A.1.1 Entity Category

- **Task**: Applications, problems to solve, systems to construct.

  E.g. information extraction, machine reading system, image segmentation, etc.

- **Method**: Methods , models, systems to use, or tools, components of a system, frameworks.

  E.g. language model, CORENLP, POS parser, kernel method, etc.

- **Evaluation Metric**: Metrics, measures, or entities that can express quality of a system/method.

  E.g. F1, BLEU, Precision, Recall, ROC curve, mean reciprocal rank, mean-squared error, robustness, time complexity, etc.

- **Material**: Data, datasets, resources, Corpus, Knowledge base.

  E.g. image data, speech data, stereo images, bilingual dictionary, paraphrased questions, CoNLL, Panntreebank, WordNet, Wikipedia, etc.

- **Other Scientific Terms**: Phrases that are a scientific terms but do not fall into any of the above classes

  E.g. physical or geometric constraints, qualitative prior knowledge, discourse structure, syntactic rule, discourse structure, tree, node, tree kernel, features, noise, criteria

- **Generic**: General terms or pronouns that may refer to a entity but are not themselves informative, often used as connection words.

  E.g model, approach, prior knowledge, them, it...

A.1.2 Relation Category

Relation link can not go beyond sentence boundary. We define 4 asymmetric relation types (*Used-for*, *Feature-of*, *Hyponym-of*, *Part-of*), together with 2 symmetric relation types (*Compare*, *Conjunction*). B always points to A for asymmetric relations

- **Used-for**: B is used for A, B models A, A is trained on B, B exploits A, A is based on B. E.g.

  The TISPER system has been designed to enable many text applications.

  Our method models user proficiency.

  Our algorithms exploits local soothness.

- **Feature-of**: B belongs to A, B is a feature of A, B is under A domain. E.g.

  prior knowledge of the model

  genre-specific regularities of discourse structure

  English text in science domain

- **Hyponym-of**: B is a hyponym of A, B is a type of A. E.g.

  TUIT is a software library

  NLP applications such as machine translation and language generation

- **Part-of**: B is a part of A... E.g.

The system includes two models: speech recognition and natural language understanding

We incorporate NLU module to the system.

- **Compare**: Symmetric relation (use blue to denote entity). Opposite of conjunction, compare two models/methods, or listing two opposing entities. E.g.

  Unlike the quantitative prior, the qualitative prior is often ignored...

  We compare our system with previous sequential tagging systems...

- **Conjunction**: Symmetric relation (use blue to denote entity). Function as similar role or use/incorporate with. E.g.

  obtained from human expert or knowledge base

  NLP applications such as machine translation and language generation

A.1.3 Coreference

Two Entities that points to the same concept.

- **Anaphora and Cataphora**:

  We introduce a machine reading system... The system...

  The prior knowledge include...Such knowledge can be applied to...

- **Coreferring noun phrase**:

  We develop a part-of-speech tagging system...The POS tagger...

A.1.4 Notes

1. Entity boundary annotation follows the ACL RD-TEC Annotation Guideline [97], with the extention that spans can be embedded in longer spans, only if the shorter span is involved in a relation.

2. Do not include determinators (such as the, a), or adjective pronouns (such as this,its, these, such) to the span. If generic phrases are not involved in a relation, do not tag them.

3. Do not tag relation if one entity is:

   - Variable bound:

     We introduce a neural based approach.. *Its* benefit is...

   - The word *which*:

     We introduce a neural based approach, *which* is a...

4. Do not tag coreference if the entity is

   - Generically-used Other-ScientificTerm:

     ...advantage gained from *local smoothness* which... We present algorithms exploiting *local smoothness* in more aggressive ways...

   - Same scientific term but refer to different examples:

     We use a *data structure*, we also use another *data structure*...

5. Do not label negative relations:

   X is not used in Y or X is hard to be applied in Y

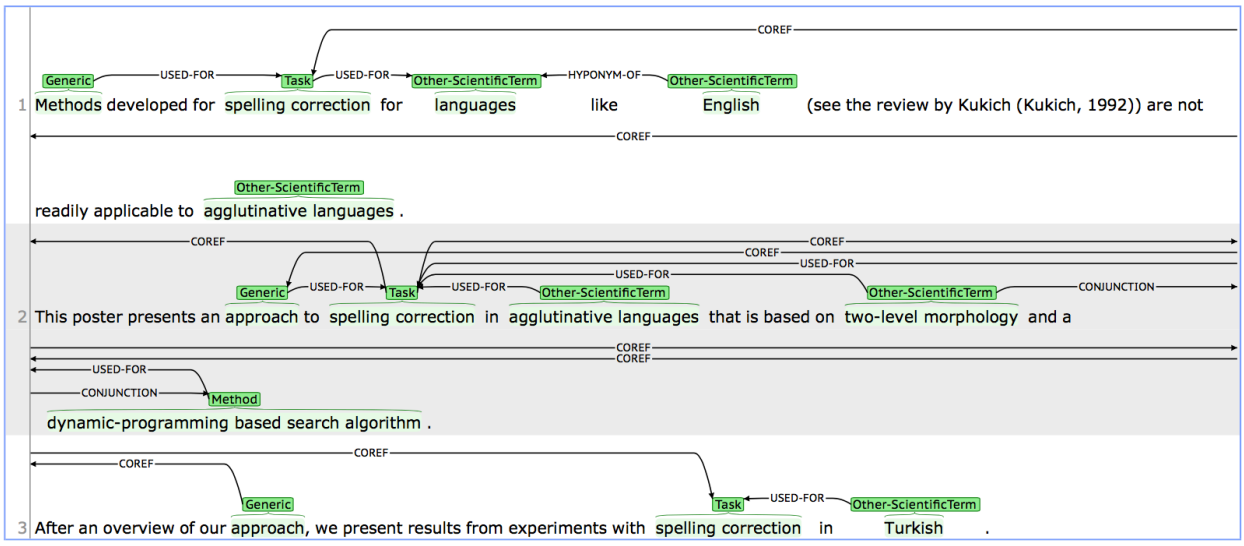


Figure A.1: Annotation example 1 from ACL

A.2 Annotation Examples

Here we take a screen shot of the BRAT interface for an ACL paper in Figure A.1. We also attach the original figure of Figure 3 in Figure B.1. More examples can be found in the project website[1].

[1] http://ssli.ee.washington.edu/tial/projects/sciIE/

Appendix B

SCIENTIFIC KNOWLEDGE GRAPH EXAMPLES

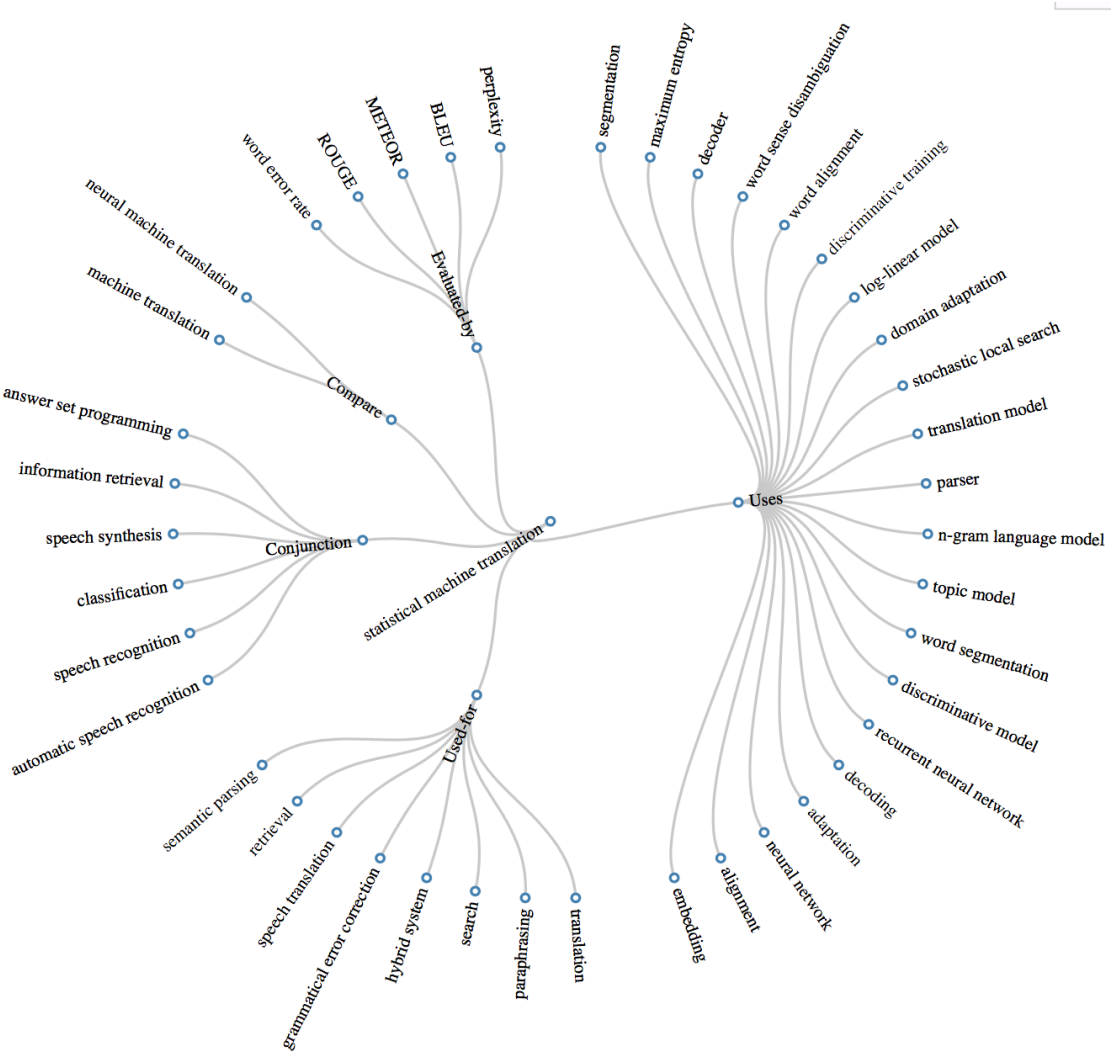


Figure B.1: An example of our automatically generated knowledge graph centered on *statistical machine translation*. This is the original figure of Figure 6.2.