Relation Extraction: from Ontological Smoothing to Temporal Correspondence

Congle Zhang

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

Doctor of Philosophy

University of Washington

2015

Reading Committee:

Daniel S. Weld, Chair

Luke Zettlemoyer

Stephen Soderlansd

Program Authorized to Offer Degree:
UW Computer Science & Engineering

University of Washington

Abstract

Relation Extraction: from Ontological Smoothing to Temporal Correspondence

Congle Zhang

Chair of the Supervisory Committee:
Professor Daniel S. Weld
Computer Science & Engineering

Relation extraction, the task of extracting facts from natural language text and creating machine readable knowledge, is a great dream of artificial intelligence. Today, most approaches to relation extraction are based on machine learning and thus starved by scarce training data. Distant supervision, which automatically creates training data, only works with relations that already populate a knowledge base. In particular, most dynamic, time dependent event relations are ephemeral and are rarely stored in a pre-existing knowledge base. This drawback seriously limits the usability of distant supervision.

To address the challenges of relation extraction, we present four novel techniques VELVET, NEWSSPIKE-PARA, NEWSSPIKE-RE, NEWSSPIKE-SCALE. They are based on two key ideas. The first is ontological smoothing, that allows us to map the target relations to database views over a background knowledge base, and thus allow distant supervision to work on the user specified relations. The second is temporal correspondence, that allows us to exploit parallel news streams to generate accurate training sentences for large sets of event relations.

In this dissertation, we formalize the characteristics necessary for ontological smoothing and temporal correspondence. We develop the algorithms that automatically learn scalable relation extractors. The results of our experiments show that the learned extractors predict high quality extractions for both static and event relations.

TABLE OF CONTENTS

Page

List of Figures iii

List of Tables vi

Chapter 1: Introduction 1
1.1 Relation Extraction and its Challenges 4
1.2 Distant Supervision and Ontological Smoothing 11
1.3 Parallel News Streams and Temporal Correspondence 16
1.4 Contributions 23

Chapter 2: VELVET: Ontological Smoothing for Relation Extraction 26
2.1 Introduction 26
2.2 Constructing Ontological Mappings 29
2.3 Relation Extraction 35
2.4 Empirical Evaluation 36
2.5 Conclusion 42

Chapter 3: NewsSpike-Para: Harvesting Parallel News Streams to Generate Paraphrases of Event Relations 45
3.1 Introduction 45
3.2 System Overview 47
3.3 Temporal Correspondence Heuristics 49
3.4 Exploiting the Temporal Heuristics 51
3.5 Empirical Evaluation 57
3.6 Conclusion 64

Chapter 4: NEWSSPIKE-RE: Exploiting Parallel News Streams for Unsupervised Event Extraction 65

4.1 Introduction . . . 65
4.2 System Overview . . . 67
4.3 Discovering Salient Events . . . 70
4.4 Generating the Training Sentences . . . 72
4.5 Sentential Event Extraction . . . 78
4.6 Empirical Evaluation . . . 78
4.7 Conclusions . . . 86

Chapter 5: NEWSSPIKE-SCALE: High Performance Event Extraction for Large, Extensible Ontologies with Minimal Human Effort . . . 88
5.1 Introduction . . . 88
5.2 System Overview . . . 90
5.3 Finding Trigger Phrases . . . 94
5.4 Empirical Evaluation . . . 97
5.5 Conclusion . . . 103

Chapter 6: Related Work . . . 106
6.1 Relation Extraction . . . 106
6.2 Ontology Mapping . . . 111
6.3 Paraphrasing . . . 113
6.4 Crowdsourcing and Relation Extraction Tools . . . 115

Chapter 7: Future Work . . . 117

Chapter 8: Conclusion . . . 120

Bibliography . . . 121

Appendix A: Resources for Distribution . . . 134

LIST OF FIGURES

Figure Number Page

1.1 The basic idea of ontological smoothing: given the target relations and a few training instances, we build a mapping from the target relations to the large background database or knowledge base, which contains millions of entities and thousands of relations. The mapping provides us database views, which allows us to retrieve many more instances that are deemed similar to those of the target relation. With the new instances, distant supervision can be employed to learn the extractors. . . . 14

1.2 The basic idea of exploiting the *temporal correspondence* for event extraction: the system aims to extract a set of target relations and has a large corpus of parallel news streams from multiple sources as its unlabeled corpus; the system first clusters the sentences from parallel news according to the target relations and then uses them as the training data; the system then learns the relation extractor from the generated training sentences. 19

1.3 A running example to show the general framework of proposed system: to exploit the parallel news streams for relation extraction, the system could be composed of three stages. First, extracting event candidates from the parallel news streams; second, identifying parallel sentences describing the events; third, clustering sentences from different NewsSpikes and generating the training sentences; finally, learning event extractors for the target relations with supervised or distant supervised algorithms. 22

2.1 System overview of VELVET: first, it maps target relations to background knowledge based according to the given ground tuples; second, silver training data is generated with distant supervision; third, tthe relation extractor is learned from the silver training data. 27

2.2 In order to map target relations to the background knowledge-base, one must consider a large space of possible database *views*. For example, the target `isCoachedBy` maps to the following expression over Freebase relations: $\pi_{PName,CName}$ Players $\bowtie$ PlaysForTeam $\bowtie$ Coach. In fact, the best mapping is a union of this expression with similar ones for other sports. 29

2.3 Relation extraction with minimal supervision. VELVET outperforms baseline conditions on Nell ontology. 38

2.4 Relation extraction with minimal supervision. VELVET outperforms baseline conditions on IC ontology. 39

3.1 NEWSSPIKE-PARA first applies open information extraction to articles in the news streams, obtaining shallow extractions with time-stamps. Next, an *NewsSpike* (NewsSpike) is obtained after grouping daily extractions by argument pairs. Temporal features and constraints are developed based on our temporal correspondence heuristics and encoded into a joint inference model. The model finally creates the paraphrase clusters by predicting the relation phrases that describe the NewsSpike. 48

3.2 an example model for NewsSpike (Armstrong, Livestrong, Oct 17). Y and Z are binary random variables. Φ^Y, Φ^Z and Φ^{joint} are factors. be founder of and step down come from article 1 while give speech at, be chairman of and resign from come from article 2. 53

3.3 an example pair of the output cluster and the gold cluster, and the corresponding precision recall numbers. 59

3.4 Precision recall curves on hard, diverse cases for NEWSSPIKE-PARA, w/oSpike, w/oTense and w/oDiscourse. 61

4.1 During its training phase, NEWSSPIKE-RE first groups parallel sentences as *NewsSpikes*. Next, the system automatically discovers a set of event relations. Then, a probabilistic graphical model clusters sentences from the NewsSpike as training data for each discovered relation, which is used to learn sentential event extractors. During the testing phase, the extractor takes test sentences as input and predicts event extractions. 68

4.2 A simple example of the edge-cover algorithm with K=2, where E_i are event relations and η_j are NewsSpikes. The optimal solution selects E_1 with edges to η_1 and η_2, and E_3 with edge to η_3. These two event relations cover all the NewsSpikes. . . 71

4.3 (a) The connected components depicted as plate model, where each Y is a Boolean variable for a relation phrase and each Z is a Boolean variable for a training sentence for with that phrase; (b) and (c) are example connected components for the event phrases *'s trip to* and *stay in* respectively. The goal of the model is to set $Y = 1$ for good paraphrases of a relation and to set $Z = 1$ for good training sentences. 74

4.4 Learning from Heuristic Labels 77

4.5 Precision pseudo-recall curves for all 30 event relations. NEWSSPIKE-RE has AUC 0.80, more than doubling R13 (0.30) and 35% higher than R13P (0.59) for all event relations. 82

4.6 Precision pseudo-recall curves for *buy(org, org)*, this figure includes the distant supervision algorithm MIML learned from matching the Freebase relation to The New York Times. NEWSSPIKE-RE has AUC 0.80, more than doubling R13 (0.30) and 35% higher than R13P (0.59) for *buy(org, org)* event relations. 83

5.1 System overview of NEWSSPIKE-SCALE: the system allows users to specify new event relations and add them to the target ontology. During the training phase, the system first finds a set of trigger phases for every relation; it presents the trigger phrases to users with their context; NEWSSPIKE-SCALE then employs the graphical model of NEWSSPIKE-RE to generate training sentences, which are used to learn sentential event extractors. During the testing phase, the extractor takes a test sentence as input and predicts event extractions. 91

5.2 An example of two trigger words acquire and deal for the event relation buy (organization, organization). We present the trigger words, the clusters of event phrases, the in-spike sentences and the general sentences in four different levels respectively. Users are asked to tag the trigger words. 98

5.3 Precision pseudo-recall curves for all 150 event relations. NEWSSPIKE-SCALE has AUC 0.79, 25% higher than NEWSSPIKE-RE (0.63) and 54% higher than NEWSSPIKE-BASE (0.51) 101

5.4 Comparing the robustness of three systems. For each system, we bin the Area Under the Curve (AUC) numbers by the scale of 0.1 and compute the number of event relations falling in the bins, and then show curves of the frequencies. More than half of the 150 relations have AUC above 0.90 for NEWSSPIKE-SCALE. . . . 102

6.1 Classification of selected ontology matching systems, based on Euzenat and Shvaiko[40]. 112

LIST OF TABLES

Table Number Page

1.1 A simple annotation task for three sentences with identified entities. Suppose there are two relations in the ontology *ceo(Person, Organization)* and *buy(Organization, Organization)*. 7

1.2 An example table from knowledge base with company CEOs. 12

1.3 Some example sentences from the unlabeled corpus. Each of them contains a pair of entities from Table 1.2. The entities are highlighted by boxes. 12

2.1 Approximate F1 scores averaged by relations. MULTIR outperforms baseline conditions on two target ontologies, NELL and IC. Condition "Manual" shows performance of an extractor trained on smoothed instances of the best manually constructed complex mapping from target relations to background knowledge-base. . . 40

2.2 Relation-specific Precision, Recall, F1 (estimated using M_1), and Accuracy at top-10 (checked manually) for 4 NELL and 2 IC relations. 41

2.3 MULTIR achieves performance comparable to state-of-the-art supervised approaches RY07 and CP10, when there exists an appropriate mapping to its background ontology. While RY07 and CP10 need fully labeled sentences, MULTIR learns with minimal supervision of just 10 ground instances per relation. Freebase does not offer an appropriate mapping for the `Kills` relation. 42

2.4 VELVET ontological mapping result on 4 NELL and 2 IC relations, with join, union, project and select operators. 44

3.1 Comparison with methods using parallel news corpora 59

3.2 Comparison with methods using the distributional hypothesis 63

4.1 Quality of the generated training sentences (count, micro- and macro- accuracy), where "all" includes sentences with all event phrases and "diverse" are those with distinct event phrases. 81

4.2 Performance of extractors by event relation, reporting both precision and the area under the PR curve. The # column shows the number of true extractions in the pool of sampled output. NEWSSPIKE-RE (labeled N-RE) outperforms two implementations of Riedel's Universal Schemas (See Section 4.6.3 for details). The advantage of NEWSSPIKE-RE over Universal Schemas is greatest on a diverse test set where each sentence has a distinct event phrase. 87

5.1 Example trigger words for users to tag 98

5.2 An ontology with 150 event relations 104

5.3 Performance of extractors by event relation, reporting both F1 at maximum recall and the area under the PR curve. The # column shows the number of true extractions in the pool of sampled output. NEWSSPIKE-SCALE (labeled N-scale) outperforms two implementations of NEWSSPIKE-RE (See chapter 4 for details). 105

ACKNOWLEDGMENTS

I am so lucky to have Dan Weld as my advisor. He has incredible wisdom, knowledge, passion and experience on a wide range of areas. It would never be enough for me to learn from him. He has extraordinary insight on new research directions that allow my research to make more impact. He is extremely supportive, always very kind and patient to me. Research could be challenging. When my progress is frustrated, he would meet with me several times a week to help me find a right path; when my work ran smoothly, he would encourage me with big smiles and let me enjoy my research. There are so many amazing moments to work with him.

Special thanks go to Raphael Hoffman and Stephen Soderland. I was so fortunate to collaborate with Raphael when I was new in research. He used his own behavior to let me know how to become a good Phd student and a good researcher. He is so diligent and resourceful. His data, software and techniques have been helping me throughout my research. I am very grateful to Stephen Soderland for advising me through my thesis projects. He is always there to provide me the mentorship I needed. He is so patient and so careful, help me organize my ideas and write down my thoughts. He makes the research much better with his great skills.

I am always amazed by the intelligence and humor of Oren Etzioni, Pedro Domingos and Mausam. I learned so much by attending their seminars. Once a while, they will develop some break though Artificial Intelligence techniques. I am always excited by their successes and inspired by their brilliant ideas.

I would like to thank Luke Zettlemoyer, Yejin Choi and Fei Xia, for always having time to help me, and their great efforts to create such an engaging and exciting natural language processing research environment at the University of Washington.

I had a great time as an intern at IBM Almaden Research Center. Yunyao Li, Benny Kimelfeld

and Howard Ho are amazing mentors. Yunyao is one of the most organized person I have ever met with. Benny is so smart and shape our research in some fascinating way. Howard is extremely nice and supportive. Thank you for giving me a great internship.

Special tribute to Ben Taskar, he was so kind to help me as a committee member. I was terribly saddened by his passing away. Unfortunately, I could not have more opportunities to work with such an outstanding scientist.

I am also deeply grateful to my colleagues in our department. I am having fun and fruitful collaboration with Fei Wu, Xiao Ling and Sai Zhang. I benefit a lot by discussion with Alan Ritter, Anthony Fader, Yoav Artzi, Robert Gens and other members in Loudlabs.

Many thanks to my friends. I am having so many great memories with them. They make my PhD life fascinating.

DEDICATION

to Yuan Wang, for her support and encouragement

Chapter 1

INTRODUCTION

Relation extraction, the task of extracting facts from natural language text and creating machine-readable text, is one of the great challenges of artificial intelligence. Today, the Internet contains an almost infinite amount of text and is still growing rapidly. It stores information about almost every aspect of human knowledge. Most of the information is still contained in unstructured documents. If relation extraction techniques could be successfully applied, the possible applications would be nearly endless. For instance, we could build question answering systems that extract, organize and understand billions of facts stated on web pages. The system could satisfy users' complex information needs and far exceed the abilities of today's search engines. We could develop analytic systems to automatically analyze the growing volume of professional reports, which are exploitable primarily by experts today. We could comprehend the interests and trends of billions of people by parsing their public posts and comments. Organizations and governments could gain better insights into their operations from these analytic systems. We could automatically discover facts from academic literature to support research, reasoning and hypothesis generation. We could invent advanced human-computer interaction systems that have the ability to recognize human's intentions in natural languages and grow their own machine intelligence from self-supervised reading.

Because of its great potential, relation extraction has been the subject of enormous amounts of research. However, the task is far more difficult than it seems. In fact, human-level understanding of natural language text is one of the ultimate goals of artificial intelligence. Different frameworks could be proposed to reach this ultimate goal. One promising and widely adopted idea is to split the overall goal of creating machine-readable text into several subtasks. Typical subtasks include:

- Representing the needed knowledge in an *ontology*, or a *knowledge base*. A typical ontology

contains a set of objects, which are often called *entities*, a set of classes of the entities, often called *types* and a set of *relations* between the entities and types that are related to one another.

- Recognizing entities from text, which is often called *name entity recognition* (NER). Typical NER tasks involve figuring out the boundary of the entity and assigning a unique or multiple types from the ontology to the entity.
- Extracting the relations between the entities from the unstructured text and *populating* the extracted facts to the knowledge base. Typical extraction takes an individual sentence or a set of sentences as input and outputs the relations among the entities stated in the input texts.

When an ontology is populated with the corresponding entities and relations, it is often called a *knowledge base*. Since the facts in the knowledge base are structured data, they can be queried much more easily with a formal language *e.g.* SQL, in contrast to searching in unstructured text.

Although it has been shown that machine learning approaches are promising for many natural language processing tasks, it remains very hard to develop successful extractors, especially for large ontologies. One major challenge is that traditional learning approaches require an enormous amount of human effort to annotate enough training data. For example, more than 100,000 annotated training words were provided for TAC KBP 2011, an evaluation of relation extraction for just 16 relations. Since hundreds and thousands of training examples per relation are needed, the cost to build the extractors can become incredibly high when we want to meet a user's arbitrary information need, which could easily scale up to thousands of relations. What is worse, annotators often come across extremely skewed data: the vast majority of sentences do not contain any facts about the target relations. For example, Riedal *et al.* [99] employed a dataset where the positive sentence ratio of the top 50 relations from Freebase was less than 2%. This means that after labeling one thousand sentences, the annotators might only have collected twenty positive examples. The annotation cost makes it impractical to naively employ supervised learning approaches.

To reduce the labeling effort, researchers introduced the idea of *distant supervision*, or *weak supervision*, a technique for automatically creating training data by heuristically matching the examples of an existing database's relation to text. Since the matching can be largely conducted

automatically, a vast amount of training data can be collected almost for free, in contrast to the costly manual labeling. Although distant supervision is very promising and works well in some situations, it has two major limitations. First, distant supervision only works when one has a large set of ground relation instances for every target relation. It often means the target relation must come directly from the off-the-shelf ontology. What can be done if a user wishes to quickly create an extractor, yet only has time to specify a handful of examples? Second, distant supervision is limited to relatively *static facts*, (*e.g. born-in(person,location)*) where the relevant examples exist in a corresponding knowledge base. But what about dynamic *event relations* such as *travel-to(person, location)*, when these time-dependent facts are ephemeral, and are rarely stored in pre-existing knowledge bases?

Our goal in this work is to develop techniques that allow users to create quality extractors for a wide range of relations. The target ontology could include both static relations and dynamic event relations. We will do this by exploiting the ideas of ontological smoothing and temporal correspondence. The key idea of ontological smoothing is to map target relations to knowledge bases and therefore enable distant supervision to use hidden instances that do not directly exist in the knowledge base. The key idea of temporal correspondence is to exploit the attributes of the parallel news streams to generate training data for time-dependent, event relations. Our key ideas may be summarized with the following thesis statement:

> We can create quality relation extractors with minimal human effort for a broad range of relations, including both static relations and event relations, by exploiting (1) ontological smoothing, and (2) parallel news streams.

In the rest of this introduction we first provide a broad overview of the problem of machine reading and relation extraction to provide the context of this dissertation. We then present our basic ideas of ontological smoothing and temporal correspondence and describe how they have the potential to build quality relation extractors. Finally, we discuss the contributions of this dissertation.

1.1 Relation Extraction and its Challenges

One of the great promises of artificial intelligence is to enable machines to understand text. To better explain how to create machine understandable text, we start with a simple example.

1.1.1 A Simple Use Case

Suppose there is the following snippet of text in the news on April 30th, 2015:

LinkedIn stock plunged by 25 percent today... "LinkedIn just purchased Lynda.com, an online learning company, for $1.5 billion," said Jeff Weiner, LinkedIn's CEO.

If a human were presented with this text, he would soon get many facts, such as

LinkedIn stock plunged by 25%
LinkedIn purchased Lynda.com
Lynda.com is an online learning company
LinkedIn spent $ 1.5 billion
Jeff Weiner is the CEO of LinkedIn

The ambitious goal of relation extraction is to enable the machine to have the intelligence to understand the text and get the facts like humans. But it immediately raises a challenging question: how could we represent and store those facts? For example, LinkedIn stock plunged by 25 percent also means LinkedIn stock drops 25 percent and also means LinkedIn stock is down 25 percent. Although these facts are all true and useful, it is not only inefficient but also impossible to store all of them in the knowledge base, considering that there could be trillions of documents. A much more practical method is to introduce the idea of *ontology*, which contains the relations that are interesting or useful. For example, a simple ontology could include the following relations:

stock fall (Organization, Percent)

stock rise (Organization, Percent)[1]
buy (Organization, Organization)
ceo (Person, Organization)
pay (Organization, Money)[2]

Given the ontology, the task of relation extraction becomes much more straightforward: the extraction system should recognize the facts stated in the text and fill in the argument "slots" for the relations in the ontology. For example, Linked Stock plunged by 25% stated a fact of fall (Organization, Percent). We only need to add one tuple, fall (LinkedIn, 25%), to the knowledge base. A successful extraction system would provide us the following tuples after reading the given snippet of text:

fall (LinkedIn, 25%)
buy (LinkedIn, Lynda.com)
pay (LinkedIn, $ 1.5 billion)
ceo (Jeff Weiner, LinkedIn)

When tuples are populated into the knowledge base, the corresponding facts are organized as structured data. We can easily query them to serve a large variety of purposes. The following simple example compares relation extraction with the traditional search engine, to show how relation extraction system can satisfy users' information needs in different ways.

Suppose that on April 30th, 2015, a user read in a news story that LinkedIn stock plunged 25%, and naturally wondered about the performance of other high-tech stocks. As a proficient user of search engines, he enumerated the names of the companies and submitted a list of queries like Google stock, Facebook stock, Twitter stock and so on. After reading many news stories and charts, he found out that Twitter's stock also dropped while others remained stable.

Given a relation extraction system and a knowledge base, the task would become much easier.

[1]For simplicity, we write them as fall/rise (Organization, Percent) in this dissertation.

[2]The paying event is implicit for the buying event.

The extractor has already scanned today's news article and extracted many facts. Since the user was interested in the relation fall (Organization, Percent), the knowledge base could immediately provide many tuples to him such as

fall (Twitter, 10%)
fall (Yelp, 20%)
fall (LinkedIn, 25%)

Even if the user did not think of Yelp, the knowledge base could still extract the fact because it knows that "Yelp" is an organization and its stock was down that day. Similarly, if the user is interested in the relation buy (Organization, Organization), the relation extractor, which has read billions of documents already, could provide the facts such as buy (Facebook, Oculus) and buy (Google, Waze) immediately with no need for users to submit queries.

1.1.2 Supervised Learning Framework

Given the task of relation extraction, supervised learning is a natural option to consider. In this section, we introduce the common framework of supervised learning approaches for relation extraction. We then discuss their limitations because of the special challenges of relation extraction in Section 1.1.3.

Similar to any supervised learning method, supervised relation extraction systems require us to (1) annotate the training data, (2) generate features for the data, and (3) learn the model. Then, we also need to (4) run the models on test data and (5) evaluate the performance.

What are the training and test data of relation extraction? It is a surprisingly hard question to answer. When a person reads an article, a sentence or even a single word, he would recall a considerable amount of relevant information and use that to better understand the text. So a data point of a relation extraction system could be almost anything: a sentence, an article, a bag of sentences/articles, a sentence with relevant context and so on. For example, for the purpose of populating the knowledge base, we could study bags of sentences describing the same entities. We

Relation	Sentence with Identified Entities
ceo(Person, Organization)	[Tim Cook] is the CEO of [Apple].
NA	The stakes are also high for [Google] and [Apple].
buy(Organization, Organization)	[Google]'s takeover of [Youtube].

Table 1.1: A simple annotation task for three sentences with identified entities. Suppose there are two relations in the ontology *ceo(Person, Organization)* and *buy(Organization, Organization)*.

call such an extraction system *aggregate-level* extraction, which uses overall statistical properties of the text to deduce likely relations without being able to identify any single sentence that confirms the relation by itself.

But for the fundamental purpose of relation extraction, *i.e.* understanding the text like humans and creating machine-readable data, the extractor should also work on individual sentences, *i.e.* justify any new addition to the knowledge base with a supporting sentence. For example, the system should immediately extract the fact buy(LinkedIn, Lynda.com) after reading the snippet LinkedIn just purchased Lynda.com., rather than waiting until it reads another sentence LinkedIn bought Lynda.com. We call such an extraction system *sentence-level* extraction or *sentential extraction*. It is not hard to see that sentence-level extraction is more fundamental and also more challenging. Therefore, we view sentence-level extraction as one of the most important goals of this work. Because of this goal, sentences are our major training and testing examples.

To annotate one training sentence, the annotator should label the relations stated in the text and fill in the arguments of that relation. In practice, the arguments are often name entities recognized by some name recognition system.

Table 1.1 shows a simple annotation task for three sentences. Note that the second sentence does not state any of the relations in the ontology. So it is labeled as *NA*, which means the sentence is a negative example for the model.

Similar to any other machine learning problem, relation extractors need to convert the training and testing sentences into a list of features. Features from the sentences should have the potential to describe the relationship between the arguments. For example, a bag of words representation could

be useful. A popular type of feature encodes both the dependency path between the arguments and the argument types. For example, given the sentence Tim Cook is the CEO of Apple, the name entity recognizer gives us the type of Time Cook, Apple as Person and Organization, respectively; while the syntactic parser gives us the following dependencies nsubj(CEO, Cook), nmodOf(CEO, Apple). With them, we could create a feature for the entity pair as Person$\leftarrow$ nsubj $\leftarrow$ CEO $\rightarrow$ nmodOf $\rightarrow$ Organization.

Given the labels and features of the sentences, various machine learning algorithms could be used to learn the model. The goal of the model is to predict whether any target relation is stated in the test sentence. In the testing phase, we could enumerate the entity pairs in the sentences, create features for every entity pair, and predict the relations for them. For example, the sentence [Tim Cook] succeed [Steve Jobs] as the CEO of [Apple] contains three name entities as indicated by the boxes. There could be 6 ordered candidate pairs that lead to 6 different test sentences. Each would have different sets of features and the learned model could be applied on each of them.

To understand how well a relation extraction system works, one typically measures its performance in terms of precision and recall,

$$\text{Precision} = \frac{tp}{tp + fp},$$

$$\text{Recall} = \frac{tp}{tp + fn}.$$

where tp (true positives) is the number of relation instances that are extracted and that are truly expressed in the text, fp (false positives) the number that are extracted but that are not actually expressed in the text, and fn (false negatives) the number that are expressed in the text but missed. Intuitively, high recall means that the system is able to detect most of the times that a relation is expressed in text, and high precision means that it does not confuse different relations in its output. If a single metric is desired, one can combine precision and recall to their harmonic mean, which is called *F-measure*,

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

1.1.3 Challenges

Supervised learning approaches seem to be a good fit for the problem of relation extraction. Annotation could be expensive, but one may think the cost to label thousands of sentences is still affordable. But unfortunately, the key issue is that there could be tens of thousands of *relations*. The cost to label enough training sentences for every relation is unacceptable. In addition, relation extraction has several special challenges that make simple supervised learning approaches fail to scale up in practice.

Skewed Data Challenge

First, the dataset tends to be extremely *skewed*. Given an ontology, most sentences do not express any facts of the relations in the ontology. For example, we analyzed the dataset used by Riedel *et al.* [100]. For the top 50 relations in Freebase, the ratio of the positive sentences is less than 2%. To understand how skewed data makes machine learning difficult, let us compare relation extraction with another NLP problem, sentiment analysis. Sentiment analysis involves predicting whether the text expresses positive or negative opinion. Suppose we have collected a set of comments from shopping websites, most snippets would express some attitude, either positive or negative. So if we annotate a small set of snippets, almost every label could contribute to the learning process in some way. Unfortunately, for relation extraction, annotators are easily overwhelmed by negative sentences: after labeling a thousand sentences, the annotators may have only collected dozens of positive examples. What is worse, most negative sentences carry very weak information and make almost no contribution to the learning process. Note that Freebase ontology includes some very common relations like contains (location, location). If the relation is less common (*e.g.* ceo (person, organization)), only a very tiny fraction of sentences in the dataset would state that relation, which makes the task of finding candidates for annotators to label very hard.

Synonym Challenge

A synonym is a word or phrase that means exactly or nearly the same as another word or phrase in the same language. For relation extraction, the synonym challenge means that there exist many different ways in which the same relation could be stated in text. To see how synonym challenge makes relation extraction difficult, consider the following examples:

Tim Cook, the CEO of Apple, . . .
Tim is the chief executive of Apple.
Cook who heads Apple . . .

CEO, chief executive, and head all state the same relation. If we want to learn a quality extractor, we must let the learning model receive those synonym signals. But how could we accomplish this? There are a large number of expression that are related to the concept but many of them occur very few times in the corpus. It could be very hard to find them and put them in a small set of examples so annotators are able to label them.

Polysemy Challenge

Just as the same relation could be stated by multiple expressions, there are often semantic variations of the same word or phrase. Consider the following examples:

Cook who heads Apple . . .
Lebron James heads to Cleveland
Barack Obama heads to Boston . . .

All these sentences share almost identical words and structures, but the word head has three different meanings. In the first sentence, head means Cook is the chief of the company Apple. In the second sentence, head means Lebron James joins the team Cleveland Cavaliers. In the third sentence, head means Barack Obama travels to the city Boston.

Why does polysemy make relation extraction challenging? Suppose we have Lebron James heads to Cleveland in the training data and the annotator correctly labels the relation as join (person, organization). This training example tells the model that when it sees some pattern like person head to location, it could predict join (person, organization), which is a clear over-generalization. How could we avoid such errors? We must provide negative signals (*e.g.* including Barack Obama heads to Boston in the training set) to tell the model that person head to location is not a good pattern. It is next to impossible to generate a manually labeled training set that provides enough such signals.

The skewed data, synonym, and polysemy challenges make relation extraction a very hard problem. In particular, simply applying the traditional supervised learning framework may work in some situations, but it is not promising for scaling up to large ontologies with arbitrary relations in which users are interested.

To address the challenges of relation extraction, we have exploited a knowledge base and a large amount of unlabeled corpus to develop weak supervision and ontological smoothing techniques. Furthermore, to handle event relations that do not populate the knowledge base, we have exploited parallel news streams and developed temporal correspondence techniques. We will introduce each of these in the next two sections.

1.2 Distant Supervision and Ontological Smoothing

Because of the challenges of relation extraction, it is hard to manually label a complex, sufficiently diverse gold training set for large sets of relations. A natural question arises: are there automated or semi-automated methods that allow us to generate a large training set? Unlike humans who can only read and label limited amount of examples, automatic methods could easily scan billions of sentences and have the potential to provide a large training corpus. Certainly, automatic methods would generate noisy training data, but hopefully the massive quantity could compensate for the quality. Intrigued by this motivation, researchers have proposed distant supervision, a technique that heuristically matches the contents of the knowledge base to the text in order to automatically create training data for learning extractors. Suppose we are interested in the relation ceo (person,

isCEOof	
Tim Cook	Apple
Steve Jobs	Apple
Satya Nadella	Microsoft
Larry Page	Google

Table 1.2: An example table from knowledge base with company CEOs.

[Tim Cook], the CEO of [Apple] ...
[Steve Jobs] co-founded [Apple] Computers ...
... [Google]'s chief business officer talks about how [Larry Page] has changed.
[Satya Nadella] is chief executive officer of [Microsoft].

Table 1.3: Some example sentences from the unlabeled corpus. Each of them contains a pair of entities from Table 1.2. The entities are highlighted by boxes.

organization), and the knowledge base has a table with company CEOs, and we also have a large corpus of unlabeled sentences. Table 1.3 shows some example sentences. Each of them happens to contain a pair of entities of the relation instances from Table 1.2.

The idea of distant supervision is to treat all sentences in Table 1.3 as training sentences for the relation ceo (Person, Organization). These new annotations can then be used to train a relation extractor in the same way they are used to train a supervised relation extractor. If a sentence is matched to an instance in the table, it is given a positive label; otherwise it gets a negative label.

Distant supervision holds the promise of generating vast amounts of training sentences. But unfortunately, the technique only works if we make certain assumptions. First, any sentence that contains a match for any instance in the knowledge base truly expresses that relation; second, any pair of entities should only belong to a single table in the knowledge base; third, we assume that we have a table in the knowledge base that contains instances of the relation we are interested in.

If the first assumption is violated, we have noisy annotations in the generated training data. In Table 1.3, [Steve Jobs] co-founded [Apple] Computers is a false positive sentence for relation

ceo (Person, Organization), though the fact in the knowledge base is correct and Steve Jobs was truly the CEO of Apple.

At least co-found is a related concept with ceo; the other sentence [Google]'s chief business officer talks about how [Larry Page] has changed is completely irrelevant to the relation, but it will still be used as a true positive for learning extractors.

If the second assumption is violated, an individual sentence would be annotated several times in the training set. For example, the knowledge base contains the fact ceo(Steve Jobs, Apple) and found (Steve Jobs, Apple), so the sentence [Steve Jobs] co-founded [Apple] Computers would have two labels. Such *overlapping* labels often mislead the learning algorithms: they are not true positives but are still used as positive examples by distant supervision.

To handle the problem of noisy training data and overlapping labels, we developed MULTIR. It is a system based on a new statistical model that uses multi-instance learning to combat the noise in the training data and allow multi-labels on the entity pairs to handle the overlapping relations. The system combines a sentence-level extraction model with a simple, corpus-level component for aggregating the individual facts. The techniques of MULTIR are beyond the scope of this dissertation, but are explained in the 2011 paper by Hoffmann *et al.* [55].

The violation of the first and second assumption may lead to low quality extractors, but if the third assumption is violated, distant supervision techniques could not even be used at all, since the knowledge base would not provide any examples for heuristic matching.

Even if the relation does exist in some knowledge base somewhere in the world, how could the user find this knowledge base for distant supervision? For example, the user is interested in building an extractor for parent (person, person). In Freebase, a very large knowledge base, there are tables father (person, person) and mother (person, person). In order to apply distant supervision, the user must first find the two tables and then merge their instances to create a new table. Consider another relation isCoachedBy(person/athlete, person/coach), which tells us who the coach of an athlete is. But in Freebase, we note that there are tables

baseballPlayerForTeam(baseballPlayer, baseballTeam)

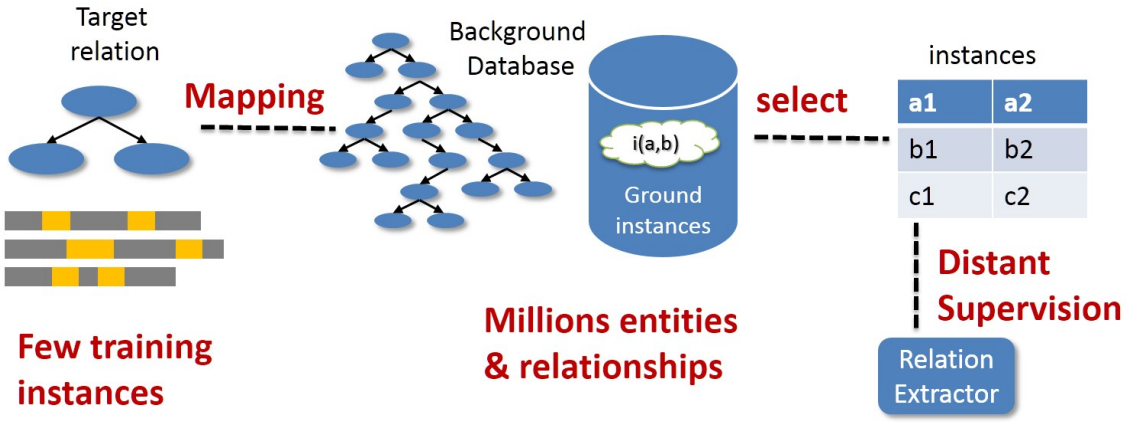

Figure 1.1: The basic idea of ontological smoothing: given the target relations and a few training instances, we build a mapping from the target relations to the large background database or knowledge base, which contains millions of entities and thousands of relations. The mapping provides us database views, which allows us to retrieve many more instances that are deemed similar to those of the target relation. With the new instances, distant supervision can be employed to learn the extractors.

baseballHeadCoach(baseballTeam, baseballCoach)

footballPlayerForTeam(footballPlayer, footballTeam)

footballCurrentCoach(footballPlayer, footballCoach)

Since the relations have been broken into separate tables for individual sports and the tables are normalized in a manner that eliminates a simple analogue, it remains unclear how to use distant supervision techniques to build the extractor.

The above examples show that distant supervision does not necessarily mean that we could build the extractors for free for any relation. In fact, it requires considerable effort to search the right knowledge base, find the right tables, and put them together to create the right instances.

What can be done if a user wishes to quickly create an extractor, yet only has time to specify a handful of examples? Could we automatically find the right instances for the relations of interest and enable the distant supervision algorithms to proceed?

To address this problem, we present VELVET, with a novel technique called ontological s-

moothing. Figure 1.1 shows the basic idea of ontological smoothing. Given the target relations and a few training instances, we build a mapping from the target relations to the large background database or knowledge base, which contains millions of entities and thousands of relations. The mapping provides us database views, which allow us to retrieve many more instances that are deemed similar to those of the target relation. With the new instances, distant supervision can be employed to learn the extractors.

The challenge of ontological smoothing is that it could be hard to map the target relation to the right tables in the background knowledge base. It is easy to see some simple keyword-based retrieval is insufficient. Instead, one should consider the large space of mappings formed by collections of database operations like join, union, project, and select. For the above parent(person, person), the ontological smoothing should return the union of two tables, *i.e.*
father (person, person) $\bigcup$ mother (person, person).

For the above isCoachedBy(person/athlete, person/coach) example, the best mapping is a union using the following expression for various sports:
playForTeam(player, team) $\bowtie$ teamCoach(team, coach)
where $\bowtie$ represents the join operator for two tables. We will present the technique details of ontological smoothing in Chapter 2.

We discussed the challenges of relation extraction in the last section. How can distant supervision and ontological smoothing be useful in handling these challenges? First, an unlabeled corpus for heuristic matching is cheap. With ontological smoothing, we would obtain a set of positive instances for every target relation, and these instances would further lead us to a set of sentences annotated as positive from the unlabeled corpus. Such a procedure helps us to overcome the skewed data challenge. Second, the large unlabeled sentence corpus contains an enormous amount of different expressions and phrases. If one expression is a good way to state a relation, it is very likely that this expression exists in the unlabeled corpus and was once used to state some facts from the knowledge base. These expressions help us to handle synonym challenges. Third, when there are multiple relations in the target ontology, sentences with the same expression but having different semantic meanings would be heuristically annotated as different relations. They

enable the learning algorithms to distinguish the polysemous meanings of the same expression.

With ontological smoothing, we significantly enhance the flexibility of distant supervision; users can now define their target relations with much more freedom, and VELVET helps the users to find relation instances and proceed with distant supervision.

Although distant supervision and ontological smoothing can work well in many situations, they have an obvious limitation: all entities and instances must come from the knowledge base. What about dynamic event relations, such as travel-to (person, location)? In the next section, we will inxtroduce a brand new idea to handle relation extraction over dynamic event relations: temporal correspondence to exploit parallel news streams.

1.3 Parallel News Streams and Temporal Correspondence

In this section, we will present an idea to exploit parallel news streams for relation extraction. As we discussed in the last section, distant supervision and ontological smoothing works for relations and instances pre-existing in the knowledge base, but they are incapable of building relation extractors for the event relations (*i.e.* fluents), such as travel-to(person, location), if no suitable knowledge base exists. We will call the task of extracting dynamic event relations *"event extraction"* in this dissertation.

Why is event extraction important? Knowledge of real-time events is crucial for making informed decisions in many fields, such as finance and politics. Indeed, stream data like news stories, blogs, and posts in social media report vast numbers of events every minute. Unlike many static facts, which are available in some knowledge bases, the facts stated in the fluent data are often exclusively contained in unstructured text. If a user is interested in the birth date of Albert Einstein, the answer is in a structured, machine-readable Wikipedia infobox because there are communities who are willing to type in birth dates for celebrities. But if he wants to know the meeting itinerary of a politician, which is expressed exclusively in the latest news articles, he must build an event extractor for meet with (person, person), let it run on recent stories, and check the output facts.

Since knowledge bases typically do not exist for event relations, one cannot use distant supervision - with or without ontological smoothing. In addition, supervised learning with human

annotated training data is also impractical for the reasons described in Section 1.1. So the question is, could we get the training signals to learn the extractors from anywhere else?

When we are interested in the relation travel to (person, location), it is straightforward to look at the sentences in news stories that contain people as the named entities and also the key phrase travel to between two persons. Fortunately, there are many such sentences:

As U.S. Secretary of State John Kerry traveled to Saudi Arabia on Tuesday . . .
. . . said Julie Bishop , who traveled to New York to negotiate . . .
Barack Obama travels to Israel for first trip as president .. . .

This means that news stories report many relevant events of the target relation, *i.e.* travel to (John Kerry, Saudi Arabia), travel to (Julie Bishop, New York), travel to (Barack Obama, Israel). But it is clear to see that it is impossible to build a high recall extractor merely with those sentences: the extractor would have no generalization ability and could only recognize the event from the sentence containing the phrase travel to.

A straightforward way to build the extractor is to use the above instances of the target relation to apply distant supervision. For example, when we know travel to (Barack Obama, Israel), we could match the entity pair Barack Obama, Israel to the unlabeled sentences and generate training data. Unfortunately, the direct usage of distant supervision will not work. It is because event relations are often strongly time dependent. When we match Barack Obama, Israel to the unlabeled corpus, we would see the following sentences:

Barack Obama restates support of Israel in synagogue speech.
US President Barack Obama told Israeli television . . .
Barack Obama faces heckler in Israel.
. . . in the relationship between the United States and Israel, Obama talked about . . .

Most of these sentences are irrelevant to the target relation travel to (person, person). Indeed, when we search Barack Obama, Israel in a search engine, not a single sentence states the travel to event on the first page! Because the true positive sentences are highly correspondent to temporal

attributes of the articles, the training data collected with simple distant supervision would become extremely noisy.

Although this naive approach is bad for distant supervision, it also brings new opportunities for relation extraction: a peculiar property, which we call *temporal correspondence*, is that when there is a spotlight event, many news articles on the same day from different sources will describe the same daily event with different text. For example, on the day when Kerry travels to Saudi Arabia, there are the following sentences in the news articles

Kerry, in Saudi visit, wins expanded Arab support
John Kerry arrived in Saudi Arabia . . .
. . . John Kerry made a previously unannounced trip to Saudi Arabia . . .

These three sentences are describing the same event travel to (John Kerry, Saudi Arabia) and using different words and phrases. If we could collect the corresponding sentences on the same day for a vast number of events, and use them as the training data for the relation travel to (person, location), we would have the opportunity to build a quality relation extractor without the expensive labeling of the training sentences.

Since news stories report events almost exclusively, we focus on news articles and use them as our corpus in this dissertation. We use the phrase *"parallel news stream"* to denote the collection of news stories published at the same time and describing the same entities.

Figure 1.3 shows the basic idea of exploiting the *temporal correspondence heuristics* for event extraction. Suppose travel to (person, location), fire (organization, person) and acquire (organization, organization) are our target relations, the system would first cluster a set of sentences for each of the target relations, and then build a relation extractor by learning from these generated training data.

Many clustering techniques have a fundamental limitation: they primarily depend on the distributional hypothesis, which states that words occurring in similar contexts tend to have similar meanings, to recognize synonyms and paraphrases. The consequence is that they tend to confuse synonym with antonym. For example, DIRT, a famous paraphrasing system, reports the closest

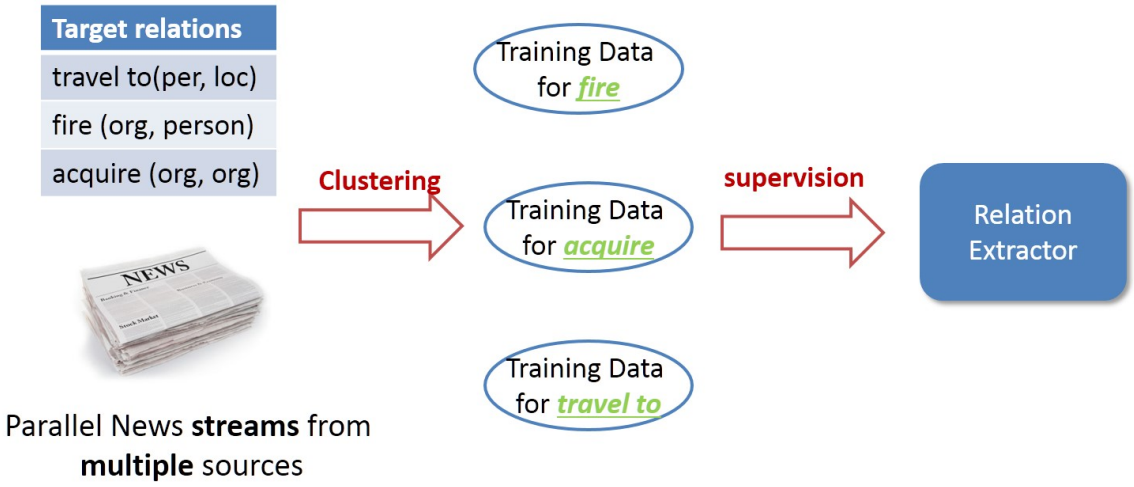

Figure 1.2: The basic idea of exploiting the *temporal correspondence* for event extraction: the system aims to extract a set of target relations and has a large corpus of parallel news streams from multiple sources as its unlabeled corpus; the system first clusters the sentences from parallel news according to the target relations and then uses them as the training data; the system then learns the relation extractor from the generated training sentences.

phrase to fall is rise, and the closest phrase to shoot is kill. In the case of relation extraction, the confusion could be terribly damaging. When a company relies on facts from relation extraction to make decisions, the confusion between fall and rise, buy and sell could cause completely opposite operations.

Fortunately, parallel news streams are helpful for handling this *antonym challenge*. This is because the temporal attributes that associate with the articles and sentences allow us to design clustering algorithms that do not primarily rely on the distributional hypothesis. Intuitively, when we see LinkedIn stock falls 10% in some news articles, we know is would be very unlikely to see LinkedIn stock rises 10% at the same time from another news report.

In this dissertation, we propose four *temporal correspondence heuristics* that characterize regularities over parallel news streams. The first heuristic comes from our basic observation: parallel sentences that share the same arguments and date tend to describe the same event. We call it *Temporal Functionality Heuristic*:

H 1 Temporal Functionality Heuristic: *News articles published at the same time that mention the same entities and use the same tense tend to describe the same events.*

Even with the Temporal Functionality Heuristic, some pairs of arguments may be a bad indicator for an event. For example, we could read the following sentences in one day's news

Barack Obama heads to the White House
Barack Obama greets reporters at the White House . . .
Barack Obama makes a speech at the White House

The argument pair (Barack Obama, the White House) suggests a mixture of events, which is bad for our purposes. We propose the *temporal burstiness heuristic* that rests on a simple observation that we can judge whether an entity pair is good for paraphrasing by looking at the history of the frequencies that the entity pair is mentioned in the news streams. Since good entity pairs tend to have a spike in their time series, we call them *NewsSpike* in this paper. That is,

H 2 Temporal Burstiness Heuristic: *If an entity or an entity pair appears significantly more frequently in one day's news than in recent history, the corresponding event candidates are likely to be good for generating paraphrases.*

However, some parallel sentences might be related but not paraphrased. We combat this problem with two heuristics. First, when journalists write news reports, they tend to avoid duplicating the same facts. We propose *one event-mention per discourse heuristics:*

H 3 One Event-Mention Per Discourse Heuristic: *A news article tends not to state the same fact more than once.*

This heuristic directs an algorithm to choose, from a news story, the single best phrase describing the event.

Second, when one of the parallel sentences contains a negated form, it suggests a non-synonymous relationship. For example, when we read

Snowden travels to Hong Kong.
Snowden cannot stay in Hong Kong as Chinese officials. . .

It is unlike that travel to and stay in are synonymous phrases because otherwise the two news stories are describing opposite events. The observation leads to:

H 4 Temporal Negation Heuristic: *Two event phrases tend to be semantically different if they co-occur in the parallel sentences, which share the argument pairs and the date, but one of them is in negated form.*

We propose NEWSSPIKE-RE, a novel relation extraction system that learns quality extractors for event relations by exploiting the temporal correspondence heuristics from parallel news streams. Figure 1.3 illustrates a running example and shows the framework of NEWSSPIKE-RE. The system is composed of the following stages:

- Crawling news streams from multiple sources: We were unable to find any suitable time-stamped, parallel new corpus, so we collected data by ourselves.
- Extracting event candidates: We process the parallel news corpus through the NLP pipeline and identify the name entities from the sentences. We group sentences sharing the same entity pairs and date together and use them as the event candidates. For example, (Snowden, Hong Kong, August 1st) is an event candidate;
- Selecting paraphrasing: Some event candidates such as (Obama, Senate, Oct 4th) are not good for the purpose of paraphrasing; some sentences sharing the argument pair and the date do not describe the main event. So we must have algorithms to find good sentences describing the event.
- Generating training sentences: Sentences from different NewsSpikes can describe the same event relation. To learn extractors with high precision and recall, the system must cluster the sentences from different NewsSpikes together. For example, the sentences describing two facts travel to (Obama, Miami) and travel to (Snowden, Hong Kong) should be clustered

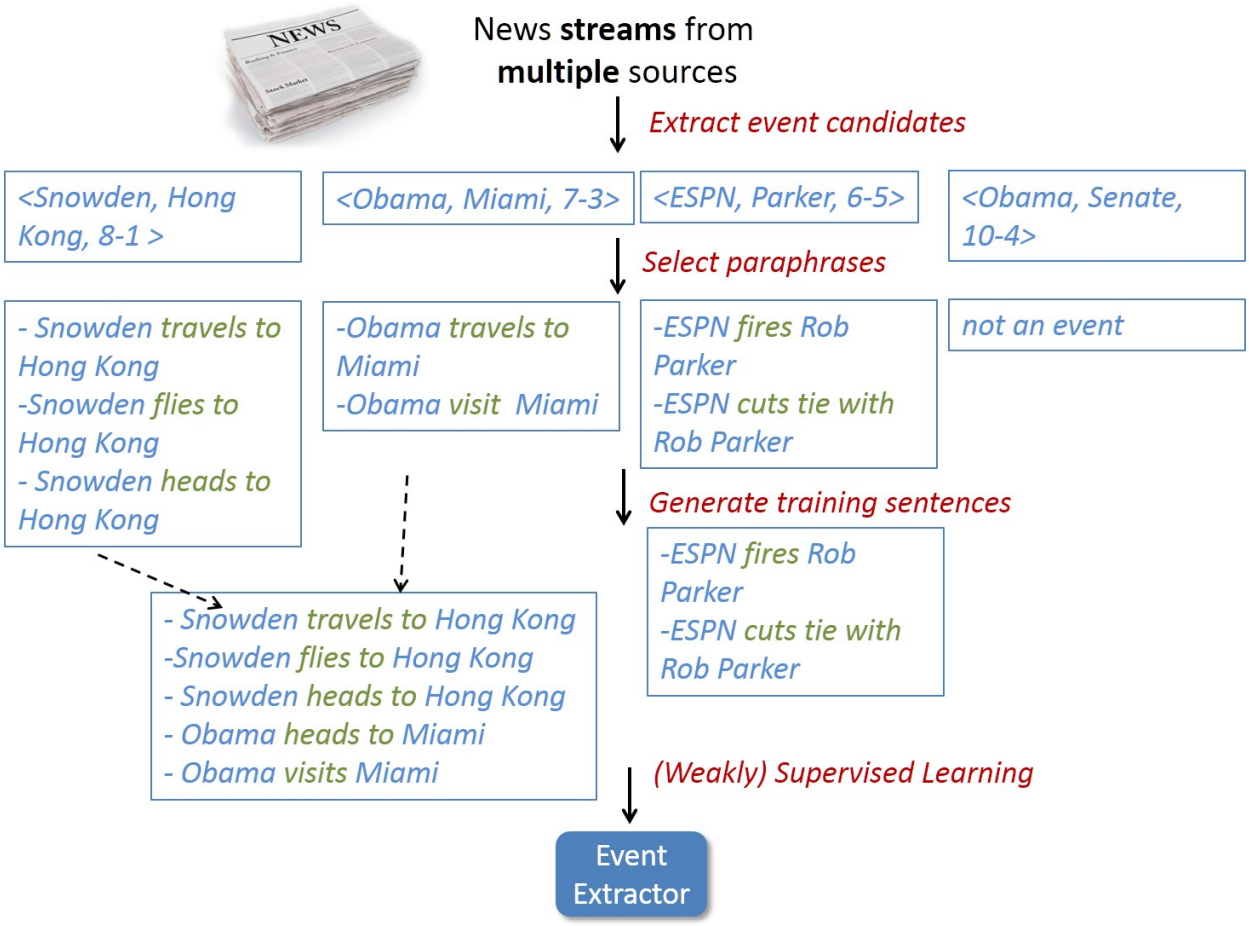

Figure 1.3: A running example to show the general framework of proposed system: to exploit the parallel news streams for relation extraction, the system could be composed of three stages. First, extracting event candidates from the parallel news streams; second, identifying parallel sentences describing the events; third, clustering sentences from different NewsSpikes and generating the training sentences; finally, learning event extractors for the target relations with supervised or distant supervised algorithms.

together as the training sentences for the target relation travel to (person, location).

- Learning the relation extractors: Given the training sets generated in the previous stage, the system could employ various learning algorithms to create the output extractors.

In Chapter 3, we will present our solution NEWSSPIKE-PARA to collect the parallel corpus and to select the paraphrases. In Chapter 4, we will present an unsupervised solution NEWSSPIKE-RE to learn the extractors for the most salient events in the news articles. In Chapter 5, we will extend NEWSSPIKE-RE to NEWSSPIKE-SCALE , an extraction system for a large set of relations, and allow users to specify target relations with flexibility.

1.4 Contributions

The previous sections outlined the challenges that we need to overcome if we want to build high-performance extractors for a large variety of relations. We introduced our major ideas to overcome the challenges: we can apply distant supervision and ontological smoothing for the static relations, and exploit parallel news streams and temporal correspondence heuristics for the event relations.

To realize these goals, we need to accurately map users' relations to the database views from a knowledge base; we need to generate high quality paraphrases and recognize sentences describing the same events; we need to cluster events together and generate the training set in pursuit of high precision high recall event extractors; we need to scale the extractors to a large set of relations and allow user-specified event relations.

The goal of this dissertation is to provide solutions to these problems. This dissertation presents the design, implementation, and evaluation of several novel techniques that enable high performance relation extraction:

VELVET**: Ontological Smoothing for Relation Extraction.** Existing distant supervision only works when one has a large set of ground relation-instances (tuples) for the relations of interest. What can be done if a user wishes to quickly create an extractor, yet only has time to specify a handful of examples? We present *ontological smoothing*, a semi-supervised technique that learns extractors for a set of minimally-labeled relations. Ontological smoothing has three phases.

First, it generates a mapping between the target relations and a background knowledge base using database join, union, project, and select operators. Second, it uses distant supervision to heuristically generate new training examples for the target relations. Finally, it learns an extractor from a combination of the original and newly-generated examples. Experiments on 65 relations across three target domains show that ontological smoothing can dramatically improve precision and recall, even rivaling fully supervised performance in many cases. We describe the details of VELVET in Chapter 2.

NEWSSPIKE-PARA**: Harvesting Parallel News Streams to Generate Paraphrases of Event Relations**: Existing paraphrasing techniques have considered generating paraphrases by mining the Web, guided by the *distributional hypothesis*. They tend to confuse antonyms with synonyms because antonymous phrases appear in similar contexts as often as synonymous phrases. We formulate a set of three *temporal correspondence heuristics* that characterize regularities over parallel news streams. We develop a novel program, NEWSSPIKE-PARA, based on a probabilistic graphical model that jointly encodes these heuristics. We present inference and learning algorithms for our model. We present a series of detailed experiments demonstrating that NEWSSPIKE-PARA outperforms several competitive baselines, and show through ablation tests how each of the temporal heuristics affects performance. To spur further research on this topic, we provide both our generated paraphrase clusters and a corpus of time-stamped news articles collected from hundreds of news sources. We describe the details of NEWSSPIKE-PARA in Chapter 3

NEWSSPIKE-RE**: Exploiting Parallel News Streams for Unsupervised Event Extraction.** Existing relation extraction approaches are either based on supervised learning and limited by scarce training data, or based on distant supervision and limited to the static relations from pre-existing knowledge bases. We present NEWSSPIKE-RE, a novel, unsupervised algorithm that discovers event relations and then learns to extract them. We develop a method to discover a set of distinct, salient event relations from news streams. We describe an algorithm to exploit parallel news streams to cluster sentences that belong to the same event relations. In particular, we propose the *temporal negation heuristic* to avoid conflating co-occurring but non-synonymous phrases. We introduce a probabilistic graphical model to generate training for a sentential event extractor with-

out requiring any human annotations. We present a series of detailed experiments demonstrating that the event extractors learned from the generated training data significantly outperform several competitive baselines, *e.g.* our system more than doubles the area under the micro-averaged, PR curve (0.80 vs. 0.30) compared to Riedel's Universal Schemas. We describe the details of NEWSSPIKE-RE in Chapter 4

NEWSSPIKE-SCALE**: high performance event extraction for large ontologies with minimal human effort.** What if a user wishes to flexibly input target relations beyond the salient relations discovered by NEWSSPIKE-RE and wishes to create extractors for large ontologies? We present NEWSSPIKE-SCALE, a semi-supervised algorithm that learn high performance event extractors for the user-specified relations with minimal human efforts. We present an algorithm to automatically find a list of most informative triggers for a relation; the user can then tag the top trigger words as positive or negative. We present a series of experiments showing that, with a few minutes annotation efforts per relation, the event extractors learned from the generated training data can achieve high and robust performance on a large set of event relations. We describe the details of NEWSSPIKE-SCALE in Chapter 5.

Chapter 2

VELVET: ONTOLOGICAL SMOOTHING FOR RELATION EXTRACTION

Relation extraction, the process of converting natural language text into structured knowledge, is increasingly important. Most successful techniques use supervised machine learning to generate extractors from sentences that have been manually labeled with the relations' arguments. Unfortunately, these methods require numerous training examples, which are expensive and time-consuming to produce. As a result, most extractors are never tested on more than a handful of relations.

This chapter presents *ontological smoothing*, a semi-supervised technique that learns extractors for a set of minimally-labeled relations. Ontological smoothing has three phases. First, it generates a mapping between the target relations and a background knowledge-base. Second, it uses distant supervision to heuristically generate new training examples for the target relations. Finally, it learns an extractor from a combination of the original and newly-generated examples. Experiments on 65 relations across three target domains show that ontological smoothing can dramatically improve precision and recall, even rivaling fully supervised performance in many cases.

2.1 Introduction

Vast quantities of information are encoded on the Web in natural language. In order to render this information into structured form for easy analysis, researchers have developed methods for *relation extraction* (RE). The most successful RE techniques use supervised machine learning to generate extractors from a training corpus comprised of sentences which have been manually labeled with the arguments of the target relations. Unfortunately, these supervised methods require hundreds or thousands of training examples per relation, and thus have proven too expensive for use in

Figure 2.1: System overview of VELVET: first, it maps target relations to background knowledge based according to the given ground tuples; second, silver training data is generated with distant supervision; third, tthe relation extractor is learned from the silver training data.

constructing Web-scale knowledge bases.

To address this problem, researchers introduced the idea of *distant supervision*, a technique for automatically creating training data by heuristically matching the contents of a database relation to text ([31]). For example, if one has a table of athletes and their coaches that included the relation instance (Jelani Jenkis, Urban Meyer), then a system can automatically create a silver training example for isCoachedBy from the following sentence: " 'Our captain, Jelani Jenkins, saved the day' said head coach, Urban Meyer." The training examples are called 'Silver' because these examples likely contain noise and aren't as valuable as 'gold standard' examples.

However, distant supervision only works when one has a large set of ground relation-instances (tuples) for the relation of interest. What can be done if a user wishes to quickly create an extractor, yet only has time to specify a handful of examples?

This chapter presents VELVET, a novel technique called *ontological smoothing*, that addresses this problem, improving both precision and recall. MULTIR learns extractors from a set of minimal labeled relations by exploiting a large background knowledge-base and unlabeled textual corpus.

As shown in Figure 2.1, VELVET works in three phases: the first step uses the few examples to generate a mapping from the target relation to a database *view* over a background knowledge-base, such as Freebase. The second step queries the background knowledge-base to retrieve many more instances that are deemed similar to those of the the target relation; these are heuristically matched to the textual corpus to create myriad silver training examples. Finally, in the third step, VELVET learns an extractor.

It is challenging to find the best mapping from a target relation to a large background knowledge-base. Simply choosing the most similar background relation is insufficient. Instead, one should consider the large space of mappings formed by collections of database operations like join, union, project and select. For example, even though Freebase is extremely comprehensive, with considerable information about athletics, the relations have been broken into separate tables for individual sports, and the schemata have been normalized in a manner that eliminates a simple analogue to isCoachedBy (Figure 2.2). For this reason, and because of Freebase's massive size, it is challenging for an average user to construct good mappings manually, since an accurate mapping requires choosing from myriad multi-join queries candidates. Secondly, one must jointly map relation, type and entity. Often a user wishes to extract several interrelated relations. VELVET uses probabilistic joint inference over a set of Markov logic constraints to find the best global mapping.

In summary, VELVET makes the following contributions:

1. We introduce ontological smoothing, a novel approach for learning relation extractors given minimal supervision.

2. Our approach is based on a new ontology mapping algorithm, which uses probabilistic joint inference on schema- and instance-level features to explore the space of complex mappings defined using database join, union, project and select operators.

3. We present experiments on 65 target relations across three ontologies, using Freebase as background knowledge, that demonstrate that ontological smoothing provides order-of-magnitude improvements over unsmoothed approaches and rivals fully supervised performance in many

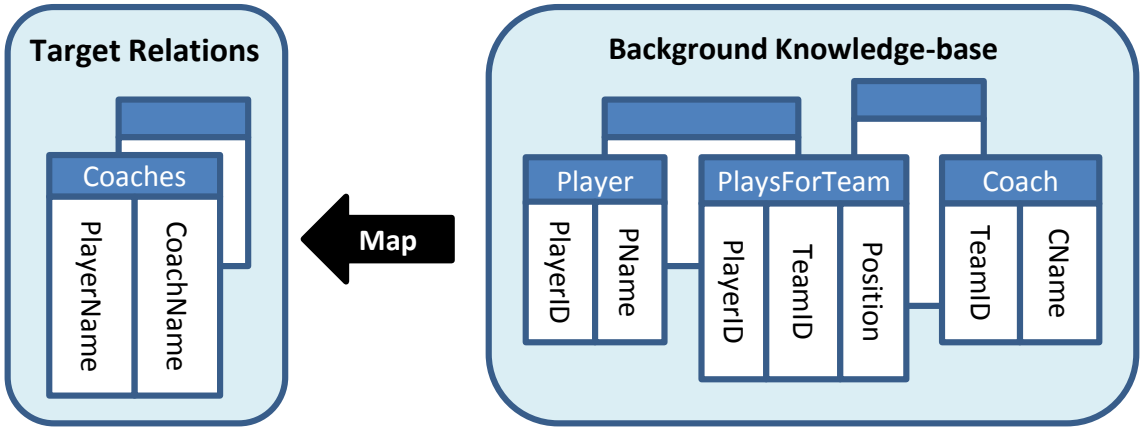

Figure 2.2: In order to map target relations to the background knowledge-base, one must consider a large space of possible database *views*. For example, the target `isCoachedBy` maps to the following expression over Freebase relations: $\pi_{PName,CName}$ Players $\bowtie$ PlaysForTeam $\bowtie$ Coach. In fact, the best mapping is a union of this expression with similar ones for other sports.

cases.

2.2 Constructing Ontological Mappings

The key intuition underlying ontological smoothing is that by finding a mapping from a user-specified target relation to a background knowledge-base, a system can automatically generate extra training data and improve the learned extractors. The key challenge is automatic construction of a good mapping from the target ontology to the background knowledge-base.

We assume that the target ontology is defined in terms of unary types $\mathtt{T}$ and binary relations $\mathtt{R}$. We express the selectional preference (i.e. type constraint) of a binary relation by $\mathtt{R}(\mathtt{T}_1, \mathtt{T}_2)$. For example, isCoachedBy (athlete,coach) is a relation in the NELL ontology [23]. We assume that each target relation comes with a set of labeled relation instances (tuples), denoted $\mathtt{R}(\mathtt{E}_1, \mathtt{E}_2)$. We also assume the presence of a large knowledge-base, $\mathcal{K}$, which is comprised of many types and relations and is populated with many instances (entities and ground relation instances); we denote these $\mathtt{t}$, $\mathtt{r}$, $\mathtt{e}$, and $\mathtt{r}(\mathtt{e}_1, \mathtt{e}_2)$ respectively.

A *mapping* between a target relation, $\texttt{R}(\texttt{T}_1, \texttt{T}_2)$, and $\mathcal{K}$, denoted $\phi(\texttt{R}, \mathcal{K})$, is a SQL expression over types and relations in $\mathcal{K}$'s schema; this expression defines a virtual relation, called a database *view*. We use a subset of SQL equivalent to relational algebra and sometimes use that notation for brevity; the symbols $\bowtie$, $\cup$, π and σ stand for database join, union, project and select operators. Given a target ontology, some ground instances of its relations, and a background knowledge-base, the *ontology mapping problem* is the task of producing a mapping for each target, R, such that the instances of $\phi(\texttt{R}, \mathcal{K})$ are semantically similar to those of R.

Ontology mapping is difficult because the space of possible views is huge. For example, Freebase contains more than 10,000 binary relations. Even if one restricts expressions to two joins with no unions or selections, there are more than 10^{12} possibilities. But selections are very important, as the following example illustrates. Suppose the target relation is stadiumInCity and consider following views:

$$\texttt{SELECT } \texttt{e}_1, \texttt{e}_2 \texttt{ FROM containedBy} \tag{2.1}$$

$$\begin{aligned}\texttt{SELECT } &\texttt{e}_1, \texttt{e}_2 \texttt{ FROM containedBy}, \texttt{sportsFacility}, \texttt{city} \\ &\texttt{WHERE containedBy.e}_1 = \texttt{sportsFacility.e} \\ &\texttt{AND containedBy.e}_2 = \texttt{city.e}\end{aligned} \tag{2.2}$$

The second view is a subset of the first and denotes a relation with very different semantics. In order for ontological smoothing to improve extractor performance, it's important to map as many ground instances as possible, but not too many! If MULTIR mapped facts about cities in states and rivers in countries (as well as stadium locations), extractor precision would plummet.

To create good mappings, VELVET considers constraints between binary relations, unary types and entities — finding analogues for all three of these elements at the same time. We describe this process below, but one intuitive example is "If entity E in the target ontology corresponds to e in $\mathcal{K}$, then the type of E should correspond to the type of e." These constraints are described in Markov logic which combines the expressiveness of first order logic with a clear probabilistic

semantics [96].

At the highest level, VELVET uses a two-stage approach to find the best mappings.

- We first restrict the set of views under consideration; this process, candidate generation, is described in the next subsection.
- VELVET next uses probabilistic joint inference to select the most likely global mapping from the candidates for each target relation, type and entity; our probability model and inference algorithm are described in the following subsections.

2.2.1 Generating Mapping Candidates

The first step in mapping construction is defining a set of *candidate mappings* for each of the target entities, types and binary relations; later these are ranked. Our model generates a set of Markov logic rules over special predicates and their negations: $\texttt{Cnddt}(\texttt{e}, \texttt{E})$ means that the mapping between $\texttt{E}$ and $\texttt{e}$ is in consideration, and the probability of $\texttt{Mp}(\texttt{e}, \texttt{E})$ signifies the quality of the mapping. We use a hard rule to ensure that two entities will only be mapped if they have similar names ($\texttt{Syn}$ stands for synonym):

$$\texttt{Syn}(\texttt{e}, \texttt{E}) \Rightarrow \texttt{Cnddt}(\texttt{e}, \texttt{E}) \tag{2.3}$$

The next rule encodes the intuition that when two entities possibly match then their types might also match.

$$\texttt{Cnddt}(\texttt{e}, \texttt{E}) \wedge \texttt{Tp}(\texttt{e}, \texttt{t}) \wedge \texttt{Tp}(\texttt{E}, \texttt{T}) \Rightarrow \texttt{Cnddt}(\texttt{t}, \texttt{T}) \tag{2.4}$$

Here, $\texttt{Tp}(\texttt{e}, \texttt{t})$ indicates $\texttt{t}$ is the type of $\texttt{e}$ in $\mathcal{K}$. The same notation applies for target terms: $\texttt{Tp}(\texttt{E}, \texttt{T})$.

We now turn to binary relations, such as $\texttt{R}(\texttt{T}_1, \texttt{T}_2)$. VELVET only considers mapping $\texttt{R}$ into views of the following form: $\cup\chi(\texttt{t}_1, \texttt{t}_2)$ where $\cup$ denotes union; χ is a join of up to 4 binary relations in $\mathcal{K}$; $\texttt{t}_\texttt{i} = \phi(\texttt{T}_\texttt{i})$ specify selection operations that only allow instances whose $\mathcal{K}$ entity arguments have types corresponding to the selectional preferences of the target, $\texttt{T}_\texttt{i}$. Our next hard

rule forces a candidate join path over $\mathcal{K}$ to contain at least one instance that is also present in the target relation.

$$\begin{aligned} &\texttt{Inst}(\texttt{R}, (\texttt{E}_1, \texttt{E}_2)) \wedge \texttt{Inst}(\chi, (\texttt{e}_1, \texttt{e}_2)) \\ \wedge &\texttt{Syn}(\texttt{e}_1, \texttt{E}_1) \wedge \texttt{Syn}(\texttt{e}_2, \texttt{E}_2) \Rightarrow \texttt{Cnddt}(\chi, \texttt{R}) \end{aligned} \tag{2.5}$$

The term, $\texttt{Inst}(\texttt{R}, (\texttt{E}_1, \texttt{E}_2))$, means that the tuple, $\texttt{R}(\texttt{E}_1, \texttt{E}_2)$, is a ground instance of target relation R. $\texttt{Inst}(\chi, (\texttt{e}_1, \texttt{e}_2))$ means that $\texttt{e}_1$ and $\texttt{e}_2$ are elements in a row of χ, which was created by joining several relations from $\mathcal{K}$.

Our last hard rules specify that only candidates can be considered as mappings (we show the case for binary relations, but similar rules govern type and entity mappings):

$$\texttt{Mp}(\chi, \texttt{R}) \Rightarrow \texttt{Cnddt}(\chi, \texttt{R}) \tag{2.6}$$

2.2.2 Specifying the Likelihood of Mappings

We now describe our model for ascribing the probability of mappings. Here we use the full power of Markov logic. Unfortunately, our treatment must be brief. The probability of a truth assignment to the Cnddt and Mp predicates is given by

$$P(x) = \frac{exp(\sum_i w_i n_i(x))}{Z_x}$$

where Z_x is a normalization constant, w_i is the weight of the ith rule, and n_i is the number of satisfied groundings of the rule. See [96] for details.

Consistency between Relations, Types and Entities: If many ground instances are shared between a target relation and its image under a mapping, then that suggests that the mapping is good. One might *think* that one could encode this as:

$$\begin{aligned}&\texttt{Inst}(\texttt{R}, (\texttt{E}_1, \texttt{E}_2)) \wedge \texttt{Inst}(\chi, (\texttt{e}_1, \texttt{e}_2))\\ \wedge&\texttt{Mp}(\texttt{e}_1, \texttt{E}_1) \wedge \texttt{Mp}(\texttt{e}_2, \texttt{E}_2) \Rightarrow \texttt{Mp}(\chi, \texttt{R})\end{aligned} \tag{2.7}$$

Unfortunately, this encoding causes problems. While this rule may look similar to Equation 2.5, this one affects the probability of both entity and relation mappings, since the probability of $\texttt{Mp}(\texttt{e}, \texttt{E})$ is also being inferred while synonyms (used in Equation 2.5) are taken as ground-truth inputs. The problem with Equation 2.7 is that it can cause VELVET to lower the probability of an (otherwise good) entity-entity mapping, whenever it dislikes a mapping between binary relations that involve those entities. Instead, we wish the inference to go one way: if many ground instances map, then the relations should be likely to map, but not vise versa. This is encoded as:

$$\begin{aligned}&\texttt{Mp}(\texttt{e}_1, \texttt{E}_1) \wedge \texttt{Mp}(\texttt{e}_2, \texttt{E}_2) \wedge \texttt{Inst}(\texttt{R}, (\texttt{E}_1, \texttt{E}_2))\\ \wedge&\left(\vee_{\texttt{k}=1}^{\texttt{K}} \texttt{Inst}(\chi_\texttt{k}, (\texttt{e}_1, \texttt{e}_2))\right) \wedge \left(\vee_{\texttt{k}=1}^{\texttt{K}} \texttt{Mp}(\chi_\texttt{k}, \texttt{R})\right)\end{aligned} \tag{2.8}$$

Note that we've replaced $\Rightarrow$ with $\wedge$ to avoid negative "information flow." We use disjunction $\vee$ among $\texttt{Mp}(\chi_\texttt{k}, \texttt{R})$ to handle overlapped relations. Note Equation 2.8 is not symmetric between χ and $\texttt{R}$; this is because the target ontology is usually small and its relations do not overlap. We specify a similar rule for types:

$$\texttt{Mp}(\texttt{e}, \texttt{E}) \wedge \texttt{Tp}(\texttt{E}, \texttt{T}) \wedge \left(\vee_{\texttt{k}=1}^{\texttt{K}} \texttt{Tp}(\texttt{e}, \texttt{t}_\texttt{k})\right) \wedge \left(\vee_{\texttt{k}=1}^{\texttt{K}} \texttt{Mp}(\texttt{t}_\texttt{k}, \texttt{T})\right)$$

Negative instance constraints: When specifying a target ontology, it is sometimes possible to declare a closed-world assumption, specify exclusion between types or otherwise present negative examples. Since these can greatly improve the quality of a mapping, we include the following hard

rule:

$$\texttt{Inst}(\chi, (\texttt{e}_1, \texttt{e}_2)) \wedge \texttt{NegInst}(\texttt{R}, (\texttt{E}_1, \texttt{E}_2))$$
$$\wedge \texttt{Mp}(\texttt{e}_1, \texttt{E}_1) \wedge \texttt{Mp}(\texttt{e}_2, \texttt{E}_2) \Rightarrow \neg\texttt{Mp}(\chi, \texttt{R}) \tag{2.9}$$

Unlike the Equation 2.8, we use $\Rightarrow$ because when $\texttt{Mp}(\chi, \texttt{R})$, $\texttt{Inst}(\chi, (\texttt{e}_1, \texttt{e}_2))$ is true but $(\texttt{E}_1, \texttt{E}_2)$ is a negative instance of R, it is very unlikely that the entity mappings are correct.

Length of Join: While joining binary relations over the background ontology greatly extends the representational ability of the views, it may also add noise from arbitrary cross products. To combat this, we add a soft rule $\texttt{short}(\chi) \Rightarrow \texttt{Mp}(\chi, \texttt{R})$, enforcing a preference for views with a small number of joins.

Unique Entities: We assume that the background knowledge base is of high quality, with little duplication among entities. This justifies the following hard rule: $\texttt{Mp}(\texttt{e}, \texttt{E}) \Rightarrow \neg\texttt{Mp}(\texttt{e}', \texttt{E})$.

Regularization: According to Ockham's Razor, VELVET should avoid predictions with weak evidence. We add soft rules for type and relation mappings: $\neg\texttt{Mp}(\texttt{t}, \texttt{T})$ and $\neg\texttt{Mp}(\chi, \texttt{R})$. With respect to entity mappings, the unique entities rules achieve regularization.

2.2.3 Maximum a Posteriori Inference

Finding a solution to $\arg\max_{\mathbf{x}} P(\mathbf{x})$ is challenging. One issue is the scale of our problem: we would like to assign truth values to thousands of grounded predicates, but our problem, which is equivalent to the weighted Maximum Satisfiability problem, is NP-hard. Furthermore, the dependencies encoded in our rules break the joint distribution into islands of high-probability states with no paths between them — a challenge for local search algorithms.

One way of solving $\arg\max_{\mathbf{x}} P(\mathbf{x})$ is to cast it into an integer linear program [82]. Although the integer linear program is intractable in our case, we can compute an approximation in polynomial time by relaxing the problem to a linear program and using randomized rounding, as proposed by [130]. For solving the linear program we use MOSEK with the interior-point optimization method.

Firstly, every grounding of the rule is converted into conjunctive normal form, denoted as $CNF_i = \wedge c_j$, where c_j is a clause. Let c_j^+ and c_j^- be the set of indices of the variables that appear in the positive and negative form in clause c_j, and let H be the set of indices of hard rules. The inference problem can be relaxed as:

$$\max \sum w_i z_i \tag{2.10}$$

$$s.t. \sum_{k \in C_j^+} y_k + \sum_{k \in C_j^-} (1 - y_k) \geq 1, \ i \in H \tag{2.11}$$

$$\sum_{k \in C_j^+} y_k + \sum_{k \in C_j^-} (1 - y_k) \geq z_i, \ i \notin H \tag{2.12}$$

$$y_k, z_i \in [0, 1]$$

where y_k indicates the truth assignment of the predicate, and z_i indicates whether one rule is satisfied. Equation 2.11 ensures hard rules to be satisfied, and Equation 2.12 allows soft rules to be broken but z_i will get smaller value then. Theoretically, when $w_i = 1$, the LP-relaxation is $3/4$-approximation algorithm.

2.3 Relation Extraction

After mapping the target relations into the background knowledge-base $\mathcal{K}$, MULTIR applies distant supervision [31] to heuristically match both seed relation instances and relation instances of the mapped relations, to corresponding text.

For example, if $\mathtt{r}(\mathtt{e}_1, \mathtt{e}_2) =$ isCoachedBy(Jenkins, `Meyer`) is a relation instance and `s` is a sentence containing synonyms for both $\mathtt{e}_1 =$Jenkins and $\mathtt{e}_2 =$Meyer, then `s` may be a natural language expression of the fact that $(\mathtt{e}_1, \mathtt{e}_2) \in \mathtt{r}$ holds and could be a useful training example.

Unfortunately, this heuristic can often lead to noisy data and poor extraction performance. To fix this problem, Riedel *et al.* [100] cast distant supervision as a form of multi-instance learning, assuming only that *at least one* of the sentences containing $\mathtt{e}_1$ and $\mathtt{e}_2$ are expressing $(\mathtt{e}_1, \mathtt{e}_2) \in \mathtt{r}$.

In our work, we use the publicly available MultiR system [53] which generalizes Riedel *et al.*'s method with a faster model that also allows relations to overlap. MULTIRuses a probabilistic, graphical model that combines a sentence-level extraction component with a simple, corpus level

component for aggregating the individual facts. MULTIR's extraction decisions are almost entirely driven by sentence-level reasoning. However, by defining random aggregate-level variables Y for individual facts and tying them to the sentence-level variables Z for extractions, a direct method for modeling weak supervision is provided. The model is trained, so that the aggregate variables Y match the facts in the database, treating the sentence-level variables Z as hidden variables that can take any value, as long as they produce the correct aggregate predictions.

During learning, MULTIRuses a Perceptron-style additive parameter update scheme which has been modified to reason about hidden variables, similar in style to the approaches of [136, 64]. To support learning, MULTIRperforms a greedy approximation to a weighted, edge-cover problem for inference.

Training examples and their features are computed following [80]. On each sentence, we first run a statistical tagger to identify named entities and their types. Each pair of entity annotations is then considered as an extraction candidate, with features being conjunctions of the inferred entity types and paths of syntactic dependencies between the entity annotations.

For tagging named entities, we use the system by [70]. Since it outputs fine-grained entity types based on the Freebase type system, we can enforce consistency by considering only examples where the types of the tagger agree with those inferred in the mapping phase. We found that this step improves efficiency and leads to more accurate extractions. For computing syntactic dependencies we use Stanford Dependency Parser [75].

2.4 Empirical Evaluation

In our experiments we examine (1) the impact of smoothing on the quality of relational extractors, (2) the quality of relation extraction using VELVET compared to supervised systems, and (3) the quality of ontological mappings inferred by VELVET.

We test on two publicly available target ontologies with seed instances. Our experiments will show that ontological smoothing substantially improves the performance of the relation extractor. It is true across many target relations, each of which is only described by a small set of labeled instances.

2.4.1 Experimental Setup

In this paper, MULTIR uses Freebase [18] as the background knowledge-base $\mathcal{K}$, which contains dozens millions of entities and tens of thousands of relations across many domains. For the unlabeled corpus, we use the New York Times [103] which contains over 1.8 million news articles published between Jan. 1987 and Jun. 2007. For practicality, we make two simplifications. First, similar to Suchanek *et al.* [116], the weights for VELVET soft rules are simply set to a fix weight 1 to ensure our setting general enough. Second, we limit the size of join computations. In particular, we remove candidate joins if there exists a setting of the join attributes that yields more than $10,000$ join tuples.

2.4.2 Relation Extraction with Smoothing

We compare VELVET to the following baseline conditions:

w/oS "without smoothed instances": Learns extractors from ground relation instances only; makes no use of background knowledge-base $\mathcal{K}$.

w/oC "without complex mappings": Maps each target relation to a single atomic relation in the background knowledge-base, that covers most ground relation instances. Type information is ignored. One-to-one mappings are also known as *alignments*.

w/oJ "without joint inference". Computes a complex mapping of target relations to the background knowledge-base involving $\bowtie$, π, and σ operators. (Note there is no obvious way to handle $\cup$ operators, without joint-inference or learning thresholds.) First, each target relation and each target type are assigned the background relations and types which cover most ground instances. Then, type constraints are enforced by taking appropriate joins.

We conduct experiments on relations of two target ontologies: NELL and IC. The **NELL ontology** [23][1] contains 118 binary relations, but only 52 relations have a small number of positive

[1] `http://rtw.ml.cmu.edu/aaai10_online/relations.xls`

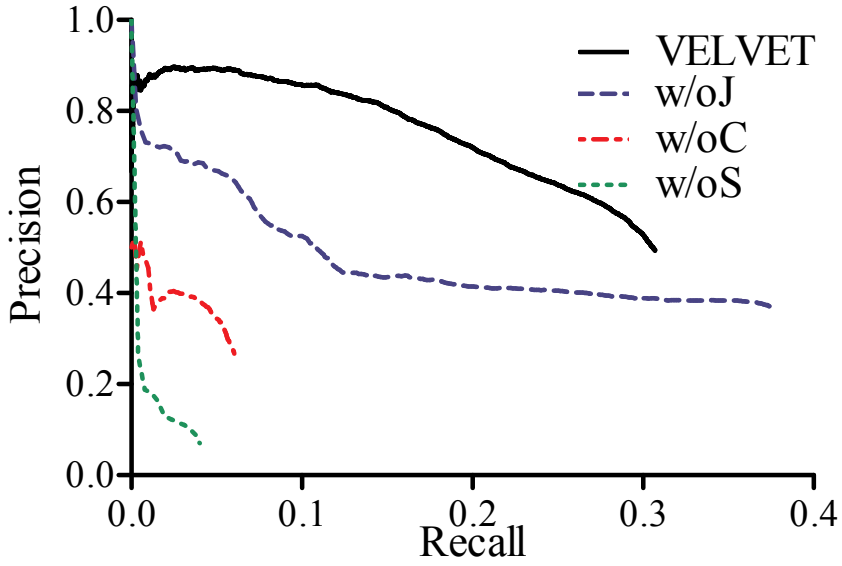

Figure 2.3: Relation extraction with minimal supervision. VELVET outperforms baseline conditions on Nell ontology.

ground instances. Many of these also have negative instances. The arguments for binary relations are typed. In total, the ground instances cover 40 different entity types and 829 unique entities. The **IC ontology** is derived from the IC dataset of the Linguistic Data Consortium[2]. The dataset contains annotations of news articles relevant to the intelligence domain. The IC ontology contains 9 binary relations, and we collected 388 positive ground instances from the annotated articles of the dataset.

We note that it is difficult to create a test set with enough gold annotations, since mentions of these 61 relations tend to be sparse. Thus we adopt the (semi-)automatic evaluation metric used in [100], which we call M_1. For each target relation, we estimate precision and recall by comparing two answer sets, Δ and Δ_V. Δ_V represents the set of predicted relation instances; Δ represents the set of relation instances in our background knowledge-base. In our work, we compute Δ by

[2] `LDC2010E07, theMachineReadingP1ICTrainingDataV3.1`

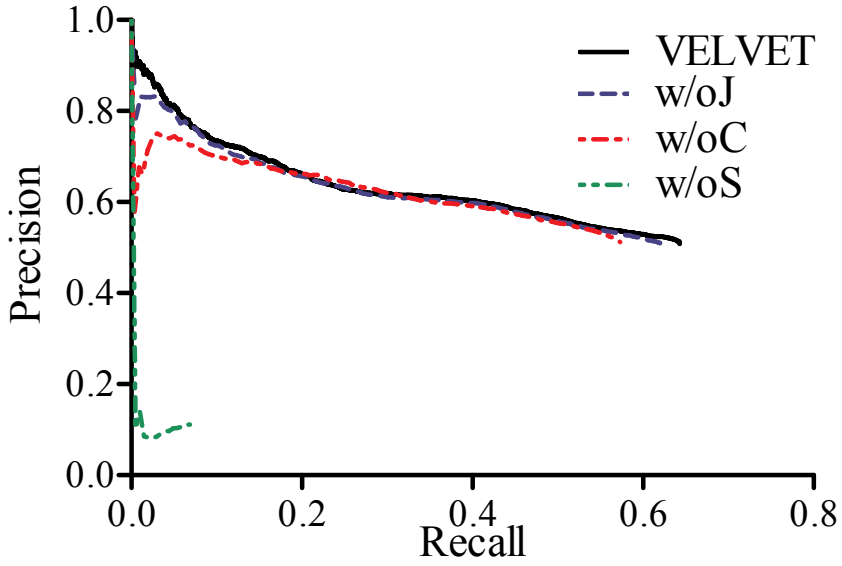

Figure 2.4: Relation extraction with minimal supervision. VELVET outperforms baseline conditions on IC ontology.

manually creating the best gold mapping from a target relation into the background knowledge-base using any combination of relational algebra operators, and then retrieving all instances. When aggregating over multiple relations, M_1 averages over instances.

Figure 2.3 and 2.4 show precision and recall curves. The poor performance of "w/oS" is due to the fact that there exist only few ground instances for each target relation, and often even fewer ground instances can be matched to sentences.

Smoothing, however, dramatically improves performance. We further observe that complex mappings are important: w/oC which only finds an alignment performs worse than w/oJ or VELVET. Upon inspection, we noticed that w/oC often maps to over-general relations. For example, background relation containedBy is mapped to target relation stadiumInCity. We therefore need type constraints, but not only type constraints: The fact that VELVET outperforms w/oJ shows that VELVET's abilities to do joint inference and support $\cup$ operators are also crucial.

Ontology	w/oS	w/oC	w/oJ	MULTIR	Manual
NELL	7.2	18.1	25.1	**27.1**	31.6
IC	11.3	37.9	39.4	**40.9**	41.4

Table 2.1: Approximate F1 scores averaged by relations. MULTIR outperforms baseline conditions on two target ontologies, NELL and IC. Condition "Manual" shows performance of an extractor trained on smoothed instances of the best manually constructed complex mapping from target relations to background knowledge-base.

Although M_1 allows (semi-)automatic evaluation on millions of sentences, it has two drawbacks: Since M_1 averages over instances, relations with many instances contribute more to the overall score than sparse relations. Furthermore, the metric only provides a conservative estimate of performance when the knowledge-base is incomplete. We therefore also evaluate MULTIR using additional metrics.

Table 2.1 compares MULTIR to our baseline conditions, averaged over relations rather than instances. The relative comparisons are consistent with our observations so far. Note, however, that averaging over relations tends to give lower numbers than averaging over instances, because the system can learn more accurately from relations with more instances.

Table 2.2 shows a breakdown of results per relation. Precision, recall, and F1 are estimated using the conservative metric M_1, but we also report top-K accuracy for $K = 10$. For each relation we took the top ten extractions for which our extractor was most confident and manually checked correctness. We obtained results in the high $70\% - 100\%$ range.

2.4.3 Comparing to Supervised Extraction

In this section, we want to show that MULTIR can achieve performance comparable to state-of-the-art supervised approaches but with much less supervision. For this experiment, we choose a standard dataset for which there exist numerous annotations.

We use the **Conll04** relation extraction dataset[3] [102]. The sentences in this dataset were taken

[3] `http://cogcomp.cs.illinois.edu/Data/ER/conll04.corp`

Relation	Rec	Pre	F1	Acc@top-10
bookWriter	31.8	43.5	36.7	100%
headquarterIn	19.1	60.1	28.9	90%
isCoachedBy	28.9	10.3	15.3	70%
stadiumInCity	51.9	77.6	62.6	100%
attendSchool	69.4	44.4	54.2	80%
isLedBy	33.8	49.7	40.2	100%

Table 2.2: Relation-specific Precision, Recall, F1 (estimated using M_1), and Accuracy at top-10 (checked manually) for 4 NELL and 2 IC relations.

from the TREC corpus and were fully annotated with entities, types and relations. There are five relations and four entity types. We use the same experimental settings as previous work [59, 102] to enable direct comparison. In this setting, there are 1437 sentences and about $18,000$ instances. However, unlike the supervised approaches, we only provide MULTIR 10 ground instances per relation and no sentence-level annotations.

MULTIR's ontology mapping component finds correct mappings for four relations, `LocatedIn`, `OrgBasedIn`, `WorkFor`, `LiveIn`, and correctly determines that Freebase does not offer an appropriate mapping for the `Kills` relation.

Table 2.3 compares MULTIR's relation extraction performance to that of CP10 [59] and RY07 [102]. When MULTIR finds correct mappings, it achieves comparable performance to the state-of-the-art supervised approach CP10 and RY07. However, MULTIR achieves this result with only a small set of labeled ground instances, while CP10 and RY07 used more than one thousand labeled sentences. Of course, MULTIR only works if the target relation has an analogue in the background KB.

2.4.4 Ontological Mapping Quality

Finally, we analyze the performance of our ontology mapping component in more detail. Note that solving the mapping problem requires finding a joint assignment to a considerable number of variables: for NELL, we computed truth values for 3055 entity mapping candidates, 252 type mapping candidates, and 729 relation mapping candidates. For the IC domain, these are 1552, 130,

Relation	MULTIR			CP10	RY07
	Rec	Pre	F1	F1	F1
`Kills`	33.4	29.4	31.3	75.2	**79.0**
`LiveIn`	65.8	49.4	**69.2**	62.9	53.0
`LocatedIn`	64.0	65.4	**64.7**	58.3	51.3
`OrgBasedIn`	67.4	47.1	55.5	**64.7**	54.3
`WorkFor`	61.8	78.5	69.1	**70.7**	53.1

Table 2.3: MULTIR achieves performance comparable to state-of-the-art supervised approaches RY07 and CP10, when there exists an appropriate mapping to its background ontology. While RY07 and CP10 need fully labeled sentences, MULTIR learns with minimal supervision of just 10 ground instances per relation. Freebase does not offer an appropriate mapping for the `Kills` relation.

and 256, respectively.

We investigate accuracy for entity, type and relation mappings by manually validating the individual decisions. Note that our algorithm does not always return a mapping element in the background knowledge-base $\mathcal{K}$ for an element in the target ontology. This often makes sense, since Freebase, although large, does not cover all entities, types or relations. It turned out that MULTIR achieves 87.9% accuracy on relation mapping, 90.9% on type mapping and 92.9% on entity mapping. As a baseline, we use a Freebase internal search API to map entities in the target ontology to entities in Freebase. This baseline gets 88.5% accuracy, which means joint inference in MULTIR results in a reduction of 30% of entity mapping errors.

Table 2.4 shows the results of mapping six relations to Freebase. MULTIR is able to accurately recover relations composed by multiple select, project, join, and union operations. The results show that our ontology mapping algorithm returns meaningful mappings, thus ensuring the robustness of the overall system.

2.5 Conclusion

Relation extraction has the potential to enable improved search and question-answering applications by transforming information encoded in natural language on the Web into structured form.

Unfortunately, most successful techniques for relation extraction are based on supervised learning and require hundreds or thousands of training examples; these are expensive and time-consuming to produce. This paper presents ontological smoothing, a novel method for learning relational extractors, that requires only minimal supervision. Our approach is based on a new ontology-mapping algorithm, which uses probabilistic joint inference over schema- and instance-based features to search the space of views defined using SQL selection, projection, join and union operators. Experiments demonstrate the method's promise, improving both precision and recall. Our MULTIR system learned significantly better extractors for 65 relations in three target ontologies and rivals fully supervised performance in many cases.

Target Relation	Mapped View
bookWriter	$\pi_{\texttt{1.name,2.name}}$/book/book^1 $\bowtie$ /book/written_work/author
	$\bowtie$ /en/author^2 $\cup$ $\pi_{\texttt{3.name,4.name}}$/film/film^3
	$\bowtie$ /film/film/story_by $\bowtie$ /en/author^4
headquarterIn	$\pi_{\texttt{1.name,2.name}}$/business/business_operation^1
	$\bowtie$ /organization/organization/headquarters
	$\bowtie$ /location/mailing_address/citytown
	$\bowtie$ /location/citytown^2
	$\cup\pi_{\texttt{3.name,4.name}}$/organization/organization^3
	$\bowtie$ /organization/organization/headquarters
	$\bowtie$ /location/mailing_address/citytown
	$\bowtie$ /location/citytown^4
isCoachedBy	$\pi_{\texttt{1.name,2.name}}$
	/american_football/football_player^1
	$\bowtie$ /american_football/football_player/passing
	$\bowtie$ /american_football/player_passing_statistics/team
	$\bowtie$ /american_football/football_team/current_head_coach
	$\bowtie$ /en/head_coach^2
	$\cup\pi_{\texttt{3.name,4.name}}$ /en/basketball_player^3
	$\bowtie$ /basketball/basketball_player/team
	$\bowtie$ /basketball/basketball_roster_position/team
	$\bowtie$ /basketball/basketball_team/head_coach
	$\bowtie$ /basketball/basketball_coach^4
stadiumInCity	$\pi_{\texttt{1.name,2.name}}$/sports/sports_facility^1
	$\bowtie$ /location/location/containedby $\bowtie$ /location/citytown^2
attendSchool	$\pi_{\texttt{1.name,2.name}}$ /people/person^1
	$\bowtie$ /people/person/education $\bowtie$ /education/education/institution
	$\bowtie$ /education/university^2
isLedBy	$\pi_{\texttt{1.name,2.name}}$ /government/political_party^1
	$\bowtie$ /government/political_party/politicians_in_this_party
	$\bowtie$ /government/political_party_tenure/politician
	$\bowtie$ /government/politician^2 $\cup\pi_{\texttt{3.name,4.name}}$
	/location/country^3 $\bowtie$ /government/governmental_jurisdiction/governing_officials
	$\bowtie$ /government/government_position_held/office_holder $\bowtie$ /government/politician^4

Table 2.4: VELVET ontological mapping result on 4 NELL and 2 IC relations, with join, union, project and select operators.

Chapter 3

NEWSSPIKE-PARA: HARVESTING PARALLEL NEWS STREAMS TO GENERATE PARAPHRASES OF EVENT RELATIONS

The distributional hypothesis, which states that words that occur in similar contexts tend to have similar meanings, has inspired several Web mining algorithms for clustering semantically equivalent phrases. Unfortunately, these methods have several drawbacks, such as confusing synonyms with antonyms and causes with effects. In this chapter, we introduce three Temporal Correspondence Heuristics, that characterize regularities in parallel news streams, and shows how they may be used to generate high precision paraphrases for event relations. We encode the heuristics in a probabilistic graphical model to create the NEWSSPIKE-PARA algorithm for mining news streams. We present experiments demonstrating that NEWSSPIKE-PARA significantly outperforms several competitive baselines. In order to spur further research, we provide a large annotated corpus of time-stamped news articles as well as the paraphrases produced by NEWSSPIKE-PARA.

3.1 Introduction

Paraphrasing, the task of finding sets of semantically equivalent surface forms, is crucial to many natural language processing applications, including relation extraction [17], question answering [42], summarization [12] and machine translation [22]. While the benefits of paraphrasing have been demonstrated, creating a large-scale corpus of high precision paraphrases remains a challenge — especially for event relations.

Many researchers have considered generating paraphrases by mining the Web guided by the *distributional hypothesis*, which states that words occurring in similar contexts tend to have similar meanings [49]. For example, DIRT [67] and Resolver [134] identify synonymous relation phrases by their distributions of the arguments. However, the distributional hypothesis has several

drawbacks. First, it can confuse antonyms with synonyms because antonymous phrases appear in similar contexts as often as synonymous phrases. For the same reasons, it also often confuses causes with effects. For example, DIRT reports that the closest phrase to `fall` is `rise`, and the closest phrase to `shoot` is `kill`.[1] Second, the distributional hypothesis relies on statistics over large corpora to produce accurate similarity statistics. For example, Resolver only targets relations appearing at least 25 times. It remains unclear how to accurately cluster less frequent relations with the distributional hypothesis.

Another common approach employs the use of parallel corpora. News articles are an interesting target, because there often exist articles from different sources describing the same daily events. This peculiar property allows the use of the temporal assumption, which assumes that phrases in articles published at the same time tend to have similar meanings. For example, the approaches by Dolan *et al.* [37] and Barzilay *et al.* [10] identify pairs of sentential paraphrases in similar articles that have appeared in the same period of time. While these approaches use temporal information as a coarse filter in the data generation stage, they still largely rely on text metrics in the prediction stage. This not only reduces precision, but also limits the discovery of paraphrases with dissimilar surface strings.

The goal of our research is to develop a technique to generate paraphrases for large numbers of event relation with high precision, using only minimal human effort. The key to our approach is a joint cluster model using the temporal attributes of news streams, which allows us to identify semantic equivalences of event relation phrases at greater precision. In summary, this chapter makes the following contributions:

- We formulate a set of three *temporal correspondence heuristics* that characterize regularities over parallel news streams.
- We develop a novel program, NEWSSPIKE-PARA, based on a probabilistic graphical model that jointly encodes these heuristics. We present inference and learning algorithms for our model.

---

[1] `http://demo.patrickpantel.com/demos/lexsem/paraphrase.htm`

- We present a series of detailed experiments demonstrating that NEWSSPIKE-PARA outperforms several competitive baselines, and show through ablation tests how each of the temporal heuristics affects performance.
- To spur further research on this topic, we provide both our generated paraphrase clusters and a corpus of 0.5M time-stamped news articles[2], collected over a period of 50 days from hundreds of news sources.

3.2 System Overview

The main goal of this work is to generate high precision paraphrases for relation phrases. News streams are a promising resource, since articles from different sources tend to use semantically equivalent phrases to describe the same daily events. For example, when a recent scandal hit, headlines read:

Armstrong steps down from Livestrong

Armstrong resigns from Livestrong

Armstrong cuts ties with Livestrong

From these we can conclude that the following relation phrases are semantically similar:

step down from

resign from

cut ties with

To realize this intuition, our first challenge is to represent an event. In practice, a question like "What happened to Armstrong and Livestrong on Oct 17?" could often lead to a unique answer. It implies that using an argument pair and a time-stamp could be an effective way to identify an event (*e.g.* (Armstrong, Livestrong, Oct 17) for the previous question). Based on this observation, this paper introduces a novel mechanism to paraphrase relations as summarized in Figure 3.2.

NEWSSPIKE-PARA first applies the ReVerb open information extraction (IE) system [41] on

[2] http://homes.cs.washington.edu/~clzhang/emnlp2013release.zip

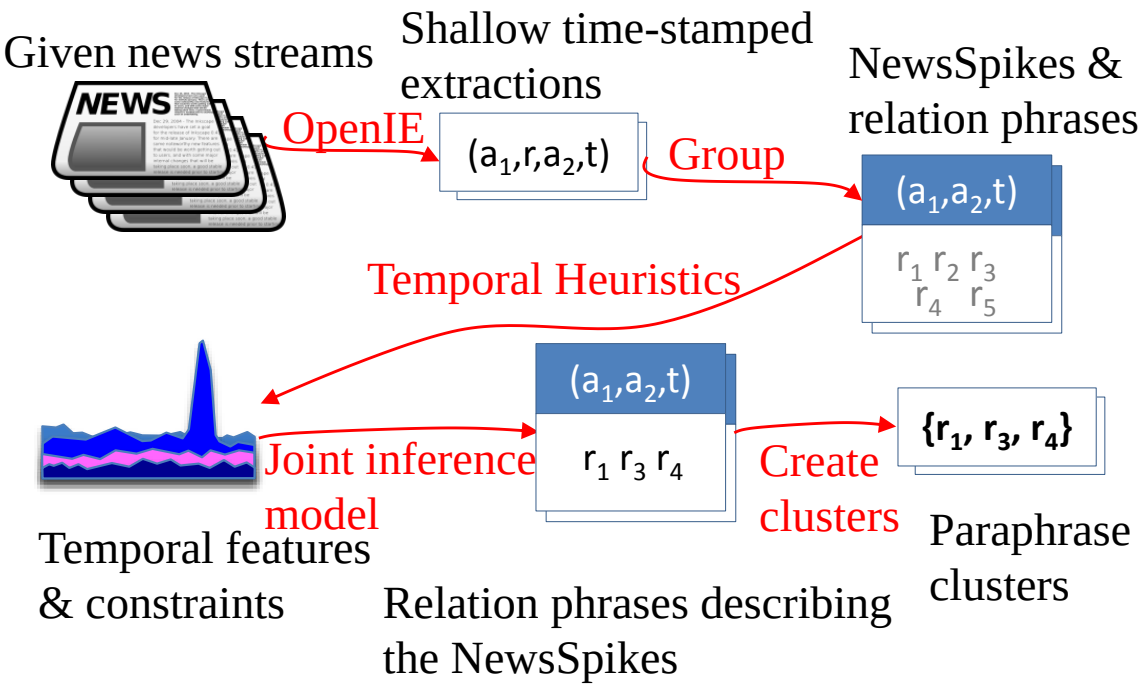

Figure 3.1: NEWSSPIKE-PARA first applies open information extraction to articles in the news streams, obtaining shallow extractions with time-stamps. Next, an *NewsSpike* (NewsSpike) is obtained after grouping daily extractions by argument pairs. Temporal features and constraints are developed based on our temporal correspondence heuristics and encoded into a joint inference model. The model finally creates the paraphrase clusters by predicting the relation phrases that describe the NewsSpike.

the news streams to obtain a set of (a_1, r, a_2, t) tuples, where the a_i are the arguments, r is a relation phrase, and t is the time-stamp of the corresponding news article. When (a_1, a_2, t) suggests a real word event, the relation r of (a_1, r, a_2, t) is likely to describe that event (*e.g.* (Armstrong, resign from, Livestrong, Oct 17). We call every (a_1, a_2, t) an NewsSpike (NewsSpike), and every relation describing the event an *event mention*.

For each NewsSpike (a_1, a_2, t), suppose there are m extraction tuples $(a_1, r_1, a_2, t) \ldots (a_1, r_m, a_2, t)$ sharing the values of a_1, a_2, and t. We refer to this set of extraction tuples as the *NewsSpike-set*, and denote it $(a_1, a_2, t, \{r_1 \ldots r_m\})$. All the event mentions in the NewsSpike-set may be semantically equivalent and are hence candidates for a good paraphrase cluster.

Thus, the paraphrasing problem becomes a prediction problem: for each relation r_i in the NewsSpike-set, does it or does it not describe the hypothesized event? We solve this problem in two steps. The next section proposes a set of temporal correspondence heuristics that partially characterize semantically equivalent NewsSpike-sets. Then, in Section 3.4, we present a joint inference model designed to use these heuristics to solve the prediction problem and to generate paraphrase clusters. The basic idea is, we frame our relation clustering problem as finding these semantically equivalent relations in the NewsSpike-set, and then generate the relation cluster.

3.3 Temporal Correspondence Heuristics

In this section, we propose a set of temporal heuristics that are useful to generate paraphrases at high precision. Our heuristics start from the basic observation mentioned previously — events can often be uniquely determined by their arguments and time. Additionally, we find that it is not just the *publication time* of the news story that matters, the *verb tenses* of the sentences are also important. For example, the two sentences

"Armstrong was the chairman of Livestrong"

"Armstrong steps down from Livestrong"

have past and present tense respectively, which suggests that the relation phrases are less likely to describe the same event and are thus not semantically equivalent. To capture these intuitions, we propose the *Temporal Functionality Heuristic*:

Temporal Functionality Heuristic: *News articles published at the same time that mention the same entities and use the same tense tend to describe the same events.*

Unfortunately, we find that not all the event candidates, (a_1, a_2, t), are equally good for paraphrasing. For example, today's news might include both

"Barack Obama heads to the White House."

"Barack Obama greets reporters at the White House".

Although the two sentences are highly similar, sharing $a_1 =$ "Barack Obama" and $a_2 =$ "White House," and were published at the same time, they describe different events.

From a probabilistic point of view, we can treat each sentence as being generated by a particular hidden event which involves several actors. Clearly, some of these actors, like Obama, participate in many more events than others, and in such cases we observe sentences generated from a *mixture* of events. Since two event mentions from such a mixture are much less likely to denote the same event or relation, we wish to distinguish them from the better (semantically homogeneous) NewsSpikes like the (Armstrong, Livestrong) example. The question becomes "How one can distinguish good entity pairs from bad?"

Our method rests on the simple observation that an entity which participates in many different events on one day is likely to have participated in events in recent days. Therefore we can judge whether an entity pair is good for paraphrasing by looking at the *history of the frequencies* that the entity pair is mentioned in the news streams, which is the *time series* of that entity pair. The time series of the entity pair (Barack Obama, the White House) tends to be high over time, while the time series of the entity pair (Armstrong, Livestrong) is flat for a long time and suddenly spikes upwards on a single day. This observation leads to:

Temporal Burstiness Heuristic: *If an entity or an entity pair appears significantly more frequently in one day's news than in recent history, the corresponding event candidates are likely to be good to generate paraphrase.*

The temporal burstiness heuristic implies that a good NewsSpike (a_1, a_2, t) tends to have a *spike* in the time series of its entities a_i, or argument pair (a_1, a_2), on day t.

However, even if we have selected a good NewsSpike for paraphrasing, it is likely that it con-

tains a few relation phrases that are related to (but not synonymous with) the other relations included in the NewsSpike. For example, it's likely that the news story reporting "Armstrong steps down from Livestrong." might also mention "Armstrong is the founder of Livestrong." and so both "steps down from" and "is the founder of" relation phrases would be part of the same NewsSpike-set. Inspired by the idea of one sense per discourse from [44], we propose:

One event-mention per discourse heuristic: *A news article tends not to state the same fact more than once.*

The one event-mention per discourse heuristic is proposed in order to gain precision at the expense of recall — the heuristic directs an algorithm to choose, from a news story, the single "best" relation phrase connecting a pair of two entities. Of course, this doesn't answer the question of deciding which phrase is "best." In Section 3.4.3, we describe how to learn a probabilistic graphical model which does exactly this.

3.4 Exploiting the Temporal Heuristics

In this section we propose several models to capture the temporal correspondence heuristics, and discuss their pros and cons.

3.4.1 Baseline Model

An easy way to use an NewsSpike-set is to simply predict that all r_i in the NewsSpike-set are event-mentions, and hence are semantically equivalent. That is, given NewsSpike-set $(a_1, a_2, t, \{r_1 \dots r_m\})$, the output cluster is $\{r_1 \dots r_m\}$.

This baseline model captures the most of the temporal functionality heuristic, except for the tense requirement. Our empirical study shows that it performs surprisingly well. This demonstrates that the quality of our input for the learning model is good: the NewsSpike-sets are promising resources for paraphrasing.

Unfortunately, the baseline model cannot deal with the other heuristics, a problem we will remedy in the following sections.

3.4.2 Pairwise Model

The temporal functionality heuristic suggests we exploit the tenses of the relations in an NewsSpike-set; while the temporal burstiness heuristic suggests we exploit the time series of its arguments. A pairwise model can be designed to capture them: we compare pairs of relations in the NewsSpike-set, and predict whether each pair is synonymous or non-synonymous. Paraphrase clusters are then generated according to some heuristic rules (*e.g.* assuming transitivity among synonyms). The tenses of the relations and time series of the arguments are encoded as features, which we call *tense features* and *spike features* respectively. An example tense feature is whether one relation is past tense while the other relation is present tense; an example spike feature is the covariance of the time series.

The pairwise model can be considered similar to paraphrasing techniques which examine two sentences and determine whether they are semantically equivalent [38, 112]. Unfortunately, these techniques often based purely on text metrics and does not consider any temporal attributes. In section 4.6, we evaluate the effect of applying these techniques.

3.4.3 Joint Cluster Model

The pairwise model has several drawbacks: *1)* it lacks the ability to handle constraints, such as the mutual exclusion constraint implied by the one-mention per discourse heuristic; *2)* ad-hoc rules, rather than formal optimizations, are required to generate clusters containing more than two relations.

A common approach to overcome the drawbacks of the pairwise model and to combine heuristics together is to introduce a joint cluster model, in which heuristics are encoded as features and constraints. Data, instead of ad-hoc rules, determines the relevance of different insights, which can be learned as parameters. The advantage of the joint model is analogous to that of cluster-based approaches for coreference resolution (CR). In particular, a joint model can better capture constraints on multiple variables and can yield higher quality results than pairwise CR models [93].

We propose an undirected graphical model, NEWSSPIKE-PARA, which jointly clusters rela-

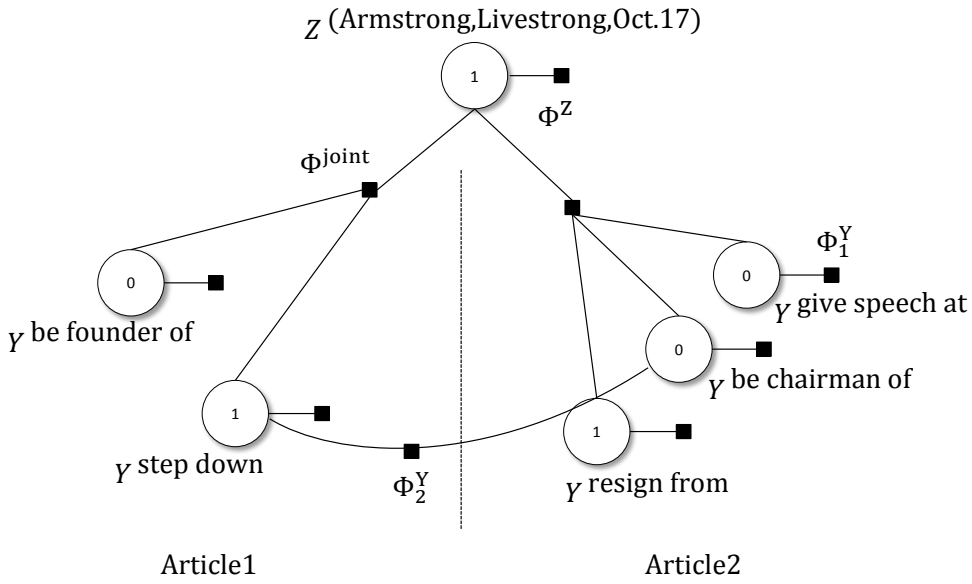

Figure 3.2: an example model for NewsSpike (Armstrong, Livestrong, Oct 17). Y and Z are binary random variables. Φ^Y, Φ^Z and Φ^{joint} are factors. be founder of and step down come from article 1 while give speech at, be chairman of and resign from come from article 2.

tions. Constraints are captured by factors connecting multiple random variables. We introduce random variables, the factors, the objective function, the inference algorithm, and the learning algorithm in the following sections. Figure 3.2 shows an example model for NewsSpike (Armstrong, Livestrong, Oct 17).

Random Variables

For the NewsSpike-set $(a_1, a_2, t, \{r_1, \ldots r_m\})$, we introduce one event variable and m relation variables, all boolean valued. The event variable $Z^{(a_1,a_2,t)}$ indicates whether (a_1, a_2, t) is a good event for paraphrasing. It is designed in accordance with the temporal burstiness heuristic: for the NewsSpike $(\texttt{BarackObama}, \texttt{theWhiteHouse}, \texttt{Oct17})$, Z should be assigned the value 0.

The relation variable Y^r indicates whether relation r describes the NewsSpike (a_1, a_2, t) or not (*i.e.* r is an event-mention or not). The set of all event-mentions with $Y^r = 1$ define a paraphrase cluster, containing relation phrases. For example, the assignments

$$Y^{\textit{step down}} = Y^{\textit{resign from}} = 1$$

produce a paraphrase cluster step down, resign from.

Factors and the Joint Distribution

In this section, we introduce a conditional probability model defining a joint distribution over all of the event and relation variables. The joint distribution is a function over *factors*. Our model contains *event factors*, *relation factors* and *joint factors*.

The event factor Φ^Z is a log-linear function with spike features, used to distinguish good events. A relation factor Φ^Y is also a log-linear function. It can be defined for individual relation variables (*e.g.* Φ_1^Y in Figure 3.2) with features such as whether a relation phrase comes from a clausal complement[3]. A relation factor can also be defined for a pair of relation variables (*e.g.* Φ_2^Y in Figure 3.2) with features capturing the pairwise evidence for paraphrasing, such as if two relation phrases have the same tense.

The joint factors Φ^{joint} are defined to apply constraints implied by the temporal heuristics. They play two roles in our model: *1)* to satisfy the temporal burstiness heuristic, when the value of the event variable is false, the NewsSpike is not appropriate for paraphrasing, and so all relation variables should also be false; and *2)* to satisfy the one-mention per discourse heuristic, at most one relation variable from a single article could be true.

We define the joint distribution over these variables and factors as follows. Let $\mathbf{Y} = (Y^{r_1} \dots Y^{r_m})$ be the vector of relation variables; let $\mathbf{x}$ be the features. The joint distribution is:

[3]Relation phrases in clausal complement are less useful for paraphrasing because they often do not describe a fact. For example, in the sentence `HeheardRomneyhadwontheelection`, the extraction (Romney, had won, the election) is not a fact at all.

$$p(Z = z, \mathbf{Y} = \mathbf{y}|\mathbf{x}; \Theta) \stackrel{\text{def}}{=} \frac{1}{Z_x}\Phi^Z(z, \mathbf{x}) \times \prod_d \Phi^{\text{joint}}(z, \mathbf{y}_d, \mathbf{x}) \prod_{i,j} \Phi^Y(y_i, y_j, \mathbf{x})$$

where $\mathbf{y}_d$ indicates the subset of relation variables from a particular article d, and the parameter vector Θ is the weight vector of the features in Φ^Z and Φ^Y, which are log-linear functions; *i.e.*,

$$\Phi^Y(y_i, y_j, \mathbf{x}) \stackrel{\text{def}}{=} \exp\left(\sum_j \theta_j \phi_j(y_i, y_j, \mathbf{x})\right)$$

where ϕ_j is the jth feature function.

The joint factors Φ^{joint} are used to apply the temporal burstiness heuristic and the one event-mention per discourse heuristic. Φ^{joint} is zero when the NewsSpike is not good for paraphrasing, but some $y^r = 1$; or when there is more than one r in a single article such that $y^r = 1$. Formally, it is calculated as:

$$\Phi^{\text{joint}}(z, \mathbf{y}_d, \mathbf{x}) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } z = 0 \wedge \exists y^r = 1 \\ 0 & \text{if } \sum_{y^r \in \mathbf{y}_d} y^r > 1 \\ 1 & \text{otherwise} \end{cases}$$

Maximum a Posteriori Inference

The goal of inference is to find the predictions $z, \mathbf{y}$ which yield the greatest probability, *i.e.*,

$$z^*, \mathbf{y}^* = \arg\max_{z,\mathbf{y}} p(Z = z, \mathbf{Y} = \mathbf{y}|\mathbf{x}; \Theta)$$

This can be viewed as a MAP inference problem. In general, inference in a graphical model is challenging. Fortunately, the joint factors in our model are linear, and the event and relation factors are log-linear; we can cast MAP inference as an integer linear programming (ILP) problem, and then compute an approximation in polynomial time by means of linear programming using

randomized rounding, as proposed in [130].

We build one ILP problem for every NewsSpike. The variables of the ILP are Z and $\mathbf{Y}$, which only take values of 0 or 1. The objective function is the sum of logs of the event and relation factors Φ^Z and Φ^Y. The temporal burstiness heuristic of Φ^{joint} is encoded as a linear inequality constraint $z \geq y_i$; the one-mention per discourse heuristic of Φ^{joint} is encoded as the constraint $\sum_{y_i \in \mathbf{y}_d} y_i \leq 1$.

Learning

Our training data consists of $N = 500$ labeled NewsSpike-set in the form of $\{(R_i, R_i^{\text{gold}}) \mid_{i=1}^{N}\}$. Each R is the set of all relations in the NewsSpike-set while R^{gold} is a manually created subset of R containing relations describing the NewsSpike. R^{gold} could be empty if the NewsSpike is not good for paraphrasing. For our model, the gold assignment $y^{r\,\text{gold}} = 1$ if $r \in R^{\text{gold}}$; the gold assignment $z^{\text{gold}} = 1$ if R^{gold} is not empty.

Given $\{(R_i, R_i^{\text{gold}}) \mid_{i=1}^{N}\}$, learning over similar models is commonly done via maximum likelihood estimation as follows:

$$L(\Theta) = \log \prod_i p(Z_i = z_i^{\text{gold}}, \mathbf{Y}_i = \mathbf{y}_i^{\text{gold}} \mid \mathbf{x}_i, \Theta)$$

For features in relation factors, the partial derivative for the ith model is:

$$\Phi_j(\mathbf{y}_i^{\text{gold}}, \mathbf{x}_i) - E_{p(z_i, \mathbf{y}_i|, \mathbf{x}_i, \Theta)} \Phi_j(\mathbf{y}_i, \mathbf{x}_i)$$

where $\Phi_j(\mathbf{y}_i, \mathbf{x}_i) = \sum \phi_j(X, Y, \mathbf{x})$, the sum of values for the jth feature in the ith model; and values of X, Y come from the assignment $\mathbf{y}_i$. For features in event factors, the partial derivative is derived similarly as

$$\phi_j(z_i^{\text{gold}}, \mathbf{x}_i) - E_{p(z_i, \mathbf{y}_i|, \mathbf{x}_i, \Theta)} \phi_j(z_i, \mathbf{x}_i)$$

It is unclear how to efficiently compute the expectations in the above formula, a brute force approach requires enumerating all assignments of $\mathbf{y}_i$, which is exponentially large with the number of relations. Instead, we opt to use a more tractable perceptron learning approach [30, 54]. In-

stead of computing the expectations, we simply compute $\phi_j(z_i^*, \mathbf{x}_i)$ and $\Phi_j(\mathbf{y}_i^*, \mathbf{x}_i)$, where $z_i^*, \mathbf{y}_i^*$ is the assignment with the highest probability, generated by the MAP inference algorithm using the current weight vector. The weight updates are the following:

$$\Phi_j(\mathbf{y}_i^{\mathsf{gold}}, \mathbf{x}_i) - \Phi_j(\mathbf{y}_i^*, \mathbf{x}_i) \tag{3.1}$$

$$\phi_j(z_i^{\mathsf{gold}}, \mathbf{x}_i) - \phi_j(z_i^*, \mathbf{x}_i) \tag{3.2}$$

The updates can be intuitively explained as penalties on errors. In sum, our learning algorithm consists of iterating the following two steps: (1) infer the most probable assignment given the current weights; (2) update the weights by comparing inferred assignments and the truth assignment.

3.5 Empirical Evaluation

We first introduce the experimental setup for our empirical study, and then we attempt to answer two questions in sections 3.5.2 and 3.5.3 respectively: First, does the NEWSSPIKE-PARA algorithm effectively exploit the proposed heuristics and outperform other approaches which also use news streams? Secondly, do the proposed temporal heuristics paraphrase relations with greater precision than the distributional hypothesis?

3.5.1 Experimental Setup

Since we were unable to find any suitable time-stamped, parallel, news corpus, we collected data using the following procedure:

- Collect RSS news seeds, which contain the title, time-stamp, and abstract of the news items.
- Use these titles to query the Bing news search engine API and collect additional time-stamped news articles.
- Strip HTML tags from the news articles using Boilerpipe [60]; keep only the title and first paragraph of each article.
- Extract shallow relation tuples using the OpenIE system [41].

We performed these steps every day from January 1 to February 22, 2013. In total, we collected 546,713 news articles, for which 2.6 million extractions had 529 thousand unique relations. These led to 79,427 NewsSpike-sets.

We used several types of features for paraphrasing: *1)* spike features obtained from time series; *2)* tense features, such as whether two relation phrases are both in the present tense; *3)* cause-effect features, such as whether two relation phrases often appear successively in the news articles; *4)* text features, such as whether sentences are similar; *5)* syntactic features, such as whether a relation phrase appears in a clausal complement; and *6)* semantic features, such as whether a relation phrase contains negative words.

Text and semantic features are encoded using the relation factors of section 3.4.3. For example, in Figure 3.2, the factor Φ_2^Y includes the textual similarity between the sentences containing the phrases *"step down"* and *"be chairman of"* respectively; it also includes the feature that the tense of *"step down"* (present) which is different from the tense of *"be chairman of"* (past).

3.5.2 Comparison with Methods using Parallel News Corpora

We evaluated NEWSSPIKE-PARA against other methods that also use time-stamped news. These include the models mentioned in section 3.3 and state-of-the-art paraphrasing techniques.

Human annotators created gold paraphrase clusters for 500 NewsSpike-sets; note that some NewsSpike-sets yield no gold cluster, since at least two synonymous phrases. Two annotators were shown a set of candidate relation phrases in context and asked to select a subset of these that described a shared event (if one existed). There was 98% phrase-level agreement. Precision and recall were computed by comparing an algorithm's output clusters to the gold cluster of each NewsSpike. We consider paraphrases with minor lexical diversity, *e.g.* $(\texttt{goto}, \texttt{gointo})$, to be of lesser interest. Since counting these trivial paraphrases tends to exaggerate the performance of a system, we also report precision and recall on *diverse clusters i.e.*, those whose relation phrases all have different head verbs. Figure 3.3 illustrates these metrics with an example; note under our diverse metrics, all phrases matching $\texttt{go} *$ count as one when computing both precision and recall. We conduct 5-fold cross validation on our labeled dataset to get precision and recall numbers when

output	{**go into**, **go to**, `speak`, `return`, **head to**}
gold	{**go into**, **go to**, `approach`, **head to**}
gold$_{\text{div}}$	{**go** $*$, `approach`, **head to**}
P/R	precision $= 3/5$ recall $= 3/4$
P/R$_{\text{div}}$	precision$_{\text{div}} = 2/4$ recall$_{\text{div}} = 2/3$

Figure 3.3: an example pair of the output cluster and the gold cluster, and the corresponding precision recall numbers.

System	P/R		P/R$_{\text{div}}$	
	prec	rec	prec	rec
Baseline	0.67	1.00	0.53	1.00
Pairwise	0.90	0.60	0.81	0.37
Socher	0.81	0.35	0.68	0.29
NEWSSPIKE-PARA	**0.92**	0.55	**0.87**	0.31

Table 3.1: Comparison with methods using parallel news corpora

the system requires training.

We compare NEWSSPIKE-PARA with the models in Section 3.4, and also with the state-of-the-art paraphrase extraction method:

Baseline: the model discussed in Section 3.4.1. This system does not need any training, and generates outputs with perfect recall.

Pairwise: the pairwise model discussed in Section 3.4.2 and using the same set of features as used by NEWSSPIKE-PARA. To generate output clusters, transitivity is assumed inside the NewsSpike-set. For example, when the pairwise model predicts that (r_1, r_2) and (r_1, r_3) are both paraphrases, the resulting cluster is $\{r_1, r_2, r_3\}$.

Socher: Socher *et al.* [112] achieved the best results on the Dolan *et al.* [37] dataset, and released their code and models. We used their off-the-shelf predictor to replace the classifier in our Pairwise model. Given sentential paraphrases, aligning relation phrases is natural, because OpenIE has already identified the relation phrases.

Table 3.1 shows precision and recall numbers. It is interesting that the basic model already obtains 0.67 precision overall and 0.53 in the diverse condition. This demonstrates that the NewsSpike-sets generated from the news streams are a promising resource for paraphrasing. Socher's method performs better, but not as well as Pairwise or NEWSSPIKE-PARA, especially in the diverse cases. This is probably due to the fact that Socher's method is purely based on text metrics and does not consider any temporal attributes. Taking into account the features used by NEWSSPIKE-PARA, Pairwise significantly improves the precision, which demonstrates the power of our temporal correspondence heuristics. Our joint cluster model, NEWSSPIKE-PARA, which considers both temporal features and constraints, gets the best performance in both conditions.

We conducted ablation testing to evaluate how spike features and tense features, which are particularly relevant to the temporal aspects of news streams, can improve performance. Figure 3.4 compares the precision/recall curves for three systems in the diverse condition: (1) NEWSSPIKE-PARA; (2) w/oSpike: turning off all spike features; and (3) w/oTense: turning off all features about tense. (4) w/oDiscourse: turning off one event-mention per discourse heuristic. There are some dips in the curves because they are drawn after sorting the predictions by the value of the corresponding ILP objective functions, which do not perfectly reflect prediction accuracy. However, it is clear that NEWSSPIKE-PARA produces greater precision over all ranges of recall.

3.5.3 Comparison with Methods using the Distributional Hypothesis

We evaluated our model against methods based on the distributional hypothesis. We ran NEWSSPIKE-PARA over all NewsSpike-sets except for the development set and compared to the following systems:

Resolver: Resolver [134] uses a set of extraction tuples in the form of (a_1, r, a_2) as the input and creates a set of relation clusters as the output paraphrases[4]. We evaluated Resolver's performance with an input of the 2.6 million extractions described in section 3.5.1, using Resolver's default parameters.

[4]Resolver also produces argument clusters, but this paper only evaluates relation clustering

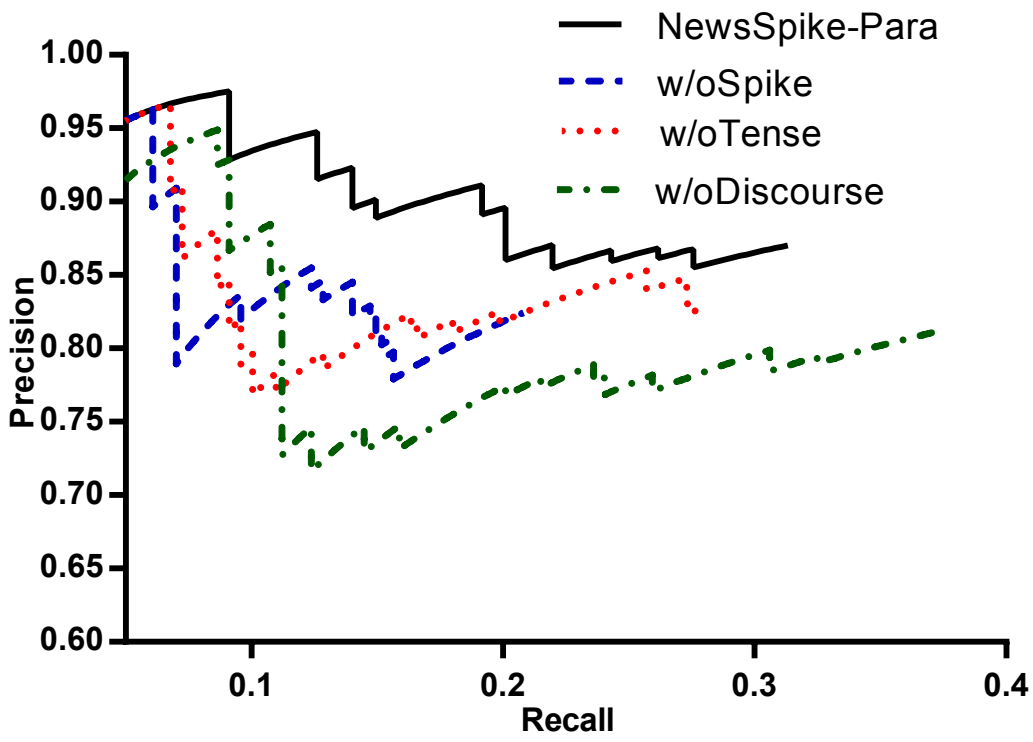

Figure 3.4: Precision recall curves on hard, diverse cases for NEWSSPIKE-PARA, w/oSpike, w/oTense and w/oDiscourse.

ResolverNYT: Since Resolver is supposed to perform better when given more accurate statistics from a larger corpus, we tried giving it more data. Specifically, we ran ReVerb on 1.8 million NY Times articles published between 1987 and 2007 obtain 60 million extractions [103]. We ran Resolver on the union of this and our standard test set, but report performance only on clusters whose relations were seen in our news stream.

ResolverNytTop: Resolver is designed to achieve good performance on its top results. We thus ranked the ResolverNYT outputs by their scores and report the precision of the top 100 clusters.

Cosine: Cosine similarity is a basic metric for the distributional hypothesis. This system employs the same setup as Resolver in order to generate paraphrase clusters, except that Resolver's similarity metric is replaced with the cosine. Each relation is represented by a vector of argument pairs. The similarity threshold to merge two clusters was 0.5.

CosineNYT: As for ResolverNYT, we ran CosineNYT with an extra 60 million extractions and reported the performance on relations seen in our news stream.

We measured the precision of each system by manually labeling all output if 100 or fewer clusters were generated (*e.g.*ResolverNytTop), otherwise 100 randomly chosen clusters were sampled. Annotators first determined the meaning of every output cluster and then created a gold cluster by choosing the correct relations[5]. Unlike many papers that simply report recall on the most frequent relations, we evaluated the total number of returned relations in the output clusters. As in Section 3.5.2, we also report numbers for the case of lexically diverse relation phrases.

As can be seen in Table 3.2, NEWSSPIKE-PARA outperformed methods based on the distributional hypothesis. The performance of the Cosine and CosineNyt was very low, suggesting that simple similarity metrics are insufficient for handling the paraphrasing problem, even when large-scale input is involved. Resolver and ResolverNyt employ an advanced similarity measurement and achieve better results. However, it is surprising that Resolver results in a greater precision than ResolverNyt. It is possible that argument pairs from news streams spanning 20 years some-

[5]The gold cluster could be empty if the output cluster was nonsensical

System	all		diverse	
	prec	#rels	prec	#rels
Resolver	0.78	129	0.65	57
ResolverNyt	0.64	1461	0.52	841
ResolverNytTop	0.83	207	0.72	79
Cosine	0.65	17	0.33	9
CosineNyt	0.56	73	0.46	59
NEWSSPIKE-PARA	**0.93**	21580	**0.87**	5681

Table 3.2: Comparison with methods using the distributional hypothesis

times provide incorrect evidence for paraphrasing. For example, there were extractions like (the Rangers, be third in, the NHL) and (the Rangers, be fourth in, the NHL) from news in 2007 and 2003 respectively. Using these phrases, ResolverNyt produced the incorrect cluster {be third in, be fourth in}. NEWSSPIKE-PARA achieves greater precision than even the best results from ResolverNytTop, because NEWSSPIKE-PARA successfully captures the temporal heuristics, and does not confuse synonyms with antonyms, or causes with effects. NEWSSPIKE-PARA also returned on order of magnitude greater number of relations than other methods.

3.5.4 Discussion

Unlike some domain-specific clustering methods, we tested on all relation phrases extracted by OpenIE on the collected news streams. There are no restrictions on the types of relations. Output paraphrases cover a broad range, including politics, sports, entertainment, health, science, etc. There are 8,885 nonempty clusters over 15,740 distinct phrases with average size 2.3. Unlike methods based on distributional similarity, NewsSpike correctly clusters infrequently appearing phrases.

Since we focus on high precision, it is not surprising that most clusters are of size 2 and 3. These high precision clusters can contribute a lot to generate larger paraphrase clusters. For example, one can invent the technique to merge smaller clusters together. The work presented here provides a foundation for future work to more closely examine these challenges.

While the work presented here gives promising results, there are still behaviors found in news streams that prove challenging. Many errors are due to the discourse context: the two sentences are synonymous in the given NewsSpike-set, but the relation phrases are not paraphrases in general. For example, consider the following two sentences: "DA14 narrowly misses Earth" and "DA14 flies so close to Earth". Statistics information from large corpus would be helpful to handle such challenges. Note in this paper, in order to fairly compare with the distributional hypothesis, we purposely forced NEWSSPIKE-PARA not to rely on any distribution similarity. But NEWSSPIKE-PARA's graphical model has the flexibility to incorporate any similarity metrics as features. Such a hybrid model has great potential to increase both precision and recall, which is one goal for future work.

3.6 Conclusion

Paraphrasing event relations is crucial to many natural language processing applications, including relation extraction, question answering, summarization, and machine translation. Unfortunately, previous approaches based on distribution similarity and parallel corpora, often produce low precision clusters. This paper introduces three Temporal Correspondence Heuristics that characterize semantically equivalent phrases in news streams. We present a novel algorithm, NEWSSPIKE-PARA, based on a probabilistic graphical model encoding these heuristics, which harvests high-quality paraphrases of event relations.

Experiments show NEWSSPIKE-PARA's improvement relative to several other methods, especially at producing lexically diverse clusters. Ablation tests confirm that our temporal features are crucial to NEWSSPIKE-PARA's precision. In order to spur future research, we are releasing an annotated corpus of time-stamped news articles and our harvested relation clusters.

Chapter 4

NEWSSPIKE-RE: EXPLOITING PARALLEL NEWS STREAMS FOR UNSUPERVISED EVENT EXTRACTION

Most approaches to *relation extraction*, the task of extracting ground facts from natural language text, are based on machine learning and thus starved by scarce training data. Manual annotation is too expensive to scale to a comprehensive set of relations. Distant supervision, which automatically creates training data, only works with relations that already populate a knowledge base (KB). Unfortunately, KBs such as FreeBase rarely cover event relations (*e.g. "person travels to location"*). Thus, the problem of extracting a wide range of events — e.g., from news streams — is an important, open challenge.

In this chapter, we introduce NEWSSPIKE-RE, a novel, unsupervised algorithm that discovers event relations and then learns to extract them. NEWSSPIKE-RE uses a novel probabilistic graphical model to cluster sentences describing similar events from parallel news streams. These clusters then comprise training data for the extractor. Our evaluation shows that NEWSSPIKE-RE generates high quality training sentences and learns extractors that perform much better than rival approaches, more than doubling the area under a precision-recall curve compared to Universal Schemas.

4.1 Introduction

Relation extraction, the process of extracting structured information from natural language text, grows increasingly important for Web search and question answering. Traditional supervised approaches, which can achieve high precision and recall, are limited by the cost of labeling training data and are unlikely to scale to the thousands of relations on the Web. Another approach, distant supervision [31, 126], creates its own training data by matching the ground instances of a

Knowledge base (KB) (*e.g.* Freebase) to the unlabeled text.

Unfortunately, while distant supervision can work well in some situations, the method is limited to relatively *static* facts (*e.g.*, *born-in(person, location)* or *capital-of(location,location)*) where there is a corresponding knowledge base. But what about dynamic *event relations* (also known as *fluents*), such as travel-to (person, location) or fire (organization, person)? Since these time-dependent facts are ephemeral, they are rarely stored in a pre-existing KB. At the same time, knowledge of real-time events is crucial for making informed decisions in fields like finance and politics. Indeed, news stories report events almost exclusively, so learning to extract events is an important open problem.

Researchers have also proposed *unsupervised* relation extraction methods, such as matrix factorization [98] and latent variable models [133]. Often these approaches cluster phrases and instances jointly, based on co-occurrence. While these methods can be used to extract event at least in theory, inspection shows that they often confuse related but semantically different phrases (*e.g.* buy and own; beat and lose), which are not synonymous and may even be antonyms. This is because the common clustering assumptions (*e.g.* low rank, latent topic) about the co-occurring observations don't produce any direct negative evidence to separate the heavily co-occurring phrases. As a result, they will be clustered together even if they are semantically different. This limitation significantly reduces the precision of the traditional unsupervised approaches.

This chapter develops a new unsupervised technique, NEWSSPIKE-RE, to both discover event relations and extract them with high precision. The intuition underlying NEWSSPIKE-RE is that the text of articles from two different news sources are not independent, since they are each conditioned on the same real-world events. By looking for rarely described entities that suddenly "spike" in popularity on a given date, one can identify paraphrases. Such *temporal correspondence* [139] allow one to cluster diverse sentences, and the resulting clusters may be used to form training data in order to learn event extractors. Furthermore, one can also exploit parallel news to obtain direct *negative* evidence. To see this, suppose one day the news includes the following:

"Snowden travels to Hong Kong, off southeastern China."
"Snowden cannot stay in Hong Kong as Chinese officials will not allow ..."

Since news stories are usually coherent, it is highly unlikely that travel to and stay in (which is negated) are synonymous. By leveraging such direct negative phrases, we can learn extractors capable of distinguishing heavily co-occurring but semantically different phrases, thereby avoiding many extraction errors. Our NEWSSPIKE-RE system encapuslates these intuitions in a novel graphical model making the following contributions:

- We develop a method to discover a set of distinct, salient event relations from news streams.
- We describe an algorithm to exploit parallel news streams to cluster sentences that belong to the same event relations. In particular, we propose the *temporal negation heuristic* to avoid conflating co-occurring but non-synonymous phrases.
- We introduce a probabilistic graphical model to generate training for a sentential event extractor without requiring any human annotations.
- We present a series of detailed experiments demonstrating that the event extractors learned from the generated training data significantly outperform several competitive baselines, *e.g.* our system more than doubles the area under the micro-averaged, PR curve (0.80 vs. 0.30) compared to Riedel's Universal Schemas [98].

4.2 System Overview

News articles report an enormous number of events every day. Our system, NEWSSPIKE-RE, aligns parallel news streams to identify and extract these events as shown in Figure 4.1. NEWSSPIKE-RE has both training and test phases. Its training phase has two main steps: event-relation discovery and training-set generation. Section 4.3 describes our event relation discovery algorithm, which processes time-stamped news articles to discern a set of salient, distinct event relations in the form of $E = e(t_1, t_2)$, where e is a representative *event phrase* and t_i are types of the two arguments. NEWSSPIKE-RE generates the event phrases using an Open Information Extraction (IE) system [41], and uses a fine-grained entity recognition system FIGER [69] to generate type descriptors such as "company ", "politician", and "medical treatment".

The second part of NEWSSPIKE-RE's training phase, described in Section 4.4, is a method

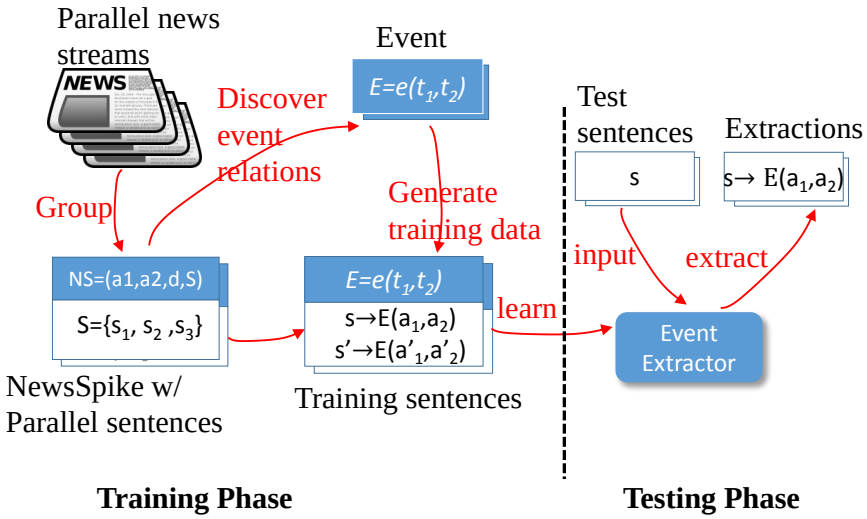

Figure 4.1: During its training phase, NEWSSPIKE-RE first groups parallel sentences as *NewsSpikes*. Next, the system automatically discovers a set of event relations. Then, a probabilistic graphical model clusters sentences from the NewsSpike as training data for each discovered relation, which is used to learn sentential event extractors. During the testing phase, the extractor takes test sentences as input and predicts event extractions.

for building extractors for the discovered event relations. Our approach is motivated by the intuition, adapted from Zhang and Weld [139], that articles from different news sources typically use different sentences to describe the same event, and that corresponding sentences can be identified when they mention a unique pair of real-world entities. For example, when an unusual entity pair (Selena, Norway) is suddenly seen in three articles on a single day:

Selena traveled to Norway to see her ex-boyfriend.

Selena arrived in Norway for a rendezvous with Justin.

Selena's trip to Norway was no coincidence.

It is likely that all three refer to the same event relation, *travel-to(person, location)*[1], and can be used as positive training examples for the relation.

Additionally, the parallel sentences allow us to cluster same-event instances more accurately.

[1]For clarity in the paper, we refer to this relation as *travel-to*, even though the phrase *arrive in* is actually more frequent and is selected as the name of this relation by our event discovery algorithm, as shown in Table 4.2.

For example, one day when we read

Murphy helps Mets defeat Yankees.

The Mets' victory against Yankees means...

The lineup as the Mets face Yankees ...

Even if the phrase *beat* is not there, it is still possible to conclude that *Mets beat Yankee*, because the shared event phrase "<-[poss]-victory-[prep-against]->". It further allows us to use the additional good sentences *"... defeat ..."*.

As in Zhang & Weld [139], we group *parallel sentences* sharing the same argument pair and date in a structure called a *NewsSpike*. But rather than only considering sentences that have OpenIE tuples, we include all sentences mentioning the arguments (*e.g. Selena's trip to Norway*) in the NewsSpike, and use the lexicalized dependency path between the arguments (*e.g.* <-[poss]-trip-[prep-to]->[2], as the event phrase. In this way, we can generalize extractors beyond the scope of OpenIE. Formally, a NewsSpike is a tuple, (a_1, a_2, d, S), where a_1 and a_2 are arguments (*e.g.* `Selena`), d is a date, and S is a set of argument-labeled sentences $\{(s, a_1, a_2, p) \ldots\}$ in which s is a sentence with arguments a_1 and a_2 and event phrase p.

It's important that non-synonomous sentences like *"Selena stays in Norway"* should be excluded from the training data for *travel-to(person, location)* even if a *travel-to* event did apply to that argument pair. In order to select only the synonomous sentences, we develop a probabilistic graphical model, described in Section 4.4.2, to accurately assign sentences from NewsSpikes to each discovered event relation E. Given this annotated data, NEWSSPIKE-RE trains extractors using a multi-class logistic regression classfier.

During the testing phase, NEWSSPIKE-RE accepts arbitrary sentences (no date-stamp required), uses FIGER to identify possible arguments, and uses the classifier to predicts which events (if any) hold between an argument pair. We describe the extraction process in Section 4.5.

Note that NEWSSPIKE-RE is an unsupervised algorithm that requires no manual labelling of the training instances. Like distant supervision, the key is to automatically generate the training

[2]The dependency path will be referred to as "'s trip to".

data, at which point a traditional supervised classifier may be applied to learn an extractor. Because distant supervision creates very noisy annotations, researchers often use specialized learners that model the correctness of a training example with a latent variable [99, 54], but we found this unnecessary, because NEWSSPIKE-RE creates high quality training data.

4.3 Discovering Salient Events

The first step of NEWSSPIKE-RE is to discover a set of event relations in the form of $E = e(t_1, t_2)$, where e is an event phrase, and t_i are fine-grained argument types generated by FIGER, augmented with the important types "number" and "money", which are recognized by the Stanford name entity recognition system [43]. To be most useful, the discovered event relations should cover salient events that are frequently reported in the news articles. Formally, we say that a NewsSpike $\eta = (a_1, a_2, d, S)$ mentions $E = e(t_1, t_2)$ if the types of a_i are t_i for each i, and one of its sentence has e as the event phrase between the arguments. To maximize the salience of the events, NEWSSPIKE-RE will prefer event relations that are *"mentioned"* by more NewsSpikes.

In addition, the set of event relations should be distict. For example, if the relation *travel-to(person, location)* is already in the set, then *visit(person, location)* should not be selected as a separate relation. To reduce overlap, discovered event relations should not be mentioned by the same NewsSpike.

Let $\mathcal{E}$ be all candidate event relations, $\mathcal{N}$ be all NewsSpikes. Our goal is to select the K most salient relations from $\mathcal{E}$, minimizing overlap between relations. We can frame this task as a variant of the bipartite graph edge-cover problem. Let a bipartite graph G have one node E_i for each event relation in $\mathcal{E}$ and one node η_j for each NewsSpike in $\mathcal{N}$. There is an edge between E_i and η_j if η_j mentions E_i. The edge-cover problem is to select a largest subset of edges subject to (1) at most K nodes of E_i are chosen and all edges incident to them are chosen as the covered edges; (2) each node of η_j is incident to at most one edge. The first constraint guarantees that there are exactly K event relations discovered; the second constraint ensures that no NewsSpike participates in two event relations. Figure 4.2 shows the optimized solution of a simple graph with $K = 2$, which can cover 3 edges with 2 event relations that have no overlapping NewsSpikes.

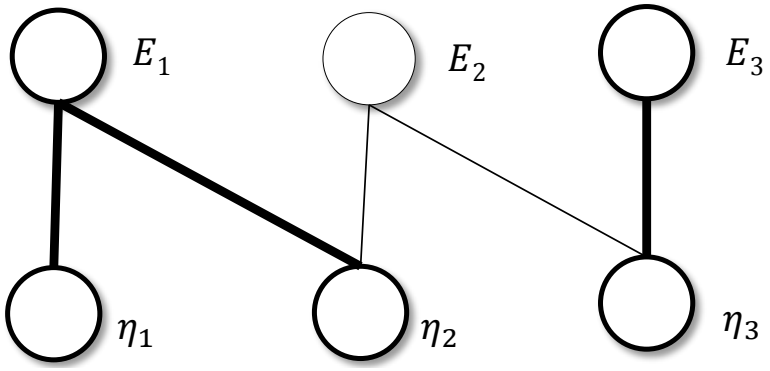

Figure 4.2: A simple example of the edge-cover algorithm with K=2, where E_i are event relations and η_j are NewsSpikes. The optimal solution selects E_1 with edges to η_1 and η_2, and E_3 with edge to η_3. These two event relations cover all the NewsSpikes.

Since both the objective function and constraints are linear, we can optimize this edge-cover problem with integer linear programming [84]. By solving the optimization problem, NEWSSPIKE-RE finds a salient set of event relations incident to the covered edges. The discovered relations with K set to 30 are shown in Table 4.2 in Section 4.6. In addition, the covered edges bring us the initial mapping between the event types and NewsSpikes, which is used to train the probabilistic model in Section 4.4.3.

Let $x_\eta^E \in \{0, 1\}$ represent the edge between η and E, while $x_\eta^E = 1$ if the edge is covered; let the value of $y^E \in \{0, 1\}$ tell us whether E is selected as the salient event. The edge-cover problem becomes the Integer Linear Programming problem:

$$\begin{aligned} \text{Max} \quad & \sum_{E,\eta} x_\eta^E \\ \text{s.t.} \quad & \forall R : y^E \leq \sum_{\eta} x_\eta^E; \sum_{E} y^E = K \\ & \forall \eta : \sum_{E} x_\eta^E \leq 1 \end{aligned} \tag{4.1}$$

Those K fluent relations with $y^E = 1$ are the our discovered events.

4.4 Generating the Training Sentences

After NEWSSPIKE-RE has discovered a set of event relations, it then generates training instances to learn an extractor for each relation. In this section, we present our algorithm for generating the training sentences. As shown in Figure 3.1, the generator takes N NewsSpikes $\{\eta_i = (a_{1i}, a_{2i}, d_i, S_i)|i = 1 \dots N\}$ and K event relations $\{E_k = e_k(t_{1k}, t_{2k})|k = 1 \dots K\}$ as input. For every event relation, E_k, the generator identifies a subset of sentences from $\cup_{i=1}^{N} S_i$ expressing the event relation as training sentences. In this section, we first characterize the paraphrased event phrases and the parallel sentences in NewsSpikes. Then we show how to encode this heuristic in a probabilistic graphical model that jointly paraphrases the event phrases and identifies a set of training sentences.

4.4.1 Exploiting Properties of Parallel News

Previous work [139] proposed several heuristics that are useful to find similar sentences in a NewsSpike. For example, the temporal functionality heuristic says that sentences in a NewsSpike with the same tense tend to be paraphrases. Unfortunately, these methods are too weak to generate enough data for training high quality event extractors: (1) they are "in-spike heuristics" that tend to generate small clusters from individual NewsSpikes. It remains unclear how to merge similar events occuring on different days and between different entities to increase cluster size. (2) they included heuristics to "gain precision at the expense of recall" (*e.g.* news articles do not state the same fact twice), because it is hard to obtain direct negative phrases inside one NewsSpike. In this paper, we exploit news streams in a cross-spike, global manner to obtain accurate positive and negative signals. This allows us to dramatically improve recall while maintaining high precision.

Our system starts from the basic observation that the parallel sentences tend to be coherent. So if a NewsSpike $\eta = (a_1, a_2, d, S)$ is an instance of an event relation $E = e(t_1, t_2)$, the event phrases in its parallel sentences tend to be paraphrases. But sometimes the sentences in the NewsSpike are related but not paraphrases. For example, one day "Snowden will stay in Hong Kong ..." appears together with "Snowden travels to Hong Kong ...". Although the fact *stay-in(Snowden, Hong*

Kong) is true, it is harmful to include "Snowden will stay in Hong Kong" in the training for *travel-to(person, location)*.

Detecting paraphrases remains a challenge to most unsupervised approaches because they tend to cluster heavily co-occurring phrases which may turn out to be semantically different or even antonymous. [139] presented a method to avoid confusion between antonym and synonyms in NewsSpikes, but did not address the problem of related but different phrases like *travel to* and *stay in* in a NewsSpike.

To handle this, our method rests on a simple observation: when you read "Snowden travels to Hong Kong" and "Snowden cannot stay in Hong Kong as Chinese officials do not allow ..." in the same NewsSpike, it is unlike that *travel to* and *stay in* are synonymous event phrases because otherwise the two news stories are describing the opposite event. This observation leads to:

Temporal Negation Heuristic 1 *Two event phrases p and q tend to be semantically different if they co-occur in the NewsSpike but one of them is in negated form.*

The temporal negation heuristic helps in two ways: (1) it provides some direct negative phrases for the event relations; NEWSSPIKE-RE uses these to heuristically label some variables in the model. (2) It creates some useful features to implement a form of transitvity. For example, if we find that *live in* and *stay in* are frequently co-occurring and the temporal negation heuristic tells us that *travel to* and *stay in* are not paraphrases, this is evidence that *live in* is unlikely to be a paraphrase of *travel to*, even if they are heavily co-occurring.

The following section describes our implementation that uses these properties to generate high quality training. Our goal is the following: a sentence (s, a_1, a_2, p) from NewsSpike $\eta = (a_1, a_2, d, S)$ should be included in the training data for event relation $E = e(t_1, t_2)$ if the event phrase p is a paraphrase of e and the event relation E happens to the argument pair (a_1, a_2) at time d.

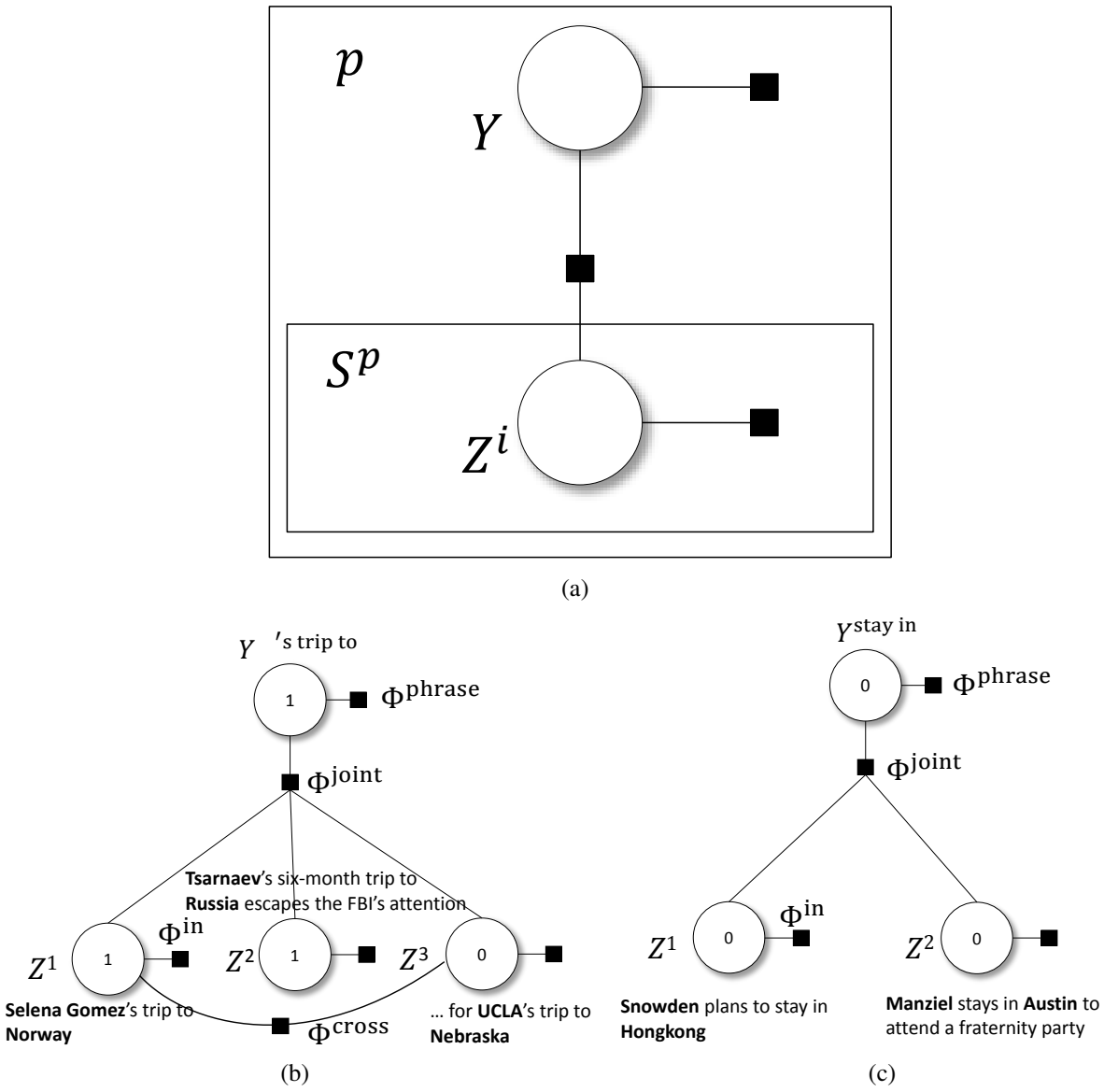

Figure 4.3: (a) The connected components depicted as plate model, where each Y is a Boolean variable for a relation phrase and each Z is a Boolean variable for a training sentence for with that phrase; (b) and (c) are example connected components for the event phrases *'s trip to* and *stay in* respectively. The goal of the model is to set $Y = 1$ for good paraphrases of a relation and to set $Z = 1$ for good training sentences.

4.4.2 Joint Cluster Model

As discussed above, to identify a high quality set of training sentences from NewsSpikes, one needs to combine evidence that event phrases are paraphrases with evidence from NewsSpikes. For this purpose, we define an undirected graphical model to jointly reason about paraphrasing the event phrases and identifying the training sentences from NewsSpikes. We first list the notation used in this section:

E	event relation
$p \in P$	event phrases
$s \in S^p$	sentences w/ the event phrase p
Y^p	Is p a paraphrase for E?
Z_p^s	Is s w/ p good training for E?
Φ	factors

Let P be the union of all the event phrases from every NewsSpike. For each $p \in P$, let S^p be the set of sentences having p as its event phrase.

Figure 4.3(a) shows the model in plate form. There are two kinds of random variables corresponding to phrases and sentences, respectively. For each event relation $E = e(t_1, t_2)$, there exists a connected component for every event phrase $p \in P$ that models (1) whether p is a paraphrase of e or not (modeled using Boolean phrase variables, Y^p); and (2) whether each sentence of S^p is a good training sentence for E (modeled using $|S^p|$ Boolean sentence variables $\{Z_p^s | s \in S^p\}$. Intuitively, the goal of the model is to find the set of good training sentences, with $Z_p^s = 1$. The union of such sentences over the different phrases, $\cup_p \{s | Z_p^s = 1\}$, defines the training sentences for the event. Figure 4.3(b) and 4.3(c) show two example connected-components for the event phrases *'s trip to* and *stay in* respectively.

Now, we can define the joint distribution over the event phrases and the sentences. The joint distribution is a function defined on factors that encode our observations about NewsSpikes as features and constraints. The *phrase factor* Φ^{phrase} is a log-linear function attaching to Y^p with the paraphrasing features, such as whether p and e co-occur in the NewsSpikes, or whether p shares the same head word with e. They are used to distinguish whether p is a good event phrase.

A sentence should not be identified as a good training sentence if it does not contain a positive event phrase. For example, if $Y^{\text{stay in}}$ in Figure 4.3(b) takes the value of 0, thus all sentences with the event phrase *stay in* should also take the value of 0. We implement this constraint with a *joint factor* Φ^{joint} among Y^p and Z_p^s variable.

In addition, good training sentences occur when the NewsSpike is an event instance. To encode this observation, we need to featurize the NewsSpikes and let them bias the assignments. Our model implements this with two types of log-linear factors: (1) the unary *in-spike factor* Φ^{in} depends on the sentence variables and contains features about the corresponding NewsSpike. The factor is used to distinguish whether the NewsSpike is an instance of $e(t_1, t_2)$, such as whether the argument types of the NewsSpike match the designated types t_1, t_2; (2) the pairwise *cross-spike factors* Φ^{cross} connect pairs of sentences. This uses features such as whether the pair of NewsSpikes for the two sentences have high textual similarity, and whether two NewsSpikes contain negated event phrases.

We define the joint distribution for the connected component for p as follows. Let $\mathbf{Z}$ be the vector of sentence variables, let $\mathbf{x}$ be the features. The joint distribution is:

$$p(Y = y, \mathbf{Z} = \mathbf{z}|\mathbf{x}; \Theta) \stackrel{\text{def}}{=} \frac{1}{Z_x} \Phi^{\text{phrase}}(y, \mathbf{x}) \times \Phi^{\text{joint}}(y, \mathbf{z}) \prod_s \Phi^{\text{in}}(z^s, \mathbf{x}) \prod_{s,s'} \Phi^{\text{cross}}(z^s, z^{s'}, \mathbf{x})$$

where the parameter vector Θ is the weight vector of the features in Φ^{in} and Φ^{cross}, which are log-linear functions. The joint factors Φ^{joint} is zero when $Y^p = 0$ but some $Z_p^s = 1$. Otherwise, it is set to 1. We use integer linear programming to perform MAP inference on the model, finding the predictions $y, \mathbf{z}$ that maximize the probability.

4.4.3 Learning from Heuristic Labels

We now present the learning algorithm for our joint cluster model. The goal of the learning algorithm is to set Θ for the log-linear functions in the factors in a way that maximizes the likelihood

Input: NewsSpikes and the connected components of the model;

Heuristic Labels:

1. find positive and negative phrases and sentences P^+, P^-, S^+, S^-;
2. label the connected componenets accordingly and create $\{(Y_i^{\mathsf{label}}, \mathbf{Z}_i^{\mathsf{label}}) \mid_{i=1}^{M}\}$.

Learning: Update Θ with the perceptron learning algorithm.

Output: the values of all variables in the connected components with the MAP inference.

Figure 4.4: Learning from Heuristic Labels

estimation. We do this in a totally unsupervised manner, since manual annotation is expensive and not scalable to large numbers of event relations.

The weights are learned in three steps: (1) NEWSSPIKE-RE creates a set of heuristic labels for a subset of variables in the graphical model; (2) it uses the heuristic labels as supervision for the model; (3) it updates Θ with the perceptron learning algorithm. The weights are used to infer the values of the variables that don't have heuristic labels. The procedure is summarized in Figure 4.4.

For each event relation $E = e(t_1, t_2)$, NEWSSPIKE-RE creates heuristic labels as follows: (1) P^+: the temporal functionality heuristic [139] says that if an event phrase p co-occurs with e in the NewsSpikes, it tends to be a paraphrase of e. We add the most frequently co-occurring event phrases to P^+. P^+ also includes e itself. (2) P^-: the temporal negation heuristic says that if p and e co-occur in the NewsSpike but one of them is in its negated form, p should be negatively labeled. We add those event phrases to P^-. If a phrase p appears in both P^+ and P^-, we remove it from both sets. (3) S^+: we first get the positive NewsSpikes from the solution of the edge-cover problem in section 4.3. We treat the NewsSpike η as positive if the edge between η and E is covered. Next, every sentence with $p \in P^+$ is added into S^+. (4) S^-: since the event relations discovered in section 4.3 tend to be distinct relations, a sentence is treated as negative sentence for E if it is heuristically labeled as positive for $E' \neq E$. In addition, S^- includes all sentences with $p \in P^-$.

With P^+, P^-, S^+, S^-, we define the heuristic labeled set to be $\{(Y_i^{\text{label}}, \mathbf{Z}_i^{\text{label}}) \mid_{i=1}^{M}\}$, where M is the number of the connected components with the corresponding event phrases $p \in P^+ \cup P^-$; $Y_i^{\text{label}} = 1$ if $p \in P^+$ and $Y_i^{\text{label}} = 0$ if $p \in P^-$. $\mathbf{Z}^i$ is labeled similarly, but note that if the sentence in the connected component doesn't exist in $S^+ \cup S^-$, NEWSSPIKE-RE doesn't include the corresponding variable in $\mathbf{Z}_i^{\text{label}}$. With $\{(Y_i^{\text{label}}, \mathbf{Z}_i^{\text{label}}) \mid_{i=1}^{M}\}$, learning can be done with maximum likelihood estimation as $L(\Theta) = \log \prod_i p(Y_i = y_i^{\text{label}}, \mathbf{Z}_i = \mathbf{z}_i^{\text{label}} \mid \mathbf{x}_i, \Theta)$. Following [30], we use a fast perceptron learning approach to update Θ. It consists of iterating two steps: (1) MAP inference given the current weight; (2) penalizing the weights if the inferred assignments are different from the heuristic labeled assignments.

4.5 Sentential Event Extraction

As shown in Figure 3.1, we learn the extractors from the generated training sentences. Note that most distant supervised [54, 119] approaches use multi-instance, aggregate-level training (*i.e.* the supervision comes from labeled sets of instances instead of individually labeled sentences). Coping with the noise inherent in these multi-instance bags remains a big challenge for distant supervision. In contrast, our sentence-level training data is more direct and minimizes noise. Therefore, we implement the event extractor as a simple multi-class, L2-regularized logistic regression classifier.

For features of the classifier, we use the lexicalized dependency paths, the OpenIE phrases, the minimal subtree of the dependency parse and the bag-of-words between the arguments. We also augment them with fine grained argument types produced by FIGER [69]. The event extractor that is learned can take individual test sentences (s, a_1, a_2) as input and predict whether that sentence expresses the event between (a_1, a_2).

4.6 Empirical Evaluation

Our evaluation addresses two questions. Section 4.6.2 considers whether our training generation algorithm identifies accurate and diverse sentences. Then, Section 4.6.3 investigates whether the event extractor, learned from the training sentences, outperforms other extraction approaches.

4.6.1 Experimental Setup

We follow the procedure described in [139] to collect parallel news streams and generate the NewsSpikes: first, we get news seeds and query the Bing newswire search engine to gather additional, time-stamped, news articles on a similar topic; next, we extract OpenIE tuples from the news articles and group the sentences that share the same arguments and date into NewsSpikes. We collected the news stream corpus from March 1st 2013 to July 1st 2014. We split the dataset into two parts: in the training phrase, we use the news streams in 2013 (named NS13) to generate the training sentences. NS13 has 33k NewsSpikes containing 173k sentences.

We evaluated the extraction performance on news articles collected in 2014 (named NS14). In this way, we make sure the test sentences are unseen during training. There are 15 million sentences in NS14. We randomly sample 100k unique sentences having two different arguments recognized by the name entity recognition system.

For our event discovery algorithm, we set the number of event relations to be 30 and ran the algorithm on NS13. The algorithm takes 6 seconds to run on a 2.3GHz CPU. Note that most previous unsupervised relation discovery algorithms require additional manual post-processing to assign names to the output clusters. In contrast, NEWSSPIKE-RE discovers the event relations fully automatically and the output is self-explanatory. We list them together with the by-event extraction performance in Table 4.2. From the table, we can see that most of the discovered event relations are salient with little overlap between relations.

While we arbitrarily set K to 30 in our experiments, there is no inherent limit to the number of relation phrases as long as the news corpus provides sufficient support to learn an extractor for each relation. In future, we plan to explore much larger sets of event relations to see if the extraction accuracy is maintained.

The joint cluster model that identifies training sentences for each event relation $E = e(t_1, t_2)$ uses cosine similarity between the event phrase p of a sentence and the canonical phrases of each relation as features in the phrase factors in Figure 4.3(a). It also includes the cosine similarity between p and a set of "anti-phrases" for the event relation which are recognized by the temporal

negation heuristic.

For the in-spike factor, we measure whether the fine-grained argument types of the sentence returned from the FIGER system matches the required t_i respectively. In addition, we implement the features from [139] to measure whether the sentence is describing the event of the NewsSpike. For the cross-spike factors, we use textual similarity features between the two sets of parallel sentences to measure the distance between the pair of NewsSpikes.

4.6.2 Quality of the Generated Training Set

The key to a good learning system is a high-quality training set. In this section, we compare our joint model against pipeline systems that consider paraphrases and argument type matching sequentially, based on the following paraphrasing techniques.

Basic is based on the temporal functionality heuristic of [139]. It treats all event phrases appearing in the same NewsSpike as paraphrases. **Yates09** uses Resolver [134] to create clusters of phrases. Resolver measures the similarity between the phrases by means of both distributional features and textual features. We convert the sentences in NewsSpikes into tuples in the form of (a_1, p, a_2), and run Resolver on these tuples to generate the paraphrases. **Zhang13**: We used the generated paraphrase set from [139]. **Ganit13**: Ganitkevitch *et al.* [46] released a large paraphrase database (PPDB) based on exploiting the bilingual parallel corpora. Note that some of these paraphrasing systems do not handle dependency paths. So when p is a dependency path, we use the surface string between the arguments as the phrase.

We also conduct ablation testing on NEWSSPIKE-RE to measure the effect of the cross-spike factors and the temporal negation heuristic: **w/o Cross** uses a simpler model by removing the cross-spike factors of NEWSSPIKE-RE; **w/o Negation** uses the same joint cluster model as NEWSSPIKE-RE but removes the features and the heuristic labels coming from the temporal negation heuristic.

We measured the micro- and macro- accuracy of each system by manually labeling 1000 randomly chosen output from each system[3]. Annotators read each training sentence, and decided if

[3]Two Odesk workers were asked to label the dataset, a graduate student then reconciled any disagreements.

system	all			diverse		
	#	mi.	ma.	#	mi.	ma.
Basic	43,718	0.50	0.62	12,701	0.38	0.51
Yates09	15,212	0.78	0.76	586	0.48	0.50
Ganit13	14,420	0.74	0.71	1,210	0.53	0.53
Zhang13	14,804	0.76	0.75	890	0.63	0.61
NEWSSPIKE-RE	20,105	**0.88**	**0.89**	2,156	**0.71**	**0.72**
w/o cross	16,463	0.86	0.86	1,883	0.67	0.69
w/o neg	33,548	0.76	0.81	4,019	0.64	0.68

Table 4.1: Quality of the generated training sentences (count, micro- and macro- accuracy), where "all" includes sentences with all event phrases and "diverse" are those with distinct event phrases.

it was a good example for a particular event. We also report the number of generated sentences. Since the extractor should generalize over sentences with dissimilar expressions, it is crucial to identify sentences with diverse event phrases. Therefore we also measured the accuracy and the count of a "diverse" condition: only consider the subset of sentences with distinct event phrases.

Table 4.1 shows the accuracy and the number of training examples. The basic temporal system brings us 0.50/0.62 micro- and macro- accuracy overall and 0.38/0.51 in the diverse condition. It shows that NewsSpikes are promising resources to generate the training set, but that elaboration is necessary. Yates09 gets 0.78/0.76 accuracy overall because its textual features help it to recognize many good sentences with similar phrases. But for the diverse condition, it gets lower precision because the distributional hypothesis fails to distinguish those correlated but different phrases.

Although Ganitkevitch13 and Zhang13 leverage existing paraphrase databases, it is interesting that their accuracy is still not good. It is largely because many times the paraphrasing must depend on the context: *e.g. "Cutler hits Martellus Bennett with TD in closing seconds."* is not good for the *beat(team, team)* relation, even though *hit* is a synonym for *beat* in general. These two systems show that it is not enough to use an off-the-shelf paraphrasing database for extraction.

The ablation test shows the effectiveness of the temporal negation hypothesis: after turning off the relevant features and heuristic labels, the precision drops about 10 percentage points. In addition, the cross-spike factors bring NEWSSPIKE-RE about 22% more training sentences and also increase the accuracy.

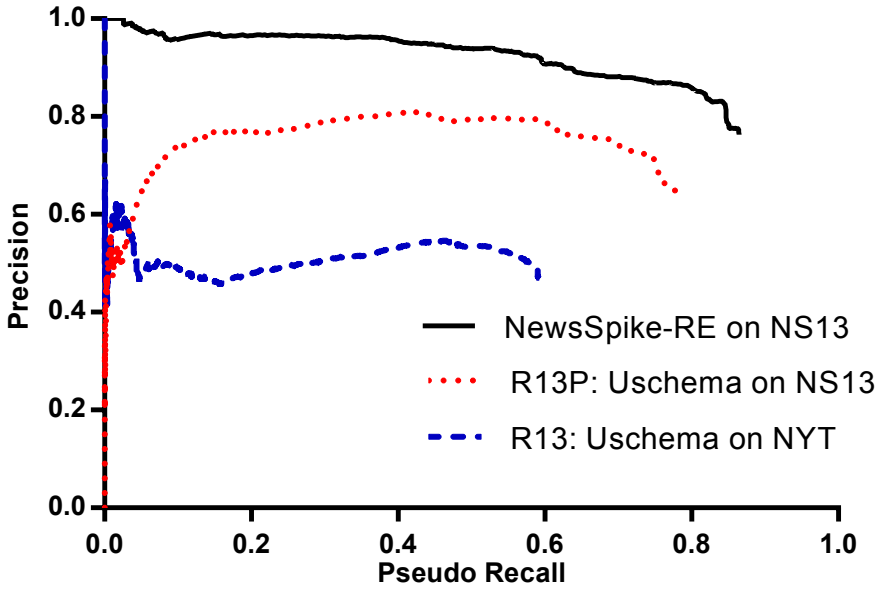

Figure 4.5: Precision pseudo-recall curves for all 30 event relations. NEWSSPIKE-RE has AUC 0.80, more than doubling R13 (0.30) and 35% higher than R13P (0.59) for all event relations.

We did bootstrap sampling to test the statistical significance of NEWSSPIKE-RE's improvement in accuracy over each comparison system and ablation of NEWSSPIKE-RE. For each system we computed the accuracy of 10 samples of 100 labeled outputs. We then ran the paired t-test over the accuracy numbers of each other system compared to NEWSSPIKE-RE. For all but w/o cross the improvement is strongly significant with p-value less than 1%. The increase in accuracy compared to w/o cross has borderline significance (p-value 5.5%), but is a clear win with its 22% increase in training size.

4.6.3 Performance of the Event Extractors

Most previous relation extraction approaches either require a manually labeled training set, or work only on a pre-defined set of relations that have ground instances from KBs. The closest work to NEWSSPIKE-RE is Universal Schemas [98], which addresses the limitation of distant supervision

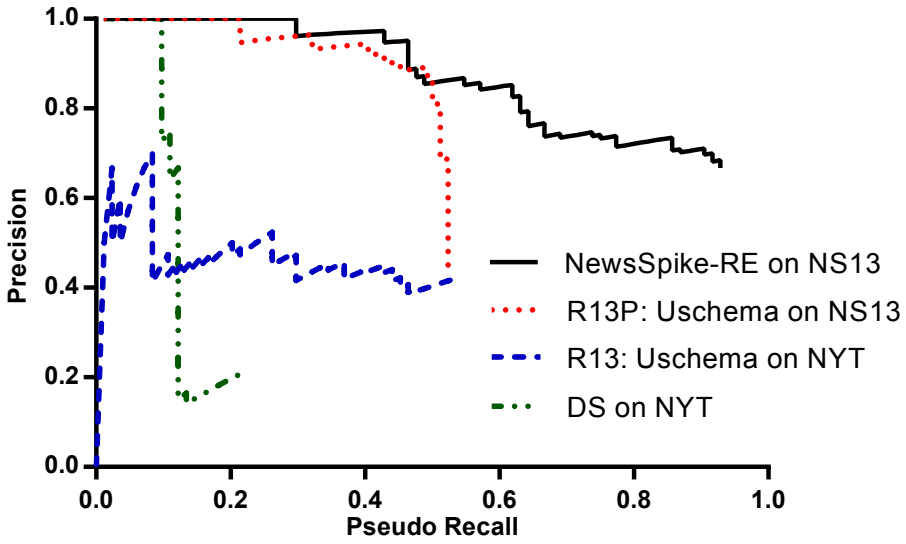

Figure 4.6: Precision pseudo-recall curves for *buy(org, org)*, this figure includes the distant supervision algorithm MIML learned from matching the Freebase relation to The New York Times. NEWSSPIKE-RE has AUC 0.80, more than doubling R13 (0.30) and 35% higher than R13P (0.59) for *buy(org, org)* event relations.

that the relations must exist in KBs. Their solution is to treat the surface strings, dependency paths, and relations from KBs as equal "schemas", and then to exploit the correlation between the instances and the schemas from a very large unlabeled corpus. In their paper, Riedel *et al.* evaluated only on static relations from Freebase and achieve state-of-the-art performance. But Universal Schemas can be adapted to handle events, by introducing the events as schemas and heuristically finding seed instances.

We set up a competing system (**R13**) as follows: (1) We take the NYTimes corpus published between 1987 and 2007 [103], the dataset used by Riedel *et al.* [98] containing 1.8 million NY Times articles; (2) The instances (*i.e.* the rows of the matrix) come from the entity pairs from the news articles; (3) There are two types of columns: some are the extraction features used by NEWSSPIKE-RE, including the lexicalized dependency paths described in Riedel *et al.*; others are event relations $E = e(t_1, t_2)$; (4) For an entity pair (a_1, a_2), if there is an OpenIE extraction (a_1, e, a_2) and the entity types of (a_1, a_2) match (t_1, t_2), we assume the event relation E is observed on that instance.

As shown in Table 4.1, parallel news streams are a promising resource for clustering because of the strong correlation between the instances and the event phrases. We train another version of Universal Schemas **R13P** on the parallel news streams NS13. In particular, entity pairs from different NewsSpikes are used as different rows in the matrix.

We would like to measure the precision and recall of the extractors. But note that it is impossible to fully label all the sentences, so we follow the "pooling" technique described in [98] to create the labeled dataset. For every competing system, we sample 100 top outputs for every event relation and add this to the pool. The annotators are shown these sentences and asked to judge whether the sentence expresses the event relation or not. After that, the labeled set become "gold" and can be used to measure the precision and pseudo-recall. There are in all 6,178 distinct sentences in the pool, since some outputs are produced by multiple systems. Among them, 2,903 sentences are labeled as positive. In Table 4.2, the # columns show the number of true extractions in the pool for every event relation.

Similar to the diverse condition in Table 4.1, it is important that the extractor can correctly pre-

dict on diverse sentences that are dissimilar to each other. Thus we conducted a "diverse pooling": for each system, we report numbers for the sentences with different dependency paths between the arguments for every discovered event.

Figure 4.5 shows the precision pseudo-recall curve for all sentences for the three systems. NEWSSPIKE-RE outperforms the competing systems by a large margin. For example, the area under the curve (AUC) of NEWSSPIKE-RE for all sentences is 0.80 while that of R13P and R13 are 0.59 and 0.30. This is a 35% increase over R13P and 2.7 times the area compared to R13. Similar increases in AUC are observed on diverse sentences. Table 4.2 further lists the breakdown numbers for each event relation, as well as the micro and macro average. Although Universal Schemas had some success for several relations, NEWSSPIKE-RE achieved the best F1 for 26 out of 30 event relations; best AUC for 26 out of 30. The advantage is even greater in the diverse condition. It is interesting to see that R13P performs much better than R13, since the data coming from NYTimes is much noisier.

A closer look shows that Universal Schemas tends to confuse correlated but different phrases. NEWSSPIKE-RE, however, rarely made these errors because our model can effectively exploit negative evidence to distinguish them.

Comparing to Distant Supervision

Although the most event relations in Table 4.2 cannot be handled by the distant supervised approach, it is possible to match *buy(org,org)* to Freebase relations with appropriate database operators such as join and select [137]. To evaluate how distant supervision performs, we introduce the system **DS on NYT** based on a manual mapping of *buy(org,org)* to the join relation[4] in Freebase. Then we match its instances to NYTimes articles and follow the steps of Surdeanu *et al.* [119] to train the extractor.

The matching to NYTimes brings us 264 positive instances having 5,333 sentences, but unfortunately the sentence-level accuracy is only 13% based on examination of 100 random sentences.

[4] `/organization/organization/companies_acquired⋈/business/acquisition/company_acquired`

Figure 4.6 shows the PR curves for all the competing systems. Distant supervision predicts the top extractions correctly because the multi-instance technique recognizes some common expressions (*e.g.* buy, acquire), but the precision drops dramatically since most positive expressions are overwhelmed by the noise.

4.7 Conclusions

Popular distant supervised approaches have limited ability to handle event extraction, since fluent facts are highly time dependent and often do not exist in any KB. This paper presents a novel unsupervised approach for event extraction that exploits parallel news streams. Our NEWSSPIKE-RE system automatically identifies a set of argument-typed events from a news corpus, and then learns a sentential (micro-reading) extractor for each event.

We introduced a novel, temporal negation heuristic for parallel news streams that identifies event phrases that are correlated, but are not paraphrases. We encoded this in a probabilistic graphical model to cluster sentences, generating high quality training data to learn a sentential extractor. This provides negative evidence crucial to achieving high precision training data.

Experiments show the high quality of the generated training sentences and confirm the importance of our negation heuristic. Our most important experiment shows that we can learn accurate event extractors from this training data. NEWSSPIKE-RE outperforms comparable extractors by a wide margin, more than doubling the area under a precision-recall curve compared to Universal Schemas.

In future work we plan to implement our system as an end-to-end online service. This would allow users to conveniently define events of interest, learn extractors for each event, and return extracted facts from news streams.

Event	F1 @ max recall				area u/ PR curve			area u/ diverse PR curve			
	#	R13	R13P	N-RE	R13	R13P	N-RE	#	R13	R13P	N-RE
acquire(organization,person)	59	.34	.33	**.58**	.26	.26	**.57**	20	.26	.17	**.58**
arrive in(organization,location)	95	.11	.40	**.56**	.01	.12	**.42**	18	.01	.02	**.50**
arrive in(person,location)	130	.61	**.86**	.86	.35	.67	**.93**	18	.26	.33	**.80**
beat(organization,organization)	178	.42	.85	**.90**	.14	.64	**.84**	24	.06	.53	**.58**
beat(person,person)	107	.57	.82	**.94**	.21	.53	**.91**	14	.08	.25	**.77**
buy(organization,organization)	84	.47	.47	**.78**	.25	.50	**.82**	34	.19	.40	**.79**
defend(person,person)	41	.37	.38	**.52**	.36	.47	**.65**	12	.13	.06	**.47**
die at(person,number)	158	.53	.97	**.98**	.31	.93	**.97**	17	.33	.83	**.94**
die(person,time)	179	.85	.91	**.97**	.66	.80	**.96**	16	.22	.63	**.87**
fire(organization,person)	39	.36	.33	**.53**	.32	.45	**.88**	8	.20	.10	**.66**
hit(event,location)	33	.00	.42	**.64**	.00	**.51**	.48	24	.00	.45	**.50**
lead(person,organization/sports_team)	119	.77	.86	**.87**	.57	.73	**.77**	14	.30	.36	**.62**
leave(person,organization)	61	.40	.52	**.59**	.14	.38	**.57**	14	.07	.13	**.38**
meet with(person,person)	137	.74	.86	**.92**	.48	.73	**.88**	14	.28	.56	**.93**
nominate(person/politician,person)	44	.12	.38	**.54**	.13	.44	**.77**	27	.11	.53	**.75**
pay(organization,money)	134	.77	.91	**.93**	.52	.85	**.90**	17	.33	**.90**	.56
place(organization,person)	34	.17	.28	**.50**	.24	.23	**.95**	16	.19	.21	**.94**
play(person/artist,person)	173	**.92**	.89	.87	**.88**	.79	.73	15	**.63**	.56	.47
release(organization,person)	30	.18	.22	**.60**	.08	.25	**.72**	16	.06	.15	**.81**
replace(person,person)	115	.82	.89	**.94**	.62	.75	**.87**	18	.46	.58	**.89**
report(government_agency,time)	140	.37	.84	**.91**	.09	.74	**.83**	35	.06	.52	**.70**
report(written_work,time)	130	.64	**.85**	.83	.43	**.82**	.74	22	.38	**.58**	.51
return to(person/athlete,location)	45	.14	.34	**.50**	.03	.30	**.49**	21	.08	.23	**.78**
shoot(person,number)	101	.71	.89	**.92**	.49	.74	**.84**	8	.35	.37	**.48**
sign with(person,organization)	129	.47	.62	**.89**	.25	.46	**.85**	44	.15	.17	**.91**
sign(organization,person)	110	.45	.71	**.85**	.26	.63	**.79**	26	.15	.27	**.66**
unveil(organization,product)	88	.43	**.71**	.44	.26	**.52**	.30	22	.31	.22	**.63**
vote(government,time)	32	.29	.24	**.74**	.32	.25	**.77**	19	.35	.22	**.83**
win at(person,location)	100	.24	.68	**.85**	.08	.60	**.90**	40	.01	.42	**.90**
win(person,event)	107	.54	.77	**.86**	.22	.63	**.77**	19	.03	.26	**.78**
micro average	2,903	.53	.70	**.81**	.30	.59	**.80**	609	.15	.31	**.71**
macro average	97	.46	.64	**.76**	.30	.56	**.76**	20	.20	.37	**.70**

Table 4.2: Performance of extractors by event relation, reporting both precision and the area under the PR curve. The # column shows the number of true extractions in the pool of sampled output. NEWSSPIKE-RE (labeled N-RE) outperforms two implementations of Riedel's Universal Schemas (See Section 4.6.3 for details). The advantage of NEWSSPIKE-RE over Universal Schemas is greatest on a diverse test set where each sentence has a distinct event phrase.

Chapter 5

NEWSSPIKE-SCALE: HIGH PERFORMANCE EVENT EXTRACTION FOR LARGE, EXTENSIBLE ONTOLOGIES WITH MINIMAL HUMAN EFFORT

We previously described NEWSSPIKE-RE, which achieves high performance on a set of salient relations, where the temporal negation heuristic allows the system to recognize co-occurring but non-synonymous phrases in an unsupervised way. What if a user wishes to flexibly input target relations and create extractors for large ontologies? In this chapter, we present NEWSSPIKE-SCALE, a semi-supervised algorithm that learns high performance event extractors for the user-specified relations with minimal human effort. We present an algorithm to automatically find a list of most informative trigger phrases for a relation, which the users can then tag as positive or negative. We present a series of experiments showing that, with a few minutes' of annotation efforts per relation, the event extractors learned from the generated training data can achieve high and robust performance on a large set of event relations.

5.1 Introduction

Relation Extraction (RE), the process of converting unstructured natural language into structured facts, may open the door to great new opportunities, including general question answering on the Web, advanced human/computer interaction, and perhaps a solid step toward artificial intelligence. Most existing relation extraction systems work on a fixed, small ontology. Why is it challenging to build extractors for large ontologies?

Traditional supervised approaches, which achieve some successes in small ontologies, are limited by the cost of labeling training data and are unlikely to scale to large ontologies. Distant supervision holds the promise of creating the training data automatically by heuristically matching

the ground instances of a knowledge base to the unlabeled corpora. However, distant supervision is limited to static relations that could be found in the pre-existing knowledge bases. They could not be scaled to dynamic, event relations, *e.g.*travel-to (person, location).

Unsupervised approaches have been developed for relation discovery and extractions. They were designed to detect similarity among words and phrases (*i.e.* paraphrasing) from a large unlabeled corpus and then further support building extractors. They are, at least in theory, scalable to large ontologies because vast amounts of unlabeled corpus are cheap and easily available. However, most unsupervised learning approaches essentially rely on the distributional hypothesis (*i.e.* words occurring in similar contexts would have similar meanings). They tend to confuse synonyms and antonyms (*e.g.* rise and fall), cause and effect (*e.g.* shoot and kill). This confusion could seriously mislead the extraction algorithms. Another drawback of unsupervised methods is that they need a lot of occurrence information to provide accurate statistics. Even worse, it is impossible to estimate how much occurrence is necessary. For these reasons, unsupervised methods usually fail to produce high precision relation extractors. In particular, they tend to become unstable on low frequency relations, which occur less often in the corpus.

We previously presented NEWSSPIKE-RE, a novel unsupervised method that exploits parallel news streams and the temporal correspondent heuristics to automatically cluster the training sentences for event relations. It achieved high performance on many salient events discovered in news stories. In particular, unlike traditional unsupervised methods, it avoids the confusion between antonyms and synonyms by leveraging a "temporal negation heuristic" which says that when two phrases are not semantically different if they appear in the same NewsSpike but one of them is in negated form. Unfortunately, when building event extraction on large ontologies, some target relations could be low frequency events. There is no guarantee that the temporal negation heuristic could provide stable signals. In addition, NEWSSPIKE-RE doesn't provide flexibility for users to specify the target relations. What if a user is interested in customizing the ontology?

In this chapter, we present NEWSSPIKE-SCALE, a semi-supervised relation extraction system. Similar to NEWSSPIKE-RE , NEWSSPIKE-SCALE exploits the temporal correspondence heuristics and generates training sentences from parallel news streams. The goal of NEWSSPIKE-SCALE is

to build high performance relation extractors for large, extensible ontologies with minimal human effort. In particular, it should be robust on less frequent relations. NEWSSPIKE-SCALE extends the scale of NEWSSPIKE-RE in three ways:

- It allows users to specify the target relation with great freedom, so it is able to work on large, extensible ontologies
- It automatically discovers the most informative seed phrases for users to annotate, so users can easily customize the extractor
- A bootstrapping algorithm is employed to automatically label more seed phrases

Given the seed phrases, NEWSSPIKE-SCALE can generate training sentences for the relations in the large ontology similar to NEWSSPIKE-RE. Since relations can overlap in large ontologies, we develop a sentential event extractor for overlapping relations. The extractor learns from the generated training sentences and in the testing phase, it extracts events over individual test sentences.

In summary, this chapter makes the following contributions:

- We propose a new framework for relation extraction on large, extensible ontologies that allows users to easily specify new event relations.
- We extend NEWSSPIKE-RE, an unsupervised system, to become a robust, extensible semi-supervised system. With minimal human effort, NEWSSPIKE-SCALE can learn stable extractors even on low frequency relations.
- We present a series of experiments demonstrating that NEWSSPIKE-SCALE builds high performance extractors on an ontology with 150 event relations. In average, the annotation costs less than 6 minutes for each relation.

5.2 System Overview

Our goal is to build high performance extractors for large, extensible ontologies with minimal human effort. This raises the following questions:

- How to define an extensible ontology so a user can easily specify new relations?

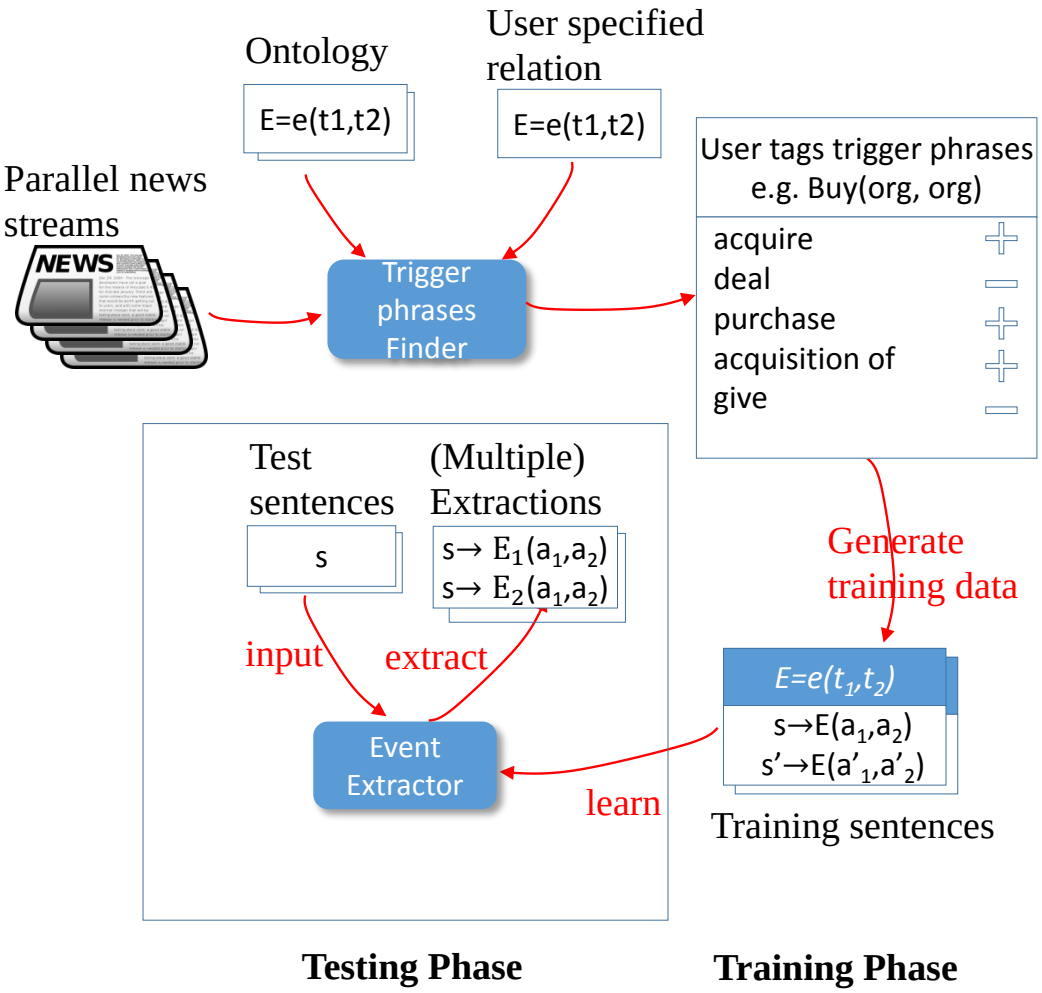

Figure 5.1: System overview of NEWSSPIKE-SCALE: the system allows users to specify new event relations and add them to the target ontology. During the training phase, the system first finds a set of trigger phases for every relation; it presents the trigger phrases to users with their context; NEWSSPIKE-SCALE then employs the graphical model of NEWSSPIKE-RE to generate training sentences, which are used to learn sentential event extractors. During the testing phase, the extractor takes a test sentence as input and predicts event extractions.

- Where do we find the training data for the target relations?
- How could a handful of annotations substantially help relation extraction?
- How could we leverage the annotations to generate the training data and thereafter to learn the extractors?

We will answer these questions in this section. As we did in Chapter 4, we represent a relation in the form of $e(t_1, t_2)$, where e is a representative *event phrase* and t_i are types of the two arguments. Why it is reasonable to represent the event relations in this way? First, a broad range of relations can be represented in this form. Second, it is very convenient for a user to specify new event relations in such forms, which makes the ontology easily extensible. Third, named entity types help to disambiguate the event phrase and make the name of the relation meaningful. And for this reason, when a sentence has two named entities e_1 and e_2 fitting the types of t_1 and t_2, and e_1 and e_2 are connected with the event phrase e, it is very likely that the sentence is a good training sentence for the event. This gives us useful initial sentences to learn the extractor.

Where do we find the training data for a target relation $e(t_1, t_2)$? An unsatisfying approach would be to find a huge unlabeled dataset and to select a small subset to annotate. But when an unlabeled dataset contains billions of sentences and only a tiny fraction of them truly state the target relation, the small subset to annotate would rarely contain any positive sentences, which cannot be used to learn the extractor. Another option is to employ some unsupervised assumption, *e.g.* use the distributional hypothesis, to select a small subset. Unfortunately, most unsupervised methods need unlimited occurrence information to compute the statistics. In fact, the challenge of narrowing the scope from billions of sentences to a handful of examples that humans can read and annotate is almost as hard as the original extraction task itself.

A better option is to generate training data from parallel news streams. As temporal functionality heuristics says (Zhang and Weld [139]), articles from different news sources typically use different sentences to describe the same event, and corresponding sentences can be identified when they mention a unique pair of real-world entities. For example, when an unusual entity pair (Selena, Norway) is suddenly seen in three articles on a single day:

Selena traveled to Norway to see her ex-boyfriend.
Selena arrived in Norway for a rendezvous with Justin.
Selena's trip to Norway was no coincidence.

It is likely that all three refer to the same event relation, *travel-to(person, location)*, and can be used as positive training examples for the relation. Compared with ordinary unlabeled data, sentences in NewsSpikes are highly coherent to each other. This means that we have the opportunity to narrow down the annotation set to a small subset but still include a large variety of expressions that truly state the target relation.

In a manner similar to that of Zhang et al. [138], we group parallel sentences sharing the same argument pair and date in a structure called a NewsSpike. Formally, a NewsSpikeis a tuple (a_1, a_2, d, S), where a_1 and a_2 are arguments, d is a date, and S is a set of argument-labeled sentences $\{(s, a_1, a_2, p) \ldots\}$ in which s is a sentence with arguments a_1 and a_2 and event phrase p. In this chapter, we extend the unsupervised methods of NEWSSPIKE-RE to a semi-supervised system, with a handful of annotations as users' input, in pursuit of robust extractors for large, extensible ontolgoies.

With NewsSpikes and target relations, another question arises: What kind of annotation should we collect? A possible option is to let users to annotate some selected sentences. Unfortunately, annotating sentences has several drawbacks:

- Reading sentences could be slow.
- Only labeling sentences from NewsSpikes could mislead the extractor, even if the label is true.

It may be surprising that even true positive labels on sentences could mislead the extractor. Suppose we are annotating sentences for relation travel-to (person, location) and a user is presented the following two sentences:

Barack Obama tells reporters in Chicago that ....
Barack Obama travels to Chicago.

A user might annotate the first sentence as a true example because he has the knowledge that Obama is living in Washington DC. But unfortunately, this annotation can be misleading and harm-

ful. By analyzing the following sentences, we can more thoroughly illustrate the point:
Barack Obama tells reporters in Washington DC.
Ed Joyner, Jr., to tell reporters in Norfolk. . .

It is obvious that Barack Obama tells reporters in Washington DC doesn't mean travel to(Barack Obama, Washington DC) because he lives there. If a user does not know who Ed Joyner, Jr. is, he probably won't annotate Ed Joyner, Jr., to tell reporters in Norfolk as a positive example. But a user might unconsciously use his background knowledge that Obama lives in Washington DC so if he appears in Chicago, he must travel to the city. So he may annotate Barack Obama tells reporters in Chicago as a positive example. Unfortunately, it is very hard to capture such background knowledge when learning a relation extractor. So this positive example may finally teach the extractors to extract travel to (Barack Obama, Washington DC) from Barack Obama tells reporters in Washington DC and travel to (Ed Joyner, Norfolk) from Ed Joyner, Jr., to tell reporters in Norfolk, which causes errors.

Compared with labeling sentences, it is much faster to label phrases. In addition, users cannot use their background knowledge to make the predictions. Figure 5.1 shows the system overview of NEWSSPIKE-SCALE. We will present the algorithm to find a set of trigger phrases in Section 5.3. After annotation, we can employ the graphical model discussed in Chapter 4 to generate the training sentences, but replacing the heuristic labeled phrases with the tagged trigger phrases.

5.3 Finding Trigger Phrases

The goal of this section is to design a mechanism that allows users to expend a minimal amount of effort to tag a handful of examples that can substantially boost the extraction performance. To pursue this goal, we need to determine:

- How to choose the most informative phrases to tag
- How to display them to users so they can efficiently read and tag them

To answer these questions, we will first introduce several observations, which connect the annotation with the performance of the learned extractors. This will guide us to design an effective

mechanism.

Our observation starts from the temporal functionality heuristics. Let p be a phrase, t_1 and t_2 be two types, we call a tuple (p, t_1, t_2) matching an argument-labeled sentence (s, a_1, a_2, p_s) if $p = p_s$ and the types of a_i are t_i respectively. Similarly, we call (p, t_1, t_2) matching a NewsSpike if it matches at least one of the sentences in the NewsSpike.

When $e(t_1, t_2)$ is the target relation and (e, t_1, t_2) matches a NewsSpike, since the sentences tend to be coherent in that NewsSpike, it is likely that they are good training sentences. From the viewpoint of a particular phrase p, if it appears in many different NewsSpikes which can be matched by (e, t_1, t_2), it is highly likely that p is a paraphrase of e and can suggest the target relation $e(t_1, t_2)$. We would like to collect the tags for p because it is either a true positive, which can teach the extractor to bring many correct extractions, or it is a related but different phrase, which can become useful negative examples in the graphical model of NEWSSPIKE-RE.

Observation 1 *For an event relation $e(t_1, t_2)$, if a phrase p appears in many NewsSpikes that are matching (e, t_1, t_2), the tag of p could be informative for learning the extractors.*

Remember that we usually use precision and recall to evaluate how well an extractor does. The precision is computed by $tp/(tp + fp)$, where tp and fp are true positives and false positives. An extractor will get a low precision score if it predicts many false positives. Our second observation is that not all phrases are equally dangerous in terms of reducing precision. To understand that, suppose there is a NewsSpike containing these sentences

AT&T buys DirectTV.

AT&T meets with DirectTV. . .

If the second sentence AT&T meets with DirectTV is accidentally included in the training sentences, it could teach the extractor to extract buy (organization, organization) from all sentences containing the phrases meet with. Unfortunately, meet with is a very common phrase in English, which means that this incorrect training sentence has the potential to result in a huge amount of false positive extractions, and therefore ruin the precision of the extractor. This leads to our second

observation:

Observation 2 *Incorrectly including a training sentence for a common phrase gives the tag of that common phrase a high impact for the learner, and could result in a large amount of false positives.*

As we discussed in Section 5.2, it is more efficient to annotate phrases than sentences. Tagging phrases could be thought of as a way of quickly tagging a cluster of sentences that share a similar expression.

Note that in NewsSpikes, there can be many sentences that are lexically similar. For example, we find the following sentences in a NewsSpike:

AOL acquires Adops.

AT&T announces deal to acquire DirectTV.

Google's bid to acquire Motorola Mobile ...

IBM announces its plan to acquire Worklight.

These four sentences have four different event phrases acquire, announce deal to acquire, bid to acquire, announces its plan to acquire, while they share the same head word acquire. Obviously, it is a waste of effort to ask a user to tag all of them, because they are so similar to each other. What if we only pick one of them and present it to a user? In fact, it is quite dangerous to do this: suppose we choose IBM announces its plan to acquire Worklight, a user could tag it as negative because the sentence does not present the fact buy (organization, organization). Unfortunately, this negative tag might teach the graphical model that all sentences containing acquire are not good training data and further result in the exclusion of all good extractions with the word acquire. How could we solve this problem? We propose to present a cluster to users,

Observation 3 *It is more efficient to tag a cluster of event phrases sharing the same head word.*

The above three observations inspire us to introduce a multi-level tagging mechanism. For every event relation $e(t_1, t_2)$, it has four levels. The first level is the trigger word level. Users will see a set of trigger words, which are strongly related to the target relation, such as acquire and deal for buy (organization, organization). The second level is the phrase level. For each trigger

word, users will see a set event phrases sharing that trigger word. For the instance of acquire, we will present acquire, announces deal to acquire, bid to acquire, etc. The third level is in-spike sentence level, where we present the sentences from NewsSpikes that match the tuple (e, t_1, t_2). For example, suppose there is a NewsSpike containing

AOL acquires Adops.

AOL buys Adops.

We will present the above two sentences. The goal of this level is to tell users how the trigger word is related to the target relation. The fourth level is the general sentence level. This level is inspired by the second observation, which says that it is necessary to let users know how the trigger word and the cluster of event phrases would perform on the general corpus. Suppose the cluster of the event phrases are P, we find a set of sentences matching (p, t_1, t_2) where $p \in P$.

Figure 5.2 shows an example of presenting two trigger words acquire and deal for the event relation buy (organization, organization). The advantage of the multi-level tagging mechanism is that it allows users to quickly tag trigger words in which they are confident, which are true in most cases (*e.g.* acquire). And it also allows users to scrutinize the context when they need more information, which can avoid many future extraction errors (*e.g.* deal).

To select the top trigger words, we first score every phrase by the number of times it appears in NewsSpikes that match the event relation. Then we group phrases together by their head word, and score the head word by summing up the scores of its individual members. We choose the words with the highest scores as the trigger words and present them to users, together with the corresponding cluster of event phrases, in-spike sentences and general sentences.

5.4 Empirical Evaluation

In this evaluation section, we are answering two questions:

- Could NEWSSPIKE-SCALE allow users to easily create a large, extensible ontology with many event relations, and to quickly tag trigger words for them?
- Could the event extractors learned from the generated training sentences on top of the trigger

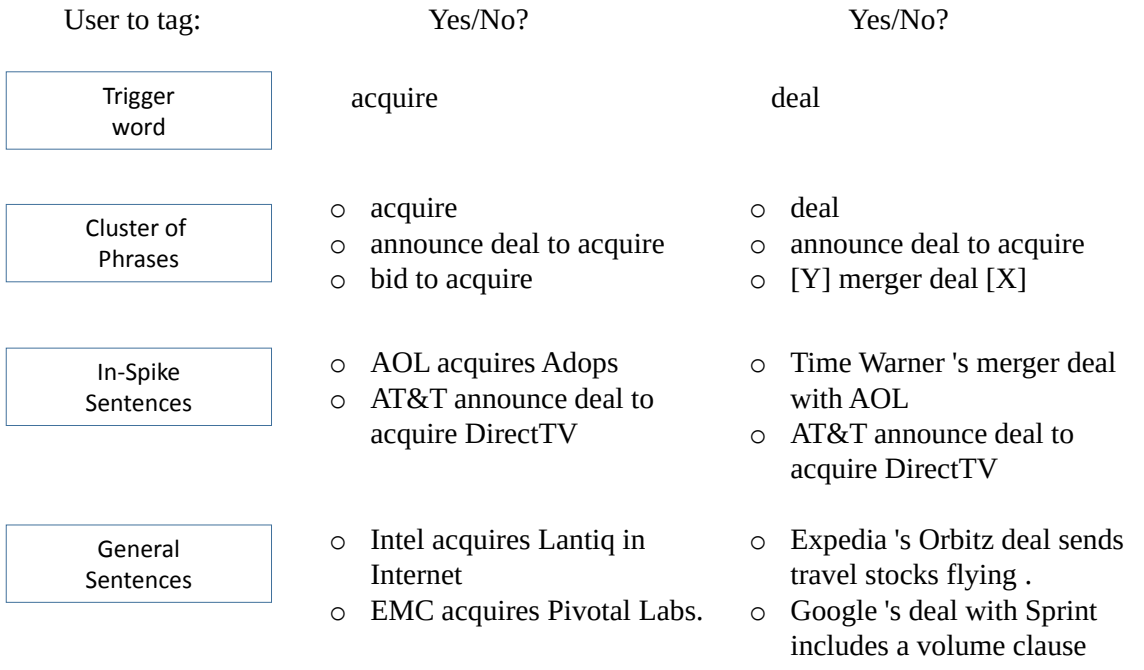

Figure 5.2: An example of two trigger words acquire and deal for the event relation buy (organization, organization). We present the trigger words, the clusters of event phrases, the in-spike sentences and the general sentences in four different levels respectively. Users are asked to tag the trigger words.

event relation	trigger words
dominate(organization,organization)	**dominate** beat defeat **crush rout** win vs. **trounce blow** shut
live in(person,location)	**live** die arrest **base** day enter fly **reside resident** visit
lose to(person,person)	**lose** beat **fall** knock oust defeat **stun** vs. face return
pay(organization,money)	**pay** agree **purchase fine settle settlement buy** deal **spend buy**
play(person/actor,person)	**play star portray** return **character reprise cast role** aka **impersonate**
report(organization,time)	**report release show reveal confirm publish note** estimate obtain warn
score(person,number)	**score add point chip finish contribute drop** hit lead match
warn(location,location)	**warn** accuse **threaten** urge blame action give attack demand talk
warn(person,person)	**warn** urge meet speak **threaten** talk **caution** give play press
win at(person,location)	**win** race **celebrate dominate** lead claim **earn flag conquer cruise**

Table 5.1: Example trigger words for users to tag

words achieve high and robust extraction performance?

In Section 5.4.2, we introduce a method to collect a parallel news stream corpus, and the way we split the training and testing data. In Section 5.4.2, we present how we create the large ontology with 150 event relations, and how much time it costs to tag the trigger words for these relations. In Section 5.4.3, we show the performance of NEWSSPIKE-SCALE and compare it with other systems.

5.4.1 Parallel Dataset

We follow the procedure described in Zhang and Weld [139] to collect parallel news streams and generate the NewsSpikes: first, we get news seeds and query the Bing newswire search engine to gather additional, time-stamped news articles on a similar topic; next, we extract OpenIE tuples from the news articles and group the sentences that share the same arguments and dates into NewsSpikes. We collected the news stream corpus from March 1st, 2013 to April 1st, 2015. We split the dataset into two parts: in the training phrase, we use the news streams in 2013 and 2014 to provide the trigger words and to generate the training sentences (NS14). In the testing phrase, we run the learned extractors on the sentences from news in 2015 (NS15). The parallel news streams is publicly available [1].

5.4.2 Ontology and Tagged Trigger words

To create a large ontology, we find all (p, t_1, t_2) where p is an event phrase and t_i are event types. Each tuple corresponds to a candidate event. We rank these candidate events by the number of NewsSpikes they can match in the training set NS14. Thereafter, we present 1000 candidates to users, together with the brief summary of the NewsSpikes they match.

What users need to do is to select the event relations that interest them by assigning 0 or 1 to each candidate relation. In this way, the author selected 150 event relations in less than 2 hours among the 1000 candidates. We show the resulting relations in Table 5.2.

---

[1] `http://www.cs.washington.edu/ai/clzhang/nsre2/sentences.tokens.gz` and `http://www.cs.washington.edu/ai/clzhang/nsre2/sentences.articleIDs.gz`

Our next step is to tag trigger words for every target relation. We follow the mechanism proposed in Section 5.3 and select 20 trigger words for every relation. In addition, we attach them with the corresponding event phrase clusters, in-spike sentences and general sentences. Figure 5.1 shows 10 example event relations and the selected trigger words. We use bold fonts to indicate these positive triggers.

It takes a user about 15 hours to tag all trigger words. On average, it costs only about 6 minutes to tag one relation. Is such a cost worthwhile? We will show how these tags can boost the extraction performance.

5.4.3 Performance of Event Extraction

Given the tagged trigger words, we use them on the graphical model of NEWSSPIKE-RE as described in Section 4.4. We generate the training sentences from NS14 dataset and then learn the extractor as described in Section 4.5. We run the extractors on NS15 dataset. We compare NEWSSPIKE-SCALE to the following two systems:

- NEWSSPIKE-BASE: Notice that the sentences in the NewsSpikes are highly coherent. In this system, we simply use all sentences from NewsSpikes that match $e(t_1, t_2)$ as the training sentences for the target relation. For example, when there is a NewsSpike AOL acquires Adops. AOL buys Adops. AOL talks with Adops. All three sentences are treated as good training data. This method will introduce some errors, but it tends to have high recall.
- NEWSSPIKE-RE: We generate the training sentences for the 150 relations without using any tagged trigger words as described in Chapter 4. That is, this system is an unsupervised system.

Similar to the empirical study in Chapter 4, we create a pool of extractions by merging the extractions from three systems together. Since the task of event extraction from sentences having different event phrases is more challenging and more interesting, we focus our evaluation on those diverse sentences. We label 20 extractions for every relation and use them as a gold set to compute precision and recall.

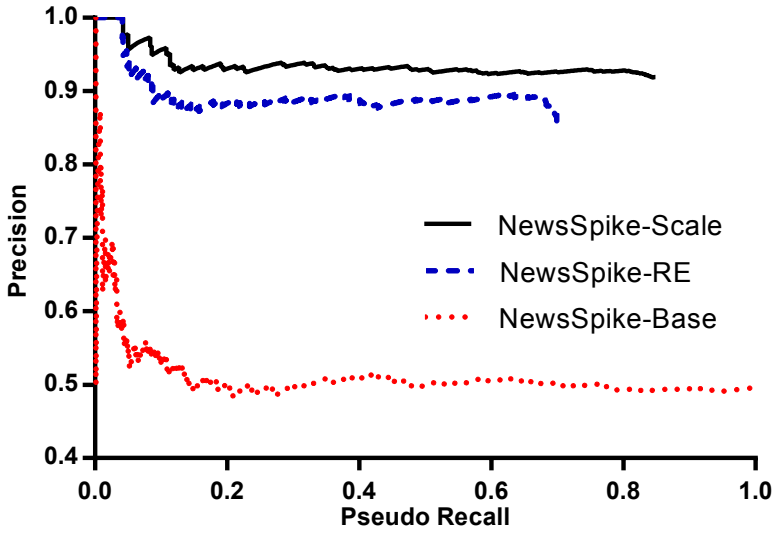

Figure 5.3: Precision pseudo-recall curves for all 150 event relations. NEWSSPIKE-SCALE has AUC 0.79, 25% higher than NEWSSPIKE-RE (0.63) and 54% higher than NEWSSPIKE-BASE (0.51)

Figure 5.3 shows the precision pseudo-recall curve for all sentences for the three systems. NEWSSPIKE-SCALE outperforms the competing systems by a considerable margin. For example, the area under the curve (AUC) of NEWSSPIKE-SCALE for all sentences is 0.79 while that of NEWSSPIKE-RE and NEWSSPIKE-BASE are 0.63 and 0.54. This is a 54% increase over NEWSSPIKE-BASE and a 25% increase over NEWSSPIKE-RE. Similar increases in AUC are observed on F1.

Table 5.3 further lists the breakdown numbers. For space limitation, we only list 30 randomly chosen event relation, as well as the micro and macro average for all 150 relations. Compared with NEWSSPIKE-RE and NEWSSPIKE-BASE , NEWSSPIKE-SCALE obtains the best F1 on 108 out of 150 relations, and best AUC on 107 out of 150 relations. NEWSSPIKE-RE is also successful for many relations. It further shows that our proposed unsupervised approach can effectively exploit the parallel news streams and build accurate extractors. It is interesting to see that NEWSSPIKE-BASE performs reasonably well in several relations. It proves the quality of parallel

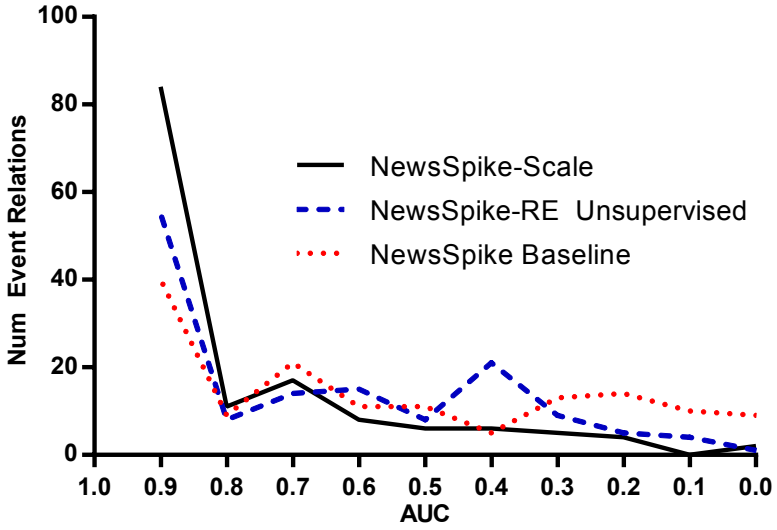

Figure 5.4: Comparing the robustness of three systems. For each system, we bin the Area Under the Curve (AUC) numbers by the scale of 0.1 and compute the number of event relations falling in the bins, and then show curves of the frequencies. More than half of the 150 relations have AUC above 0.90 for NEWSSPIKE-SCALE.

news streams and encourages us to design new algorithms and applications over them.

To further show the robustness of the three systems, we show the AUC frequency distributions in Figure 5.4. For each system, we compute the AUC numbers for every relation and then bin the AUC numbers by the scale of 0.1; we compute the number of event relations falling in the bins, and show the curves of the frequencies. It is clear that more than half of the relations for NEWSSPIKE-SCALE fall in the left, high-performance bin, which the curves for NEWSSPIKE-RE and NEWSSPIKE-BASE are more flat. It shows that NEWSSPIKE-SCALE is more robust and stable with the least amount of human effort.

5.5 Conclusion

Popular distant supervised approaches have limited ability to handle event extraction, since fluent facts are highly time dependent and often do not exist in any knowledge base. NEWSSPIKE-RE was a step forward, with a novel unsupervised approach for event extraction that exploits parallel news streams. In this chapter, we extend NEWSSPIKE-RE to NEWSSPIKE-SCALE , a semi-supervised approach that enables event extraction on large, extensible ontologies. NEWSSPIKE-SCALE allows users to specify the target relation with great freedom, so it is able to work on large, extensible ontologies. NEWSSPIKE-SCALE introduces a new mechanism for users to tag the most informative trigger words. NEWSSPIKE-SCALE then uses the users' input to learn a graphical model to generate the training sentences for the target relations.

Experiments on an ontology with 150 event relations confirms the efficiency of the method for creating the ontology and tagging the trigger words. Experiments further demonstrate the high quality of the learned relation extractors and confirm the importance of the human effort. The area under the curve (AUC) of NEWSSPIKE-SCALE for all sentences is 0.79 while that of NEWSSPIKE-RE and NEWSSPIKE-BASE are 0.63 and 0.54. This is a 54% increase over NEWSSPIKE-BASE and 25% increase over NEWSSPIKE-RE. In addition, the frequency distribution chart on by-relation AUC numbers confirms the robustness of the proposed system.

accuse(location,location)	accuse(person,person)	acknowledge(person,time)
acquire(organization,person)	activate(organization,person)	admit(person,person)
agree(location,time)	allow(organization,organization)	apologize to(person,person)
apologize(person,time)	approve(organization,organization)	arrive in(person,location)
arrive in(person,time)	assure(person,person)	attend(person,organization)
beat(organization,organization)	beat(person,person)	begin(location,time)
believe(person,person)	buy(organization,organization)	cast as(person,person)
challenge(person,person)	confirm(person,time)	confront(person,person)
congratulate(person,person)	criticize(person,person)	defend(person,person)
deny(organization,organization)	die at(person,number)	die in(person,location)
die in(person,time)	die(person,number)	die(person,time)
dominate(organization,organization)	earn(organization,money)	earn(person,money)
embrace(person,person)	end(location,time)	engage to(person,person)
enter(person,organization)	expect(organization,person)	extend(organization,person)
face up to(person,number)	fall to(organization,number)	fall(organization,percent)
fire(organization,person)	follow(person,person)	grow up in(person,location)
guide(person,organization)	hit(person,person)	hope(organization,person)
host(organization,organization)	improve to(organization,number)	insist(person,person)
insist(person,time)	interview(person,person)	join(person,person)
kill(person,person)	lead(organization,number)	lead(person,organization)
lead(person,person)	leave(person,location)	leave(person,organization)
live in(person,location)	lose to(organization,organization)	lose to(person,person)
lose(organization,money)	lose(organization,number)	lose(organization,person)
love(person,person)	marry in(person,location)	marry(person,person)
meet in(location,location)	meet with(person,person)	meet(organization,time)
mock(person,person)	name(organization,person)	nominate(person,person)
offer(organization,money)	offer(organization,person)	open up about(person,person)
order(organization,organization)	pass(person,person)	pay tribute to(person,person)
pay(organization,money)	pick up(person,ordinal)	place(organization,person)
play at(org/sportsteam,location)	play(organization,organization)	play(person/actor,person)
play(person,time)	praise(person,person)	present(person,person)
reach out to(person,person)	recall(organization,person)	recommend(org,org)
replace(person,person)	report(organization,person)	report(organization,time)
resign(person,time)	retire from(person,organization)	return to(person,location)
return to(person,organization)	reunite with(person,person)	rise to(location,number)
rise(organization,percent)	rule(organization,time)	run(person,organization)
save(person,organization)	score for(person,organization)	score with(person,number)
score(person,number)	shake hand with(person,person)	shoot(person,number)
sign with(person,organization)	sign(organization,person)	sit down with(person,person)
speak at(person,organization)	speak in(person,location)	speak with(person,person)
spend(person,time)	split from(person,person)	stay in(person,location)
sue(organization,organization)	suspend(organization,person)	talk(person,person)
testify before(person,organization)	testify on(person,location)	testify(person,time)
trade(organization,person)	turn(person,number)	unveil(organization,product)
urge(location,location)	urge(person,location)	urge(person,organization)
visit(person,location)	visit(person,person)	vote(organization,number)
vote(organization,time)	warn(location,location)	warn(person,location)
warn(person,person)	warn(person,time)	welcome(person,person)
win at(person,location)	win(org/sportsteam,number)	win(organization,time)
win(person,award)	withdraw from(person,event)	work at(person,organization)

Table 5.2: An ontology with 150 event relations

Event	#	F1 @ max recall			area u/ PR curve		
		N-Scale	N-RE	N-b	N-Scale	N-RE	N-b
acknowledge(person,time)	4	**1.0**	**1.0**	0.55	**1.0**	**1.0**	0.30
allow(organization,organization)	4	**1.0**	**1.0**	0.73	**1.0**	**1.0**	0.37
apologize to(person,person)	3	**1.0**	**1.0**	0.40	**1.0**	**1.0**	0.23
beat(person,person)	9	**0.95**	0.74	0.62	**0.99**	0.61	0.34
buy(organization,organization)	12	**0.96**	0.87	0.77	**0.92**	0.77	0.70
challenge(person,person)	5	**0.89**	0.57	0.53	**0.80**	0.40	0.31
criticize(person,person)	11	**0.95**	0.71	0.85	0.91	0.55	**0.92**
dominate(organization,organization)	4	**0.86**	0.67	0.36	**0.75**	0.50	0.13
end(location,time)	2	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
engage to(person,person)	9	0.95	**1.0**	0.82	0.74	**1.0**	0.64
face up to(person,number)	6	0.91	0.80	**1.0**	0.83	0.67	**1.0**
grow up in(person,location)	4	**1.0**	0.67	0.80	**1.0**	0.50	0.80
hit(person,person)	4	0.25	**0.73**	0.44	0.03	0.39	**0.42**
host(organization,organization)	12	**0.96**	0.59	0.75	**0.84**	0.42	0.70
improve to(organization,number)	2	**1.0**	0.67	**1.0**	**1.0**	0.50	**1.0**
insist(person,time)	4	0.89	**1.0**	0.62	0.94	**1.0**	0.27
leave(person,organization)	6	**0.83**	0.67	0.52	**0.80**	0.61	0.27
lose(organization,person)	5	0.89	0.33	**0.91**	0.80	0.20	**0.86**
offer(organization,person)	3	**0.80**	**0.80**	0.60	**0.67**	**0.67**	0.60
pay tribute to(person,person)	6	**0.91**	0.67	0.71	**0.83**	0.50	0.54
play(person/actor,person)	16	0.84	0.84	**0.89**	0.77	0.78	**0.92**
shoot(person,number)	10	**1.0**	0.89	0.95	**1.0**	0.80	0.98
sign(organization,person)	9	**0.80**	0.50	0.64	**0.67**	0.33	0.52
sit down with(person,person)	2	**1.0**	0.80	0.40	**1.0**	0.42	0.13
sue(organization,organization)	7	**0.88**	0.73	0.64	**0.94**	0.57	0.53
testify before(person,organization)	2	0.67	0.67	**1.0**	0.50	0.50	**1.0**
turn(person,number)	4	**1.0**	**1.0**	0.86	**1.0**	**1.0**	0.91
urge(person,organization)	9	0.62	0.56	**0.62**	**0.44**	0.42	0.38
visit(person,person)	10	0.95	0.75	**0.95**	**0.90**	0.60	0.87
warn(person,time)	2	**1.0**	**1.0**	0.18	**1.0**	**1.0**	0.06
win(person,award)	13	**0.92**	0.88	0.90	**0.81**	0.76	0.75
micro average	843	**0.88**	0.77	0.66	**0.79**	0.63	0.51
macro average	6	**0.90**	0.80	0.74	**0.84**	0.71	0.62

Table 5.3: Performance of extractors by event relation, reporting both F1 at maximum recall and the area under the PR curve. The # column shows the number of true extractions in the pool of sampled output. NEWSSPIKE-SCALE (labeled N-scale) outperforms two implementations of NEWSSPIKE-RE (See chapter 4 for details).

Chapter 6

RELATED WORK

In this chapter, we discuss the related work of relation extraction, ontology mapping, paraphrasing, and smart annotation with crowdsourcing.

6.1 Relation Extraction

6.1.1 Supervised Methods

Supervised learning approaches (Soderland *et al.* [113]) have been widely developed for relation extraction. They often focus on a hand-crafted ontology and train the extractor with manually created training data. They often need sets of positive and negative training sentences. For each sentence, a set of features can be extracted from the text. Based on the features, classifiers can be learned to predict new sentences with the corresponding feature vectors. Kambhatla [58] proposed a set of syntactic and semantic features for relation extraction and further used a log-linear model. Zhao and Grishman [140] used SVM with polynomial kernels for classifying different relations, while Guodong *et al.* [48] also used SVM but applied linear kernels.

Since relation extraction involves structured representation, it is natural to exploit richer representations from the sentences. String kernels were discussed in Lodhi *et al.* [72], where the similarity between two strings is computed based on the number of subsequences that are common to both of them. Mooney *et al.* [81] used the word context around the name entities for extracting protein interactions from MEDLINE abstracts. In contrast to the bag-of-words kernels, dependency trees of the sentences could be exploited for kernels. Zelenko *et al.* [135] replaced the strings in the kernel with a structured shallow parse tree built on the sentence. Culotta and Sorensen [33] used another form of rich structural information from trees. Bunescu and Mooney [21] proposed to use the shortest path between the two entities in a dependency parse to represent the relationship

between the entities.

Supervised methods can offer high precision and recall in some situations when appropriate training sets are provided. Unfortunately, finding a suitable set of sentences for annotators to label is a very challenging problem, which makes the annotation prohibitively expensive. The lack of training data makes the supervised methods hard to scale up to large sets of relations. Supervised methods were also developed for event extraction. But they are often domain-specific (*e.g.* biological events(Riedel *et al.* [97], McClosky *et al.* [77]) and entertainment events (Benson *et al.* [15]), Reichart and Barzilay [95]), and are hard to scale over the events on the Web.

6.1.2 Distant Supervised Methods

Distant supervision (also known as weak- or self- supervision) refers to a broad class of methods: Craven and Kumlien [32] introduced the idea by matching the Yeast Protein Database (YPD) to the abstracts of papers in PubMed and training a Naive Bayes extractor. Bellare and McCallum [13] used a database of BibTex records to train a CRF extractor on 12 bibliographic relations.

Several relation extraction systems were built based on collaboratively built Wikipedia articles. The Kylin system applied distant supervision to learn relations from Wikipedia, treating infoboxes as the associated database [125]; Wu *et al.* [124] extended the system to use smoothing over an automatically generated infobox taxonomy. Hoffmann *et al.* [51] describe a system similar to Kylin, but which dynamically generates lexicons in order to handle sparse data, learning over 5000 infobox relations with an average F1 score of 61%. Note that the text of Wikipedia is very different from other text. For example, a common assumption is that an article focuses on an individual entity, which is not true in general articles. Thus the extractors learned from Wikipedia text could hardly be applied immediately to text from other resources.

Mintz *et al.* [80] proposed to learn relation extractors by matching Freebase facts to news articles. In this way, the extractors are more general and could predict relations over Web text. Yao *et al.* [131] perform distant supervision, while using selectional preference constraints to jointly reason about entity types. Riedel *et al.* [100], combine distant supervision and multi-instance learning in a more sophisticated manner, training a graphical model, which assumes only that *at*

least one of the matches between the arguments of a Freebase fact and sentences in the corpus is a true relational mention. MultiR (Hoffman *et al.* [54]) presents a novel approach for multi-instance learning with overlapping relations that combines a sentence-level extraction model with a simple, corpus-level component for aggregating the individual facts. Surdeanu *et al.* [118] extended MultiR by jointly modeling both multiple instances and multiple labels.

Matrix factorization can be used for distant supervision by considering the sentences as the rows of a matrix, and the features or patterns created from the sentences as the columns, in which some of the rows are treated as weakly labeled. Nickel *et al.* [85] factorize YAGO to predict new links. Universal Schema [98] was proposed to treat the surface strings, dependency paths, and relations from KBs as equal "schemas," and then to exploit the correlation between the instances and the schemas from a very large unlabeled corpus. In this work, Riedel *et al.* evaluated only on static relations from Freebase and achieved state-of-the-art performance. But Universal Schemas can be adapted to handle events, by introducing the events as schemas and heuristically finding seed instances.

Distant supervision can use a vast amount of training sentences for free, where there is a corresponding table in the knowledge base. However, there are several limitations of distant supervision. First, it is often hard to locate the table of interest in the database. For example, the relation schema is isCoachedBy(athlete, coach) while the database has tables player(person, team) and coach(person, team) for different sports. It is necessary to consider join, projection and selection to find the correct database views over the background knowledge base. We proposed ontological smoothing to address this problem. Second, people usually only populate static facts to knowledge bases, which makes it hard to apply distant supervised methods on dynamic event relations, such as travel to(person, location). To address this problem, we generate training data from parallel news streams, which exclusively contain a large variety of fluents and dynamic events. Third, distant supervision often creates very noisy training sentences, especially for the time-dependent relations. Although there are techniques like multi-instance and multi-label learning, it is still hard to create high precision and high recall extractors from sentences of extremely low quality. This work exploits the temporal correspondences to generate high quality training

sentences to address this problem.

There are other approaches that target different kinds of background knowledge. Some approaches (Chang *et al.* [25], Smith and Eisner [110], Bellare and McCallum [14]) allows learning with soft constraints - for example, in the form of labeled features. WordNet [115] could be used to learn more general extraction patterns, and Cohen and Sarawagi [29] used domain-specific dictionaries. Hierarchical structure of an ontology (McCallum *et al.* [76], Wu and Weld [124]) was leveraged to smooth parameter estimates of a learned model.

6.1.3 Open Information Extraction

Open Information Extraction systems perform self-supervised learning of relation-independent extractors. They do not assume a set of relations pre-defined in an ontology, but use the surface strings from the text to represent the relations. The advantage of Open IE systems is that they can read arbitrary text from any domain on the Web, and extract meaningful information by converting the unstructured text into tuples and tables.

Preemptive IE (Shinyama and Sekine [107]) and On-Demand IE (Sekine [105]) avoid relation-specific extractors, but rely on document and entity clustering, which is too costly for Web-scale IE. The first Web-scale Open IE system was TextRunner [7, 8], which used a Naive Bayes model with unlexicalized POS and NP-chunk features, and trained using examples heuristically generated from the Penn Treebank. The WOE systems [127] introduced by Wu and Weld make use of Wikipedia as a source of training data for their extractors, which leads to further improvements over TextRunner. Reverb [41] uses shallow syntactic processing to identify relation phrases that begin with a verb and occur between the argument phrases. OLLIE (Mausam *et al.* [104]) expanded the syntactic scope of relation phrases to cover a much larger number of relation expressions, and expanded the Open IE representation to allow additional context information such as attribution and clausal modifiers. An OpenIE system was also proposed to extract events (Ritter *et al.* [101]) from Twitter data.

Open IE methods can scale to millions of documents by performing self-supervised learning of relation-independent extractions. But they are unable to output canonicalized relations. As a

result, applications built upon Open IE must deal with homonymy and synonymy challenges. In this work, we use Open IE systems to convert text to semi-structured tuples and use them as the names of the event relations, and also the event phrases in NEWSSPIKE-RE's graphical models.

6.1.4 Unsupervised Learning

Unsupervised approaches have been developed for relation discovery and extractions. These algorithms are usually based on some clustering assumptions over a large unlabeled corpus. Common assumptions include the distributional used by Hasegawa *et al.* [50] and Shinyama and Sekine [109], latent topic assumption by Yao *et al.* [133, 132], and low rank assumption by Takamatsu *et al.* [120] and Riedel *et al.* [98]. Since the assumptions largely rely on co-occurrence, previous unsupervised approaches tended to confuse correlated but semantically different phrases during extraction.

Poon and Domingos [89] proposed Unsupervised Semantic Parsing to transform dependency trees into quasi-logical forms and cluster them by their semantics. Their OntoUSP system [91] extended the USP technique to the problem of knowledge acquisition from text. Markov Logic Network is used to encode the human knowledge and rules in first order logic rules to improve the clustering performance. But it can also be very time consuming to develop many rules when we scale up the techniques to large ontologies. In addition, these rules can easily become too complicated for even modern computers to do inference and learning.

In contrast to previous approaches, our unsupervised system NEWSSPIKE-RE largely avoids these errors by exploiting the temporal negation heuristic in parallel news streams. In addition, unlike many unsupervised algorithms requiring human effort to canonicalize the clusters, our work automatically discovers events with readable names.

6.1.5 Bootstrapping

Bootstrapping is another common extraction technique. This typically takes a set of seeds as input, which can be ground instances or key phrases. The algorithms then iteratively generate more positive instances and phrases. Brin [20] proposed the Dual Iterative Pattern Relation Expansion

(DIPRE) method to identify authors of books. Snowball (Agichtein and Gravano [1]) has similar system architecture as DIPRE to extract locate in relations. Huang and Riloff [57] proposed a bootstrapped dictionary to recognize civil unrest events.

While there are some successful examples of bootstrapping on small ontologies, the challenge is to avoid semantic drift when it is applied on a large scale. Kozareva and Hovy [62, 56, 61] suggested ways to avoid this problem by using doubly-anchored patterns as well as graph structures. Large-scale systems often require extra processing such as manual validation between the iterations or additional negative seeds as the input. The NELL system [23] had the initial knowledge consisting of a selectional preference constraint and 20 ground fact seeds. NELL then matched entity pairs from the seeds to a Web corpus, but instead of learning a probabilistic model, it bootstrapped a set of extraction patterns using semi-supervised methods for multi-task learning. The SOFIE system [117] integrated logical constraint reasoning with pattern-based bootstrapping, and cast the problem into Max-Sat solver. PROSPERA [83] extended this with a new notion of n-gram item sets for richer patterns.

6.2 Ontology Mapping

Dhamankar *et al.* [35] define schema *matching* to be the first step in the process of constructing a *mapping*, *i.e.* a function converting descriptions of objects in one ontology into corresponding descriptions in another. We consider ontologies comprised of *types* (unary relations, also known as concepts, organized in a taxonomy) and binary *relations*. Relations may connect two types (*e.g.*, *Person*) or may link a type to a primitive value, such as numbers, dates and strings (*e.g.*, *BirthDate*), which are often called *attributes* or *properties*. Each type is associated with a set of instances, called *entities*.

A *mapping* from a background ontology onto a target ontology is a set of partial functions whose ranges are entities, types and relations in the target ontology. Ullman [121] noted that these mappings can be thought of as view definitions, *e.g.* defined using SQL operations such as selection, projection, join and union.

Euzenat and Shvaiko [40], Rahm and Bernstein [92] carve the set of approaches for ontology

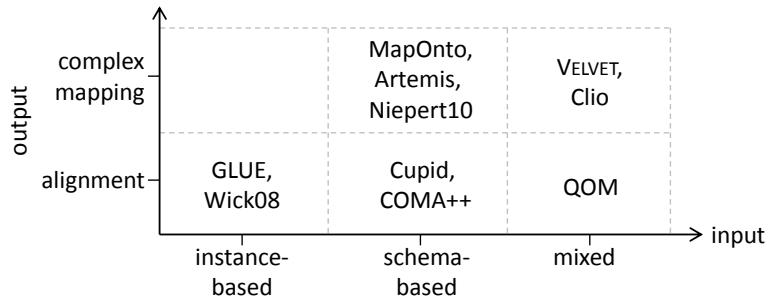

Figure 6.1: Classification of selected ontology matching systems, based on Euzenat and Shvaiko[40].

matching into several dimensions. The input of the matching algorithm can be *schema-based*, *instance-based* or *mixed*. The output can be an *alignment* (*i.e.*, a one-to-one function between objects in the two ontologies) or a *complex mapping* (*e.g.*, defined as a view). Figure 6.1 plots some previous methods along these dimensions.

The majority of existing systems focus on the alignment problem. Doan *et al.* [36] present GLUE, which casts alignment of two taxonomies into classification and uses learning techniques. The more recent system by Wick and McCallum [123] applies a learning approach to a single probabilistic model that considers all matching decisions jointly. While these systems operate on instances, others align schemas: Cupid [73] matches tree-structures in three phases, that include linguistic matching, structural matching, and aggregation. COMA++[6] enables parallel composition of matching algorithms. Niepert *et al.* [86] propose a joint probabilistic model based on Markov logic. QOM [39] matches both, instances and schemas, and is able to trade off between efficiency and quality.

Far less work has been done with finding complex mappings between ontologies. Artemis [24] creates global views using hierarchical clustering of database schema elements. MapOnto [2] produces mapping rules between two schemas expressed as Horn clauses. Miller *et al.*'s tool Clio [78][79] generates complex SQL queries as mappings, and ranks these by heuristics.

For ontological smoothing to work, it is essential that one can find complex mappings involving

selections, projections, joins, and unions. While MapOnto and Clio handle complex mappings, they are semi-automatic tools that depend on user guidance. In contrast, we designed VELVET to be fully autonomous. Unlike the other two, VELVET uses a probabilistic representation and performs joint inference to find the best mapping.

6.3 Paraphrasing

The vast majority of paraphrasing work falls into two categories: approaches based on the distributional hypothesis or on correspondences between parallel corpora (Anfroutsopoulos and Malakasiotis [3] and Madnani and Dorr [74]).

6.3.1 Using Distributional Similarity

Lin and Pantel's DIRT [67] employs mutual information statistics to compute the similarity between relations represented in dependency paths. Resolver [134] introduces a new similarity metric called the Extracted Shared Property (ESP) and uses a probabilistic model to merge ESP with surface string similarity.

Identifying the semantic equivalence of relation phrases is also called *relation discovery* or *unsupervised semantic parsing*. Often, techniques don't compute the similarity explicitly but rely implicitly on the distributional hypothesis. Poon and Domingos' USP [90] clusters relations represented with fragments of dependency trees by repeatedly merging relations having similar context. Yao *et al.* [132, 133] introduces generative models for relation discovery using an LDA-style algorithm over a relation-feature matrix. Chen *et al.* [26] focuses on domain-dependent relation discovery, extending a generative model with meta-constraints from lexical, syntactic, and discourse regularities.

Our NEWSSPIKE-PARA solves a major problem with these approaches, avoiding errors such as confusing synonyms with antonyms and causes with effects. Furthermore, NEWSSPIKE-PARA doesn't require massive statistical evidence as do most approaches based on the distributional hypothesis.

6.3.2 Using Parallel Corpora

Comparable and parallel corpora, including news streams and multiple translations of the same story, have been used to generate paraphrases, both sentential (Barzilay and Lee [10], Dolan *et al.* [37], Shinyama and Sekine [108]) and phrasal (Barzilay and McKeown [11], Shen *et al.* [106], Pang *et al.* [87]). Typical methods first gather relevant articles and then pair sentences that are potential paraphrases. Given a training set of paraphrases, models are learned and applied to unlabeled pairs (Dolan and Brockett [38], Socher *et al.* [112]). Phrasal paraphrases are often obtained by running an alignment algorithm over the paraphrased sentence pairs.

While prior work uses the temporal aspects of news streams as a coarse filter, it largely relies on text metrics, such as context similarity and edit distance, to make predictions and alignments. These metrics are usually insufficient to produce high precision results; moreover they tend to produce paraphrases that are simple lexical variants (*e.g.* {*go to, go into*}.). In contrast, NEWSSPIKE-PARA generates relation clusters with both high precision and high diversity.

In recent years, the Paraphrase Database (PPDB)[1] has been constructed by exploiting the bilingual parallel corpora. PPDB version 1.0 [46] follows Bannard and Callison-Burch [9]'s bilingual pivoting method, which assumes that two English strings that translate to the same foreign string have the same meaning. PPDB version 2.0 [45] includes a discriminatively re-ranked set of paraphrases that achieve a higher correlation with human judgments than PPDB version 1.0's heuristic rankings. Although PPDB and other paraphrase datasets have been shown useful for a variety of natural language processing tasks, they are not enough to learn the relation extractors for several reasons: first, the semantics of the paraphrases are often context dependent; second, the generated paraphrases are often in small clusters and it remains challenging to merge them for the purpose of training an extractor. Finally, a reliable source of negative training data is needed to complement even a large set of paraphrases, and these negative examples are best if they are "near misses." Our work extends previous paraphrasing techniques, notably that of Zhang and Weld [139], but we focus on generating high-quality positive and negative training sentences for the discovered events

[1] `http://paraphrase.org/#/`

in order to learn extractors with high precision and recall.

6.3.3 Other Related Work

Textual entailment [34], which finds a phrase implying another phrase, is closely related to the paraphrasing task. Berant *et al.* [16] notes the flaws in distributional similarity and proposes local entailment classifiers, which are able to combine many features. Lin *et al.* [68] also use temporal information to detect the semantics of entities. In a manner similar to our approach, Recasens *et al.* [94] mine parallel news stories to find opaque coreferent mentions.

6.4 Crowdsourcing and Relation Extraction Tools

We have seen that a traditional supervised learning framework is not enough for relation extraction, since it is extremely hard to locate an appropriate set of examples for users to annotate. So a natural question is, could we design some smart strategies that could collect human intelligence for relation extraction? One idea is to first let a large amount of workers annotate the dataset, and then merge the annotations correctly. The second idea is to implement relation extraction tools that allow users to quickly build their extractors.

The crowdsourcing platforms like Amazon Mechanical Turk and oDesk make it less costly to annotate training examples. In particular, an active learning framework could be used to select the most informative data.

Since most annotation is performed by unevenly-trained crowdsourced workers, errors could be rampant. It is common practice to request a dozen or more duplicate labels to ensure peak annotation accuracy in natural language processing (Snow *et al.* [111]). The need to duplicate annotation increases the cost of annotation and often invalidates current methods for active learning. Many efforts have been used to enhance the simple majority vote mechanism. Whitehill *et al.* [122] and others have developed a variety of expectation-maximization (EM)-style algorithms by learning the worker's skill levels. But learning worker's skills could be very hard and costly. So decision-theoretic control is proposed to decide which are the best questions to ask and allocate specific

tasks to the most appropriate workers. CASCADE [27] created a globally consistent taxonomy by crowdsourcing micro work from many individuals. DELUGE [19] improved CASCADE, using significantly less crowd labor, but produced comparable quality results. CLOWDER [65] provides the user with an adaptive programming language so that non-experts can write POMDPs without knowing anything about them. Lin *et al.* [66] studied the problem of when it is more efficient to relabel an existing example and when to label a new example.

The potential of using crowdsourcing for relation extraction attracts increasing interest nowadays and is an active relation area. Gormley *et al.* [47] introduced a design that allowed non-expert to correct automatically generated relation annotations. Pershina *et al.* [88] and Angeli *et al.* [5] proposed systems to combine the benefits of supervision for difficult examples with the coverage of a large distantly supervised corpus, and achieved a significant performance increase. Liu *et al.* [71] combined the crowdsourcing techniques with relation extraction algorithms, and showed that careful attention to crowdsourcing quality control could yield much larger improvements.

Soderland *et al.* [114] showed that Open IE can form the basis of a high precision extractor for a set of target relations, and the extractor can be built with a minimum of human knowledge engineering in 3 hours. Hoffmann [52] designed Readr that allowed users to quickly and interactively create rule-based extractors. SystemT [28] from IBM is built around AQL, a declarative rule language enables an efficient execution plan for the annotator. Based on that, WizIE and other systems [63, 129] lower the barriers to entry into text analytics for novice developers.

DeepDive [128] enables one to tackle extraction, integration, and prediction problems in a single system. It allows users to rapidly construct sophisticated end-to-end data pipelines. They have achieved remarkable success on many tasks, *e.g.* the 2014 KBP's slot filling task [4].

Chapter 7

FUTURE WORK

In this chapter, we would like to look at our challenges more broadly. We set ourselves the goal of high performance relation extraction with minimal human effort. We have proposed ontological smoothing and temporal correspondence techniques to approach this goal. What should be our next milestones? How could we generalize our extraction task? How could we further improve the extraction performance? How could we enable users to better use their effort? What changes need to happen so we can reach these milestones?

In this work, we define our target relations in the form of the event phrases with the argument types. It is a very natural way to represent the target relations, but it obviously is not the only way. For example, what if the relation is described in a piece of text? Or the arguments of the relation are not name entities? Or the relation has only one argument or multiple arguments? We believe ontological smoothing and temporal correspondence would still be very useful. For ontological smoothing, it first remains unclear how distant supervision itself could be applied to general relations. For example, when the target relation is a unary relation, simply using one name entity to heuristically match the training sentences would be unlikely to result in a useful dataset. It would be very important for us to develop advanced distantly supervised algorithms to handle this. Second, we need to develop methods to create the complicated database views. It is probable that a few seeds would not provide enough supervision for this purpose, so it might be more effective to introduce some new mechanism and interfaces for users to define rules over the two ontologies. These rules could further lead to the training examples for the relations. For temporal correspondence, one simple idea would be using the collected paraphrases as the features on other relation extraction systems. Another idea would be mapping the target relations to the $e(t_1, t_2)$ relations. It would certainly enhance the extraction performance if we could design specific extractors for

different types of relations, based on the general temporal correspondence heuristics.

It is widely believed that unsupervised approaches are more promising for real learning problems because they do not need expensive labeled data. However, it is ironic that there are no "pure" unsupervised settings in practice, because we need to label some examples for the purpose of evaluation. When we face a new real-world problem, the first challenge for us could often be the way to evaluate the outputs instead of the learning algorithms. But it doesn't mean that unsupervised methods are not useful. Unsupervised learning methods could often quickly give us a reasonable baseline, and their errors could tell us the challenges of the problem. Possibly, the unsupervised learning methods alone may be good enough, but they could leave big room for improvement. Active learning has been used to show which examples are most informative and worth labeling. But note that, many times, the costs to develop the extractors could be even higher than the cost of annotations. So could we use unsupervised learning methods to tell the engineers how to develop the extractor? In the future, we need to introduce some learning systems that combine the advantages of both unsupervised and supervised learning for large-scale relation extraction. In particular, it should peek at the large amounts of data and find out the best way for the developers to attack the problem.

Nowadays, researchers have proposed many types of weakly supervised signals and shown that algorithms learned from those weak signals could even outperform supervised learning methods. Usually it does not work very well to use these signals naively. So researchers proposed various well-designed learning algorithms to encode these signals for the extractor. However, it remains unclear how these weak supervised signals could be combined together for some greater purpose? In particular, is it possible to design some general framework that allows users to apply different learning signals flexibly? In pursuit of this goal, we need to propose some intermediate component that makes the various signals become transparent to the underlying extractors, and transmit the weak signals into the intermediate component. Furthermore, weak signals could be used to evaluated the extractors. For example, previous work using distant supervision often evaluate the aggregate-level performance of the extractor merely based on entity pairs retrieved from the knowledge base. These methods enable quick evaluations. But since there are many errors

coming together with the weak signals, the precision recall numbers could be very inaccurate. It is interesting to investigate how we could accurately evaluate the extractors with those weak learning signals.

We discussed in the last chapter that researchers and industries are developing complex, integrated extraction systems, such as SystemT and DeepDive. While individual learning algorithms are important components in these integrated systems, the database engines and human interaction interfaces are also crucial. In the future, we would like to incorporate ontological smoothing and temporal correspondence into these integrated extraction systems. It is likely that users will find novel heuristics from the datasets for their specific domains that we haven't discovered.

Today, governments and industries have greatly benefited from relation extraction systems by having the resources to hire expert operators to build the extractors. How could our work help ordinary people? Question answering and search are two obvious areas in which relation extraction could be helpful. In the future, we need to develop helpful software and applications for ordinary users. Probably, the application should be built on top of today's search platforms because users would feel most comfortable using them. In addition, the techniques of relation extraction and summarization could be combined to provide users smoother results.

We also hope to make our temporal correspondence heuristics more useful. In this work, we focus on sentences sharing the same date. In this way, we can generate more accurate training sentences. But it is natural to consider sentences from two different but close dates. By using them, we could have greater opportunities to increase the recalls, but we are also facing the challenges of reduced precision. It requires us to propose advanced observations and algorithms to handle the challenges.

Chapter 8

CONCLUSION

Relation extraction, the process of extracting structured information from natural language text, grows increasingly important for Web search and question answering applications. Traditional supervised approaches, which can achieve high precision and recall, are limited by the cost of labeling training data. Distant supervision creates its own training data by matching the ground instances of a knowledge base to the unlabeled text. But they are limited by the scope of relation instances that have been populated into the knowledge bases. In particular, they cannot handle event relations that are crucial for making informed decisions.

This dissertation considered two major ideas: ontological smoothing and temporal correspondence. The idea of ontological smoothing is to map the target relations to database views over a background knowledge base, and thus allow distant supervision to work on the user-specified relations. The idea of temporal correspondence is to exploit the highly coherent sentences from parallel news streams to provide accurate training sentences for the extractors.

We presented four systems, VELVET, NEWSSPIKE-PARA, NEWSSPIKE-RE, and NEWSSPIKE-SCALE-based on ontological smoothing and temporal correspondence. VELVET generates a mapping between the target relations and a background knowledge base using database join, union, project, and select operators. NEWSSPIKE-PARA avoids the confusion between synonyms and antonyms and generates the paraphrases from parallel news streams. NEWSSPIKE-RE is an unsupervised algorithm that discovers event relations and then learns to extract them. NEWSSPIKE-SCALE is a semi-supervised algorithm that learns high performance event extractors for the user-specified relations with minimal human effort. Our work makes progress toward solving many of the problems related to large-scale relation extraction.

BIBLIOGRAPHY

[1] Eugene Agichtein and Luis Gravano. *Snowball*: extracting relations from large plain-text collections. In *Proceedings of the ACM Conference on Digital Libraries (DL)*, pages 85–94, 2000.

[2] Yuan An, Alex Borgida, and John Mylopoulos. Discovering the semantics of relational tables through mappings. In *LNCS 4244 - Journal on Data Semantics VII*, pages 1–32, 2006.

[3] Ion Androutsopoulos and Prodromos Malakasiotis. A survey of paraphrasing and textual entailment methods. In *Journal of Artificial Intelligence Research*, pages 135–187, 2010.

[4] Gabor Angeli, Sonal Gupta, Melvin Jose, Christopher D Manning, Christopher Ré, Julie Tibshirani, Jean Y Wu, Sen Wu, and Ce Zhang. Stanfords 2014 slot filling systems. *TAC KBP*, 2014.

[5] Gabor Angeli, Julie Tibshirani, Jean Y Wu, and Christopher D Manning. Combining distant and partial supervision for relation extraction. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014.

[6] David Aumueller, Hong-Hai Do, Sabine Massmann, and Erhard Rahm. Schema and ontology matching with coma++. In *SIGMOD*, pages 906–908, 2005.

[7] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the Web. In *Procs. of IJCAI*, 2007.

[8] Michele Banko and Oren Etzioni. The tradeoffs between open and traditional relation extraction. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics (ACL-08)*, pages 28–36, 2008.

[9] Colin Bannard and Chris Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 597–604. Association for Computational Linguistics, 2005.

[10] Regina Barzilay and Lillian Lee. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *HLT-NAACL*, pages 16–23. Association for Computational Linguistics, 2003.

[11] Regina Barzilay and Kathleen R McKeown. Extracting paraphrases from a parallel corpus. In *ACL*, pages 50–57. Association for Computational Linguistics, 2001.

[12] Regina Barzilay, Kathleen R McKeown, and Michael Elhadad. Information fusion in the context of multi-document summarization. In *ACL*, pages 550–557. Association for Computational Linguistics, 1999.

[13] Kedar Bellare and Andrew McCallum. Learning extractors from unlabeled text using relevant databases. In *Sixth International Workshop on Information Integration on the Web*, 2007.

[14] Kedar Bellare and Andrew McCallum. Generalized expectation criteria for bootstrapping extractors using record-text alignment. In *EMNLP*, pages 131–140, 2009.

[15] Edward Benson, Aria Haghighi, and Regina Barzilay. Event discovery in social media feeds. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, pages 389–398, 2011.

[16] Jonathan Berant, Ido Dagan, and Jacob Goldberger. Global learning of typed entailment rules. In *ACL-HLT*, pages 610–619. Association for Computational Linguistics, 2011.

[17] Rahul Bhagat and Deepak Ravichandran. Large scale acquisition of paraphrases for learning surface patterns. In *ACL*, volume 8, pages 674–682. Association for Computational Linguistics, 2008.

[18] Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.

[19] Jonathan Bragg, Daniel S Weld, et al. Crowdsourcing multi-label classification for taxonomy creation. In *First AAAI conference on human computation and crowdsourcing*, 2013.

[20] Sergey Brin. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*, pages 172–183. 1999.

[21] Razvan C Bunescu and Raymond J Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724–731. Association for Computational Linguistics, 2005.

[22] Chris Callison-Burch, Philipp Koehn, and Miles Osborne. Improved statistical machine translation using paraphrases. In *NAACL*, pages 17–24. Association for Computational Linguistics, 2006.

[23] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka Jr., and Tom M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI-10)*, 2010.

[24] Silvana Castano and Valeria De Antonellis. Artemis: Analysis and reconciliation tool environment for multiple information sources. In *SEBD*, pages 341–356, 1999.

[25] Ming-Wei Chang, Lev-Arie Ratinov, and Dan Roth. Guiding semi-supervision with constraint-driven learning. In *ACL*, 2007.

[26] Harr Chen, Edward Benson, Tahira Naseem, and Regina Barzilay. In-domain relation discovery with meta-constraints via posterior regularization. In *ACL-HLT*, pages 530–540. Association for Computational Linguistics, 2011.

[27] Lydia B Chilton, Greg Little, Darren Edge, Daniel S Weld, and James A Landay. Cascade: Crowdsourcing taxonomy creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008. ACM, 2013.

[28] Laura Chiticariu, Vivian Chu, Sajib Dasgupta, Thilo W Goetz, Howard Ho, Rajasekar Krishnamurthy, Alexander Lang, Yunyao Li, Bin Liu, Sriram Raghavan, et al. The systemt ide: an integrated development environment for information extraction rules. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 1291–1294. ACM, 2011.

[29] William W. Cohen and Sunita Sarawagi. Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 89–98, 2004.

[30] Michael Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.

[31] Mark Craven and Johan Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 77–86, 1999.

[32] Mark Craven and Johan Kumlien. Constructing biological knowledge bases by extracting information from text sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86, 1999.

[33] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423. Association for Computational Linguistics, 2004.

[34] Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. Recognizing textual entailment: Rational, evaluation and approaches. *Natural Language Engineering*, 15(04):i–xvii, 2009.

[35] Robin Dhamankar, Yoonkyong Lee, Anhai Doan, Alon Halevy, and Pedro Domingos. imap: Discovering complex semantic matches between database schemas. In *SIGMOD*, pages 383–394, 2004.

[36] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Learning to map between ontologies on the semantic web. In *Proceedings of the Eleventh International WWW Conference*, 2002.

[37] Bill Dolan, Chris Quirk, and Chris Brockett. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Computational Linguistics*, page 350. Association for Computational Linguistics, 2004.

[38] William B Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of IWP*, 2005.

[39] Marc Ehrig and Steffen Staab. Qom c quick ontology mapping. In *ISWC*, pages 683–697, 2004.

[40] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

[41] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1535–1545, 2011.

[42] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Paraphrase-driven learning for open question answering. In *ACL*. Association for Computational Linguistics, 2013.

[43] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 363–370, 2005.

[44] William A Gale, Kenneth W Church, and David Yarowsky. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pages 233–237. Association for Computational Linguistics, 1992.

[45] Ellie Pavlick1 Pushpendre Rastogi2 Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. *Volume 2: Short Papers*, page 425.

[46] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB: The paraphrase database. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2013)*, pages 758–764, 2013.

[47] Matthew R Gormley, Adam Gerber, Mary Harper, and Mark Dredze. Non-expert correction of automatically generated relation annotations. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 204–207. Association for Computational Linguistics, 2010.

[48] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics, 2005.

[49] Zellig S Harris. Distributional structure. *Word*, 1954.

[50] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *ACL*, page 415. Association for Computational Linguistics, 2004.

[51] R. Hoffmann, C. Zhang, and D. Weld. Learning 5000 relational extractors. In *ACL*, Uppsala, Sweden, July 2010.

[52] Raphael Hoffmann. Interactive learning of relation extractors with weak supervision. *dissertation*.

[53] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*, 2011.

[54] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL-HLT*, pages 541–550, 2011.

[55] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke S. Zettlemoyer, and Daniel S. Weld. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 541–550, 2011.

[56] Eduard Hovy, Zornitsa Kozareva, and Ellen Riloff. Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 948–957. Association for Computational Linguistics, 2009.

[57] Ruihong Huang and Ellen Riloff. Multi-faceted event recognition with bootstrapped dictionaries. In *the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, pages 41–51, 2013.

[58] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, page 22. Association for Computational Linguistics, 2004.

[59] Rohit J. Kate and Raymond J. Mooney. Joint entity and relation extraction using card-pyramid parsing. In *COLING*, 2010.

[60] Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. Boilerplate detection using shallow text features. In *WSDM*, pages 441–450. ACM, 2010.

[61] Zornitsa Kozareva and Eduard Hovy. Learning arguments and supertypes of semantic relations using recursive patterns. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1482–1491. Association for Computational Linguistics, 2010.

[62] Zornitsa Kozareva, Ellen Riloff, and Eduard H Hovy. Semantic class learning from the web with hyponym pattern linkage graphs. In *ACL*, volume 8, pages 1048–1056, 2008.

[63] Yunyao Li, Laura Chiticariu, Huahai Yang, Frederick R Reiss, and Arnaldo Carreno-Fuentes. Wizie: a best practices guided development environment for information extraction. In *Proceedings of the ACL 2012 System Demonstrations*, pages 109–114. Association for Computational Linguistics, 2012.

[64] Percy Liang, A. Bouchard-Côté, Dan Klein, and Ben Taskar. An end-to-end discriminative approach to machine translation. In *COLING/ACL*, 2006.

[65] Christopher H Lin, Daniel S Weld, et al. Towards a language for non-expert specification of pomdps for crowdsourcing. In *First AAAI Conference on Human Computation and Crowdsourcing*, 2013.

[66] Christopher H Lin, Daniel S Weld, et al. To re (label), or not to re (label). In *Second AAAI Conference on Human Computation and Crowdsourcing*, 2014.

[67] Dekang Lin and Patrick Pantel. Discovery of inference rules for question-answering. *Natural Language Engineering*, 7(4):343–360, 2001.

[68] Thomas Lin, Oren Etzioni, et al. No noun phrase left behind: detecting and typing unlinkable entities. In *EMNLP*, pages 893–903. Association for Computational Linguistics, 2012.

[69] Xiao Ling and Daniel S Weld. Fine-grained entity recognition. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2012.

[70] Xiao Ling and Daniel S. Weld. Fine-grained named entity tagging. In *AAAI*, 2012.

[71] Angli Liu, Stephen Soderland, and Daniel S Weld. Crowd supervision for relation extraction.

[72] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *The Journal of Machine Learning Research*, 2:419–444, 2002.

[73] Jayant Madhavan, Philip Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *VLDB*, pages 49–58, 2001.

[74] Nitin Madnani and Bonnie J Dorr. Generating phrasal and sentential paraphrases: A survey of data-driven methods. *Computational Linguistics*, 36(3):341–387, 2010.

[75] Marie-Catherine De Marneffe, Bill Maccartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2006.

[76] Andrew McCallum, Ronald Rosenfeld, Tom M. Mitchell, and Andrew Y. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 359–367, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[77] David McClosky, Mihai Surdeanu, and Christopher D Manning. Event extraction as dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (HLT-ACL)*, pages 1626–1635, 2011.

[78] Renée J. Miller, Laura M. Haas, and Mauricio A. Hernández. Schema mapping as query discovery. In *VLDB*, pages 77–88, 2000.

[79] Renee J. Miller, Mauricio A. Hernandez, Laura M. Haas, Lingling Yan, C. T. Howard, Ronald Fagin, Ho Ronald, and Lucian Popa. The clio project: Managing heterogeneity, 2001.

[80] Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL-2009)*, pages 1003–1011, 2009.

[81] Raymond J Mooney and Razvan C Bunescu. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178, 2005.

[82] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[83] Ndapandula Nakashole, Martin Theobald, and Gerhard Weikum. Scalable knowledge harvesting with high precision and high recall. In *Proceedings of the fourth ACM international conference on Web search and data mining (WSDM)*, pages 227–236, 2011.

[84] George L Nemhauser and Laurence A Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1988.

[85] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. Factorizing yago: scalable machine learning for linked data. In *Proceedings of the 21st international conference on World Wide Web*, pages 271–280. ACM, 2012.

[86] Mathias Niepert, Christian Meilicke, and Heiner Stuckenschmidt. A probabilistic-logical framework for ontology matching. In *AAAI*, 2010.

[87] Bo Pang, Kevin Knight, and Daniel Marcu. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *NAACL*, pages 102–109. Association for Computational Linguistics, 2003.

[88] Maria Pershina, Bonan Min, Wei Xu, and Ralph Grishman. Infusion of labeled data into distant supervision for relation extraction. In *Proceedings of ACL*, 2014.

[89] Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–10, 2009.

[90] Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *EMNLP*, pages 1–10. Association for Computational Linguistics, 2009.

[91] Hoifung Poon and Pedro Domingos. Unsupervised ontology induction from text. In *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*, pages 296–305. Association for Computational Linguistics, 2010.

[92] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB JOURNAL*, 10:2001, 2001.

[93] Altaf Rahman and Vincent Ng. Supervised models for coreference resolution. In *EMNLP*, pages 968–977. Association for Computational Linguistics, 2009.

[94] Marta Recasens, Matthew Can, and Dan Jurafsky. Same referent, different words: Unsupervised mining of opaque coreferent mentions. In *Proceedings of NAACL-HLT*, pages 897–906, 2013.

[95] Roi Reichart and Regina Barzilay. Multi event extraction guided by global constraints. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL)*, pages 70–79, 2012.

[96] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 2006.

[97] Sebastian Riedel, David McClosky, Mihai Surdeanu, Andrew McCallum, and Christopher D Manning. Model combination for event extraction in BioNLP 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 51–55, 2011.

[98] Sebastian Riedel, Limin Yao, Benjamin M. Marlin, and Andrew McCallum. Relation extraction with matrix factorization and universal schemas. In *Joint Human Language Technology Conference/Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2013.

[99] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, pages 148–163, 2010.

[100] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Proceedings of the Sixteenth European Conference on Machine Learning (ECML-2010)*, pages 148–163, 2010.

[101] Alan Ritter, Oren Etzioni, Sam Clark, et al. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 1104–1112, 2012.

[102] D. Roth and W. Yih. Global inference for entity and relation identification via a linear programming formulation. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[103] Evan Sandhaus. *The New York Times annotated corpus*. Linguistic Data Consortium, 2008.

[104] Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, et al. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 523–534. Association for Computational Linguistics, 2012.

[105] Satoshi Sekine. On-demand information extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 731–738. Association for Computational Linguistics, 2006.

[106] Siwei Shen, Dragomir R Radev, Agam Patel, and Güneş Erkan. Adding syntax to dynamic programming for aligning comparable texts for the generation of paraphrases. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 747–754. Association for Computational Linguistics, 2006.

[107] Y. Shinyama and S. Sekine. Preemptive information extraction using unrestricted relation discovery. In *HLT-NAACL*, 2006.

[108] Yusuke Shinyama and Satoshi Sekine. Paraphrase acquisition for information extraction. In *Proceedings of the second international workshop on Paraphrasing-Volume 16*, pages 65–71. Association for Computational Linguistics, 2003.

[109] Yusuke Shinyama and Satoshi Sekine. Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, pages 304–311, 2006.

[110] Noah A. Smith and Jason Eisner. Contrastive estimation: Training log-linear models on unlabeled data. In *ACL*, 2005.

[111] Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics, 2008.

[112] Richard Socher, Eric H Huang, Jeffrey Pennington, Andrew Y Ng, and Christopher D Manning. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *NIPS*, 24:801–809, 2011.

[113] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1314–21, 1995.

[114] Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S Weld. Open information extraction to kbp relations in 3 hours.

[115] Mark Stevenson and Mark A. Greenwood. A semantic approach to ie pattern induction. In *ACL*, 2005.

[116] Fabian M. Suchanek, Mauro Sozio, and Gerhard Weikum. Sofie: A self-organizing framework for information extraction. In *Proceedings of the International Conference on World Wide Web (WWW)*, 2009.

[117] Fabian M Suchanek, Mauro Sozio, and Gerhard Weikum. Sofie: a self-organizing framework for information extraction. In *Proceedings of the 18th international conference on World wide web*, pages 631–640. ACM, 2009.

[118] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2012.

[119] Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP)*, pages 455–465, 2012.

[120] Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Probabilistic matrix factorization leveraging contexts for unsupervised relation extraction. In *Advances in Knowledge Discovery and Data Mining*, pages 87–99. 2011.

[121] Jeffrey D. Ullman. Information integration using logical views. In *ICDT*, pages 19–40, 1997.

[122] Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R Movellan, and Paul L Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems*, pages 2035–2043, 2009.

[123] Michael L. Wick, Khashayar Rohanimanesh, Karl Schultz, and Andrew McCallum. In *KDD*, pages 722–730, 2008.

[124] F. Wu, R. Hoffmann, and D. Weld. Information extraction from Wikipedia: Moving down the long tail. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-08)*, pages 731–739, New York, NY, USA, 2008. ACM.

[125] F. Wu and D. Weld. Autonomously semantifying Wikipedia. In *Proceedings of the ACM Sixteenth Conference on Information and Knowledge Management (CIKM-07)*, Lisbon, Porgugal, 2007.

[126] Fei Wu and Daniel S. Weld. Autonomously semantifying wikipedia. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, pages 41–50, 2007.

[127] Fei Wu and Daniel S. Weld. Open information extraction using wikipedia. In *The Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 118–127, 2010.

[128] Sen Wu, Ce Zhang, Feiran Wang, and Christopher Ré. Incremental knowledge base construction using deepdive. *arXiv preprint arXiv:1502.00731*, 2015.

[129] Huahai Yang, Daina Pupons-Wickham, Laura Chiticariu, Yunyao Li, Benjamin Nguyen, and Arnaldo Carreno-Fuentes. I can do text analytics!: designing development tools for novice developers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1599–1608. ACM, 2013.

[130] Mihalis Yannakakis. On the approximation of maximum satisfiability. In *Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, SODA '92, pages 1–9, 1992.

[131] L. Yao, S. Riedel, and A. McCallum. Collective cross-document relation extraction without labelled data. In *Procs. of EMNLP*, 2010.

[132] Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In *EMNLP*, pages 1456–1466. Association for Computational Linguistics, 2011.

[133] Limin Yao, Sebastian Riedel, and Andrew McCallum. Unsupervised relation discovery with sense disambiguation. In *ACL*, pages 712–720. Association for Computational Linguistics, 2012.

[134] Alexander Yates and Oren Etzioni. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, 34(1):255, 2009.

[135] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *The Journal of Machine Learning Research*, 3:1083–1106, 2003.

[136] Luke Zettlemoyer and Michael Collins. Online learning of relaxed CCG grammars for parsing to logical form. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP/CoNLL)*, 2007.

[137] Congle Zhang, Raphael Hoffmann, and Daniel S Weld. Ontological smoothing for relation extraction with minimal supervision. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2012.

[138] Congle Zhang, Stephen Soderland, and Daniel S Weld. Exploiting parallel news streams for unsupervised event extraction. *Transactions of the Association for Computational Linguistics*, 3:117–129, 2015.

[139] Congle Zhang and Daniel S Weld. Harvesting parallel news streams to generate paraphrases of event relations. In *Proceedings of the 2013 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP)*, pages 455–465, 2013.

[140] 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.

Appendix A

RESOURCES FOR DISTRIBUTION

The dataset for VELVET is available at the following url: `http://aiweb.cs.washington.edu/ai/clzhang/velvet.tgz`. It includes

- a dump of Freebase.
- the gold mapping from the target relations (NELL) to Freebase.
- the ontology mapping generated by VELVET.

The parallel news stream corpus from March 1st, 2013 to April 1st, 2015 is available in `http://aiweb.cs.washington.edu/ai/clzhang/nsre2/sentences.tokens.gz` and `http://aiweb.cs.washington.edu/ai/clzhang/nsre2/sentences.articleIDs.gz`

The dataset for NEWSSPIKE-PARA is available at the following url: `http://aiweb.cs.washington.edu/ai/clzhang/paraphrase.tgz`. It includes

- the gold paraphrase clusters to learn the model.
- the generated paraphrases.

The code and dataset for NEWSSPIKE-RE and NEWSSPIKE-SCALE are available at the following url: `https://github.com/zhangcongle/NewsSpikeRe`. It includes

- a crawler to collect the parallel news streams.
- an NLP pipeline that integrates Stanford parser, fine grained NER and open IE.
- the event extraction algorithms.
- the generated training sentences and the learned event extraction models.