Report Linking:

Information Extraction for Building Topical Knowledge Bases

by

Travis Wolfe

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

August, 2017

Abstract

Human language artifacts represent a plentiful source of rich, unstructured information created by reporters, scientists, and analysts. In this thesis we provide approaches for adding structure: extracting and linking entities, events, and relationships from a collection of documents about a common topic. We pursue this linking at two levels of abstraction. At the document level we propose models for aligning the entities and events described in coherent and related discourses: these models are useful for deduplicating repeated claims, finding implicit arguments to events, and measuring semantic overlap between documents. Then at a higher level of abstraction, we construct knowledge graphs containing salient entities and relations linked to supporting documents: these graphs can be augmented with facts and summaries to give users a structured understanding of the information in a large collection.

Thesis Committee: († advisors)

Mark Dredze† (Associate Professor, Computer Science, Johns Hopkins University)

Benjamin Van Durme† (Assistant Professor, Computer Science, Johns Hopkins University)

Philipp Koehn (Professor, Computer Science, Johns Hopkins University)

Acknowledgments

Throughout my tenure as a PhD student at Johns Hopkins, I have had a lot of help from those around me. My advisors have aided me through many late-night paper-finishing sessions, extended experiment-planning meetings, and practice talks. I've drawn inspiration and learned a lot from all the professors, researchers, and students around me.

First I'd like to thank Marius Paşca for making me a better researcher by teaching me about picking a topic and being "ruthless" about finishing it. I'd like to thank Mark Dredze for being my first academic advisor and helping with my research no matter where it went. I've learned more about intelligibly structuring my thoughts from him than anyone else. I'd like to thank Benjamin Van Durme for not only being a great advisor, but also for broadening my academic horizons to include linguistics and cognitive science and reminding me that NLP is a sub-field of artificial intelligence rather than a series of ever-changing engineering experiments. I'd also like to thank Ken Church for encouraging me to come to Johns Hopkins and offering up some interesting perspectives on the field of NLP. Jason Eisner also had an impact on me, showing that creativity, intellectual levity, and dedication to big ideas are all important parts of being a great researcher.

I want to also thank Benjamin and Mark for being very personable advisors and running groups which were helpful in making Johns Hopkins feel like a welcoming and comfortable place to do research.

I would also like to thank the following people, who have been helpful in making me the researcher that I am today though numerous stimulating conversations: Nicholas Andrews, Charley Bellar, Chris Callison-Burch, Tongfei Chen, Ken Church, Ryan Cotterell, Jason Eisner, Frank Ferraro, Juri Ganitkevitch, Matt Gormley, Craig Harman, Rebecca Knowles, Keith Levin, Tom Lippincott, Chandler May, James Mayfield, Paul McNamee, Meg Mitchell, Michael Paul, Ellie Pavlick, Violet Peng, Matt Post, Pushpendre Rastogi, Kyle Rawlins, Rachel Rudinger, Keisuke Sakaguchi, Lenhart Schubert, Ves Stoyanov, Max Thomas, Ehsan Variani, Tim Vieira, Svitlana Volkova, and Xuchen Yao.

Dedication

This thesis is dedicated to my father, William E. Wolfe.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

For many professionals today, the ability to do their job is tied up in the ability to store, organize, and retrieve information. This information can be used to make important business decisions and find key people and organizations in a new area. Right now these information-based tasks are done by people, knowledge workers, who are trained experts and in demand. Methods for helping these people perform their jobs more efficiently and at larger scale than is possible today is a key challenge for modern artificial intelligence research.

Some have framed this problem as "information overload" (Maes, 1994). The problem of knowledge workers being faced with a deluge of information (e.g. emails, reports, tables) which they must spend their attention on understanding before getting to the job of weighing evidence and making complex decisions. One way to view this problem is one of filtering and recommendation: either the task of showing only relevant materials to a knowledge worker or the task of routing information to the knowledge worker who is most apt to consume it. Another way to view the problem, which we pursue in this thesis, is as search and exploration: how can knowledge workers most directly find the information relevant to the decisions they have to make?

Search engines are one of the most popular tools for finding information today. These technologies have been honed to work very well when there is an information need which can be clearly expressed via a short query. Search engines today are very good at matching single queries with single snippets of information, either in the form of a short answer for factoid QA (Ferrucci et al., 2010), a snippet of text from a page (Callan, 1994), or an entire webpage listed in the results. Search engines are even more powerful when they can exploit supervision relating queries to satisfactory results (e.g. click throughs) and when they have access to high quality content which actively adapts to the needs of users (driven through competition in the attention economy (Davenport and Beck, 2001)).

All of this depends on a knowledge workers' ability to formulate their information need as a short query. This is not always possible in cases where there is a lot of new information to take in and organize, when the relevant keywords and important questions to ask have not been recognized yet. In cases like this, methods for exploring the data are more beneficial than query-based search methods. Exploration requires some system of organizing the information so that a user is not forced to simply explore by enumerating documents, which may waste a lot of time. This thesis is concerned with creating better schemes for organizing information in text.

Sensemaking (Russell et al., 1993), as studied in information retrieval and human-computer interaction, is the process of building representations of data for answering task-specific questions. These representations often span a range of levels of abstraction and finding good ones is often domain-specific and difficult to formalize (Pirolli and Card, 2005). This work, understood as a step in sensemaking, provides a automatically and quickly generated low level representation which cuts down on the cost of information foraging (Pirolli and Card, 1999).

1.1.1 Knowledge Workers

Up to this point we have been not been specific about the types of knowledge workers that we are interested in helping, and what their typical information needs are. For this work, knowledge workers are defined as anyone who regularly uses textual information to make decisions as a part of their job.[1] Knowledge workers don't have to be specialists according to our definition, but they often are in practice. Some examples include:

1. *financial analysts* who study a particular area of business in order to make recommendations on what investments or decisions should be made. They are interested in statements made by companies and high level employees, announcements of mergers and acquisitions, lawsuits, regulatory changes, and related news.

2. *scientists* who study the causal mechanisms governing a domain like the growth of plants, the efficiency of an economy, or the regulation of proteins by genes. They read papers which discuss experiments and observations which have implications for the theorized relationships between entities in the domain.

[1] This is an ad-hoc definition. We are interested in those who use *textual* information as a means of limiting scope rather than as an essential aspect of knowledge workers in general.

3. *lawyers* who need to read documents explaining the contacts between, actions of, and agreements between parties in a legal transaction. This material may be collected from police reports, paralegal reports, financial reports, the news, or other sources.

Knowledge workers often have domain knowledge about what evidence constitutes a pattern they're looking for. This evidence can come in the form of types of events (e.g. situations where someone is arrested), or roles entities played in a given event (e.g. one company buying another), or simply the existence of any relationship between two entities (e.g. the presence of a particular type of protein in a diseased organism).

Knowledge workers often have to de-duplicate or synthesize evidence from many different sources. In order to find as much relevant information as possible, knowledge workers will often have to read materials which discuss facts which are already known or stated elsewhere. Finding the subset of claims which are novel or surprising is an important task for knowledge workers (Pirolli and Card, 1999).

Finally, knowledge workers often produce *reports* as a product of their analysis. These reports can explain a particular phenomenon or event within the domain (e.g. an arrest report or a scientific paper) and may be used as a source for other knowledge workers with related jobs. Organizations who employ knowledge workers may produce large collections of reports which hold and transmit information from one knowledge worker to another and have great value to the organization.

1.1.2 Reports

Reports are a written form meant to communicate information between knowledge workers. Examples of reports include academic papers, crime/incident reports, financial reports, and news articles. For this work, we focus on reports which are expressed with natural language, though they may take other forms including tables and diagrams.

Reports discuss entities and events relevant to the author's (knowledge worker's) domain of interest, and sometimes use specialized language to do so. For example, the entities themselves may have names which are particular to a set of knowledge workers, which are not known to the general population of host language speakers, and may be opaque to outsiders (e.g. someone may have no idea what "Galactose-alpha-1,3-galactose" is but be able to identify it as an entity and parse sentences containing it). The same is true of how events are described in reports, which may use a specialized lexicon which is not widely used. Together, we can refer to this language as *jargon*. In this work we are concerned with creating automatic tools for processing reports which may contain jargon, and an important assumption is that this jargon does not make the language indecipherable to general purpose natural language tools like parsers, taggers, and segmenters.

1.2 Report Linking

The goal of this thesis is to develop methods for organizing information into structured graphs which we collectively refer to as *report linking*. The goal of report linking is to link together relevant pieces of information in a collection of reports. This link structure constitutes a set of abstract views based on various ways of automatically organizing information in reports which can help knowledge workers explore and find novel information quickly.

We refer to the structure induced by report linking tools collectively as a *topic knowledge base* (TKB). A TKB is a graph where the nodes are either entities or reports and the edges (or links) indicate some relationship between the two nodes connected. The TKB offers a way for knowledge workers to explore the information contained in reports without paying the high price of reading all of the reports' text.

In this work entity nodes represent people, places, and organizations discussed in the reports. We chose these entity types, and not others such as websites, phone numbers, consumer products, or weapons because they are important to a wide variety of knowledge workers and because relatively robust tools exist which we can build upon.

As we will discuss in greater detail in the rest of this thesis, we offer two *views* of entity nodes in a TKB. The first view shows all of the sentences which mention an entity which provides a high-recall method for finding entity-centric information across reports. The second view is comprised of a short natural language summary of all of the information in the source reports. This view is meant to be informative but brief, allowing a user to view only the most important information reported about an entity without any duplication.

Reports, the other node type in TKBs, have a view which displays the text of the report itself, but with entity and event mentions rendered at hypertext, linking either back to an entity node or to other adjacent reports. This hypertext is used to link individual entities and events discussed within a report to either other reports or entity nodes.

There are three types of edges in a TKB, entity-to-entity, entity-to-report, and report-to-report. Each of these edges may have various views which implement a form of analysis which seeks to explain how the two endpoints are related. Knowledge workers use these edge views to guide their exploration of the TKB graph.

1.3 Outline

The rest of this thesis goes into detail on the steps required for building a TKB, which is comprised of many tasks. In Chapter 2 we discuss background material, covering the most prominent methods and conceptions of how information should be extracted from text and organized. This chapter covers four major themes in extracting information from text and the rest of the thesis makes contributions in each category.

In Chapter 3 we discuss the first steps of construction of a topic knowledge base: identifying the entities and the pairwise relationships between them. Our contributions include new methods for efficient entity mention search, a new method for jointly disambiguating pairs of entities, and a method for inferring how related two entities are from text. Our experiments verify that the proposed methods for the first step of constructing a TKB are very high precision. Work in this chapter was also described in Wolfe et al. (2016a).

In Chapter 4 we address the problem of putting informative labels on the entity-to-entity edges. The contributions in this category fall into two categories. The first is an unsupervised method for inferring trigger words which characterize the relationship between two entities (§4.2). This method does not depend on a relational schema or training data, so it is appropriate for a wide variety of different entities and relationships. The second category of contributions are on distant supervision for relation extraction, described in

§4.3. This work proposes a novel objective for learning from distant supervision which includes a measure of entity type diversity and makes weak mention-level assumptions. Additionally this work proposes a novel syntactically-informed method for building high-precision extractors. The work in §4.2 appeared in Wolfe et al. (2016a) and the work in §4.3 in (Wolfe et al., 2017).

Chapter 5 focuses on event-centric methods for extracting information from text. These events will be used in creating structured report-to-report edges/links. Our contributions include a transition-based model with global features for frame semantic parsing as well as a detailed analysis of methods for training greedy global models with imitation learning. The work in this chapter were previously published in Wolfe et al. (2016b).

Given the answers of *what* to link provided in chapters 3 (entities) and chapter 5 (events), chapter 6 explains *how* to link these items in structured report-to-report links. The contributions in that chapter include two models for linking, one is a feature based model which makes use of a wide range of semantic resources (Wolfe et al., 2013) and another which uses structured inference to jointly predict links events and their arguments (Wolfe et al., 2015). Both models were state of the art at the linking task at the time, and the second still is.

In Chapter 7 we introduce entity summarization: the task of producing informative summaries of entities from their descriptions in a large corpus (similar to the first paragraph of a Wikipedia page). Our contributions also include an entity summarization model which can jointly perform relation extraction and summarization which outperforms a strong baseline on this new task (Wolfe et al., 2017).

In Chapter 8 we conclude this thesis with a discussion of the applicability of the methods described in this thesis and of future work on report linking.

Chapter 2

Extracting and Organizing Information from Text

2.1 Introduction

In this thesis we are concerned with designing tools for knowledge workers which can organize and provide access to a wide variety of information which can be inferred from text. This is a very broad goal, and there likely won't be one approach which will accomplish all of our goals. In this section we survey some of the methods which accomplish related goals in a variety of ways. Much of the work which we will discuss takes a narrow view of their particular task, but in this section we aim to organize these efforts into a coherent view of the topic.

Much of the way that these topics have been traditionally organized has to do with the field of study from which an interest in a topic originally came. Over time, the interests of information retrieval, natural language processing, and linguistics have converged towards rich methods for searching and representing knowledge gleaned from natural language. We've chosen to organize the background material in this thesis into four categories. They are not purely mutually exclusive nor purely orthogonal, but they are prominent themes which provide a good basis for describing research in this area.

The four categories of work which we will discuss are event-centric, knowledge base-centric, corpus-centric, and report-centric methods. These categories can be grouped into two groups. The first, event-centric and KB-centric methods, are concerned with *abstractions* over natural language. Both propose a *latent*, *canonical*, and often *symbolic* form for representing information. Their power generally lies in their explicit use of disambiguation to avoid confusion arising from shallow readings of natural language.

The second group, corpus-centric and report-centric methods, are concerned with *transformations* of text. Both propose storing information in the form of *natural language*, and propose a variety of ad-hoc text-to-text transformations to solve problems like indexing/search and comparison (e.g. coreference). These methods include search engines and extractive summarization tools, and in general tend to be task-driven rather than theory-driven. These approaches often use surface features and machine learning over theories of parsing, inference, and latent forms in order to implement these text-to-text transformations.

2.2 Event-centric

The first category of methods in this chapter are event-centric methods. These methods focus on abstracting away from the text by inferring the *events* described in text. Events are a natural concept (people talk about events without being told what one is) and show up frequently in news articles, textbooks, and other repositories of human knowledge. While it is not difficult to see hints of how language maps onto events through verbs ("John *bought* a candy bar"), nouns ("Kennedy's *death* saddened the nation), and other shallow linguistic cues, coming up with a general theory which explains what events are, how to recognize them from text, and how to reason about their antecedents and consequences is a very different matter. Methods in this category all offer some definition of what an event is and how to recognize them. The methods are laid out roughly chronologically and reflect a changing focus from theory-driven to task-driven approaches to understanding events in natural language.

2.2.1 Scripts and Frames

One of the first conceptions of how natural language understanding should work is a branch of artificial intelligence called story understanding. This line of work included Minsky (1974), Schank (1975), Fillmore (1976), Schank and Abelson (1977), Charniak (1977), Wilensky (1978), and Norvig (1983).[1] They used simple examples childrens' stories and descriptions of common situations like getting dinner at a restaurant as motivating examples of their theories. They observed that there was a lot of meaning contained in

[1] See Ferraro and Van Durme (2016) for an explanation of the relationship between various conceptions of frames.

short stories which was not directly expressed in the text. They were concerned with any form of meaning for which a human could confidently infer from reading the story, but could not be linked to particular textual proposition. A lot of the meaning they observed as missing from the text, but not the human understanding they were trying to mimic, had to do with the intentions of agents in the story and events which implicitly occurred.

Artificial intelligence at the time often viewed problems through the framework of search and planning, so initial attempts to recover this meaning involved logical inference over basic propositions observed in the story (e.g. "Jane broke open her piggy bank") and postulates in a pre-programmed knowledge base (e.g. "piggy banks contain money"). These postulates and inference were intended to allow the system to recover this missing meaning, but it quickly became clear that this inference process was under-specified and very computationally intensive (McDonald, 1978). The proposed solution to this were abstractions which went by a variety of names including *frames*, *scripts*, and *plans*.

The common intuition amongst these approaches is a template which has many slots which can be filled by entities observed in the story. The meaning of these slots are relative to the template (frame or script) they belong to, but can represent things like the Killer in a Murder template. The script or frame itself was a template in the sense that it instantiated actual scenarios with slot values which were expressed in a story.

The benefits of frames and scripts fall into two categories: semantic and computational. The semantic benefits have to do with the ability to recognize that there are slots with missing values. Some argued for default values for these slots (Minsky, 1974), while others argued for more complex resolution schemes (Bobrow and Winograd, 1977; Brachman and Schmolze, 1985) i.a., but either way frames and scripts pushed ambiguity from the "unknown unknowns" to "known unknowns" category, which was conceptual progress.

There are computational benefits arising from frames and scripts in that an understanding algorithm can build up large propositions (the frames themselves and all slot bindings) directly by instantiating a frame rather than having a set of general and highly productive rules which reach the same propositions. Put another way, frames allow for a degree of specificity and sparsity in the inference rules in the knowledge base which would not be otherwise possible, and this has nice computational implications.

Though this work on frames, scripts, and plans was capable of describing a wide range of story understanding phenomena, the work ignored a lot of the complexity in building *robust* systems for understanding stories. For one, it was common practice to publish papers describing frame and script processing engines before the authors had implemented them. When they did implement them, they had to make strict assumptions about their input, like the fact that they only need to work on a small number of short stories. The implementation details were not the focus of their published work, which left other researchers, even those in the field, confused about how to re-implement their ideas (McDonald, 1978). These authors systematically ignored some of the more language-related (less knowledge-related) challenges in inference like identifying word senses, resolving syntactic ambiguity, and handling open vocabulary entities like people and organizations. Partly this was consciously ignored as an incremental strategy to make their systems work on the "hard stuff" (knowledge and inference) first on "easy cases" (domains with very limited vocabularies), leaving questions of text processing on "difficult cases" as an implementation detail to return to later. This sort of text processing like part of speech tagging, syntactic parsing, named entity recognition, and reference resolution later became the majority of the focus of research on extracting knowledge from text.

A modern conception of these some of these ideas is FrameNet (Baker et al., 1998). The knowledge base contains frames and slots (frame elements), but as the task has come to be studied, there is no inference[2] involved and no modeling of goals and intentions (other than recognizing spans of text which explicitly refer to them such as "She $[\text{broke}]_{\text{Cause\_to\_fragment}}$ the piggy bank $[\text{to get the coins out}]_{\text{Explanation}}$.") Annotations are sentential, meaning that the original goal of understanding two sentences in the story "Jane broke her piggy bank. She used the money to buy candy." cannot link the "breaking [a] piggy bank" to "money" or "to buy candy", unless the author is careful to put all of these phrases into one sentence. We discuss FrameNet in more depth in Chapter 5.

2.2.2 Narrative Chains

A more recent event-centric group of methods involve narrative chains (Chambers and Jurafsky, 2008, 2009, 2011; Cheung et al., 2013; Balasubramanian et al., 2013; Frermann et al., 2014; Rudinger et al., 2015; Alayrac et al., 2016). This work is inspired by the work on frames and scripts, but the emphasis changed from "frames and scripts are needed for story understanding" to "if we read a bunch of stories, then we should be able to statistically infer frames and scripts." This work primarily took the form of generative models of the text describing stories or data mining techniques for finding patterns in stories. These patterns

[2] Inference in the senses of knowledge-based (e.g. "Socrates is a man; men are mortal; Socrates is mortal") and linguistic (e.g. reference resolution by checking number, gender, and animacy properties) inference rather than *statistical* inference common in modern machine learning methods.

or generative templates were meant to stand in place of the frames, scripts, and plans. They would not be as richly typed and structured, but they would be learned from data, solving the knowledge acquisition bottleneck problem (Olson and Rueter, 1987). In a sense this is a bolder goal since statistical learning when the hypothesis class is as big as the space of frames or scripts requires a lot of data and inductive bias.

We begin to lose some control over the abstractions or forms here. Before, we had world knowledge which was expressed in the language that the AI programmer chose. This move into the statistical era meant that this knowledge took the form of distributions over observable features, which has a few problems. First, we can't easily assign these distributions *names*, which is essential in building up bigger pieces of knowledge. Second, because we have to fit these distributions, the number of them we can work with is not that large. In the symbolic era, you could create symbols for all sorts of things, and because you didn't need to do statistical estimation, the only cost you had to pay for adding more symbols and rules was computational, which were reasonable in that time and trivial today.

2.2.3 Information Extraction for Events

A different set of event-centric methods treat recognizing events as an engineering problem, which we call information extraction (IE) based methods. The notion of a general theory of events was dropped in favor of IE methods which could recognize a restricted set of event types such as terrorism, political conflicts, and natural disasters. With this type of restriction, small event schemas could be manually created instead of learned. Additionally these schemas did not attempt to be deep in the sense of some earlier work on scripts and frames, their only role was to characterize a handful of slot types which had a reasonable correlation with lexical choice.

This shift towards IE over deeper frame-based methods for natural language understanding started with the Message Understanding Conferences (MUC) (Sundheim, 1996). Midway through the MUC conferences, the organizers commented on the common wisdom up to that time regarding natural language understanding:

> These challenges have also resulted in a critical rethinking of assumptions concerning the ideal system to submit for evaluation. Is it a “generic” natural language system with in-depth analysis capabilities and a well defined internal representation language designed to accommodate the translation of various kinds of textual input into various kinds of output? Or is it one that uses only shallow processing techniques and does not presume to be suitable for language processing tasks other than information extraction?
>
> (Sundheim and Chinchor (1993))

In the final version of the MUC tasks, systems had to complete four tasks: Named Entity (NE) Coreference (CO) Template Element (TE) Scenario Template (ST). The first three have to do with recognizing entities and are not event-centric, but the final task of scenario template extraction is about filling slots for three types of scenarios of interest: “aircraft order”, “labor negotiations”, “management succession”. For ST, the organizers manually constructed hierarchy of templates which were to be filled by IE systems. The restriction of the ST task reflects both the motivation for IE-based event-centric methods and their weakness: good results in recognizing events are possible when there is only a small number of types of events to model (Grishman and Sundheim, 1996).

The Automatic Context Extraction (ACE) (Doddington et al., 2004) program was the source of more IE-based work. They continued with the goals of MUC and greatly expanded the annotation efforts to aide in both training of machine learning IE models as well as evaluation. ACE annotated entities,[3] events,[4] and relations,[5] the latter two being most relevant to this section. The types of events was broadened from 3 scenarios in MUC to 33 event types (across 8 coarse grain event types). Further details on differences between the MUC, ACE, and other IE-based event representations can be found in Aguilar et al. (2014). The annotation was extensive, covering more than 300k words.

FrameNet (Baker et al., 1998) and Propbank (Kingsbury and Palmer, 2002; Palmer et al., 2005) are two semantic role labeling (SRL) (Gildea and Jurafsky, 2002) annotation projects which also fall into the category of IE-based event-centric work. These projects focus on annotating a wide range of frames and their roles to train statistical models to recognize them. The goals (richness of the frames and inference related to recognizing them) of this work was more humble than the original work on frames, but what they lacked in aspirations they made up for in annotations. These two datasets lead to an enormous amount of work on statistical systems for recognizing events (Punyakanok et al., 2004; Xue and Palmer, 2004; Carreras and Màrquez, 2005; Haghighi et al., 2005; Johansson and Nugues, 2008b; Toutanova et al., 2008; Surdeanu et al., 2008; Hajič et al., 2009; Björkelund et al., 2009; Das et al., 2010; Pradhan et al., 2013; Täckström et al., 2015). More information on the resources can be found later in §5.2.

2.2.4 Connection to this Thesis

In this thesis we make contributions to the information-extraction body of work on event-centric methods. In Chapter 5 we adopt the IE view of recognizing events and

[3] Included from inception in 2000

[4] Started in 2005

[5] Started in phase 2 in 2002

propose statistical methods for identifying events and their participants based on FrameNet (Baker et al., 1998) and Propbank (Palmer et al., 2005). In Chapter 6 we make IE-based contributions on event coreference by jointly modeling linking of entities and events.

2.3 Knowledge Base centric

Another important line of work in storing information gleaned from text are centered around knowledge bases (KBs). In general, a knowledge base is a set of concepts with relationships between them. In this work we focus on KBs which are similar to those defined in NIST's Text Analysis Conference's Knowledge Base Population track (McNamee and Dang, 2009). These concepts are typically take to be either classes (e.g. birds, animals, living things) or entities (e.g. George Washington, Statue of Liberty). Knowledge bases are often conceived of as graphs with concepts as nodes and relations as edges. Relations can have multiple types like isa which encodes subset relationships between concepts (e.g. a bird isa animal) or instance (e.g. George Washington instance living things). The contents (concepts and relations) in KBs varies, but in general they are designed to be a symbolic means of storing useful information. KBs typically do not contain events, at least of the type described earlier,[6] but they do contain knowledge for understanding events, such as information about entities which can help in understanding an event. For example a KB may contain a concept for George Washington and the Delaware river, so recognizing the event described in "George Washington crossed the Delaware river" can be understood as an event involving a person (George Washington instance person) and a place (the Delaware

[6] Some other notions of KBs which we do not study here contain historical events like the `https://en.wikipedia.org/wiki/American_Civil_War`, but do not aim to store most events described in news or stories.

river instance location). Basic inference can also result from applying facts in the KB. For example the a KB might also contain a relation between the Delaware river and New Jersey (near), which would let you understand more about the "crossing" event (that it happened near New Jersey).

Some argue that one cannot understand language without some representation of the knowledge that humans have when they understand language (Hobbs, 1987). KBs offer a plausible model of storing and working with knowledge towards this goal. There are two major problems related to knowledge bases for this work: populating them with concepts and relations and recognizing references to their contents in language. We will discuss work on these two problems next and then return to other applications of KBs in natural language understanding.

2.3.1 Semantic Web and Public Knowledge Bases

Towards the first goal, of building knowledge bases which contain lots of useful information, many have taken the position that we simply need to write down all the facts and collect them into a knowledge base. This was initially motivated by AI programmers who saw all of the value in having common sense knowledge about the world and thought it would not take that long to formalize it all and put it in one place. After all, a human could more or less do it in 18 years. Surely if you just payed lots of people to write down everything they had learned you could create a knowledge base with an average human's worth of knowledge in a short period of time. Cyc (Lenat et al., 1986) was a knowledge base which took this approach towards manually and richly encoding common sense information. The project consumed a lot of resources and is not used by many today in light of its complexity.

The project has been criticized as a costly and over-ambitious mistake (Domingos, 2015).

A more modern approach to knowledge engineering is based on two ideas: use a simpler set of concepts and relations (eschewing inference altogether) and reliance on a large number of interested parties to help with populating it. These ideas are implemented in technologies which go by the name “semantic web” (Aberer et al., 2003; Halevy et al., 2003; Kementsietsidis et al., 2003). They propose methods for linking together information aggregated in many locations for many different purposes. The ability to link makes the KB a distributed system which can scale up to handle as much information as necessary. These methods also address issues of handling inconsistency and data provenance.

Another set of knowledge bases are derived from Wikipedia. These include Freebase (Bollacker et al., 2008), yago (Suchanek et al., 2007), yago2 (Hoffart et al., 2013), and DBpedia (Auer et al., 2007). These databases primarily draw on facts manually entered by Wikipedia contributors into infoboxes. There is a large amount of work needed to normalize, merge, and link information together to form these relatively clean KBs which provide information in the form of triples[7]. These KBs sometimes offer additional features like linking into specialize geographical databases for generalizing knowledge about places or the addition of temporal modifiers which can track things like facts which have start and end times (e.g. George W. Bush was president from January 20, 2001 to January 19, 2009).

WordNet (Miller, 1995) is a lexical knowledge base designed to store the relationships between words in English. They group words into synsets which all refer to a common concept. Synset concepts are related to each other via relationships like hyper-

[7] A triple is a tuple of a subject entity, a “verb” or relation, and object. Objects may be entities or some other type like a number (e.g. 1957) or string (e.g. “real estate agent”).

nymy, meronymy, and anytonymy. Wordnet can also map lexical concepts between parts of speech (e.g. “French” is the adjectival form of the nominal concept “France”).

2.3.2 Entity Linking

Given a knowledge base, an important task in understanding language is being able to link mentions in text to concepts in the KB. If these concepts correspond to entities (e.g. people, organizations, and locations), then this is called entity linking (Mihalcea and Csomai, 2007; Cucerzan, 2007; Milne and Witten, 2008; Kulkarni et al., 2009; Ratinov et al., 2011; Hoffart et al., 2011), i.a.

It has been argued by many that linking mentions of proper nouns to a knowledge base can help in tasks like coreference resolution because knowing the type of an entity or some facts about it can help understand what nominal phrases are licensed for referring to it (Haghighi and Klein, 2009; Recasens et al., 2013; Durrett, 2016), addressing one of the more difficult issues in coreference resolution.

There has been a lot of work on how to make entity linkers work well including methods which use name matching heuristics (aka lists, acronyms, transliteration) (McNamee et al., 2011), context matching heuristics (entity language models) (Han and Sun, 2012) and joint disambiguation which consider the named entity types and links together (Durrett, 2016), relations between the entities being linked (Cheng and Roth, 2013), and consider more than one linking decision together (Han et al., 2011).

These methods tend to work well when a mention’s entity is in the knowledge base (with accuracies as upwards of 86% (Han et al., 2011)), for a few of reasons. First, there is a lot of training data to support discriminative models which can effectively use surface features rather than deeper inference (Cucerzan, 2007; Ellis et al., 2015). Second, there are strong rich-get-richer effects when it comes to names: for any one name the baseline method of linking to the most popular entity with that name works well. Lastly, leaving popularity aside, the two biggest sources of signal, an entity's name and the distribution of words used to describe them, are largely independent, making the task easier when a lot of context is available. See Hachey et al. (2013) and Cornolti et al. (2013) for more detailed comparison of various entity linking techniques, variants of the task, and system performances.

Given how well these methods work when the KB contains the referent of a mention, more recent work has focused on how to *add* to a knowledge base. The TAC Knowledge Base Population (KBP) task has run since 2009 (McNamee and Dang, 2009). This task has involved three problems: entity linking, slot filling, and cold start KBP (since 2012). Slot filling is the first step in adding relations between entities in a KB. Systems are given an entity (e.g. George Washington) and a slot (e.g. where was this person born?), and must return a filler (e.g. Virginia) which may or may not appear in the KB itself. Cold start is defined as building a full KB containing entities and relations from just text. See McNamee et al. (2012) and Ellis et al. (2015) for more details on the cold start task.

2.3.3 Distant Supervision

Under certain circumstances, a knowledge base is a powerful source of *supervision* about how to understand natural language. The information extraction (IE) paradigm described in §2.2.3 is built on supervised machine learning methods which require labeled sentences. Annotating sentences is costly, and to a degree depending on the classification model used, the ability for an IE model to generalize the training data depends on the number of labeled sentences provided. Knowledge bases, plus the ability to recognize entities by linking them to a knowledge base, offer a different way of providing supervision: at the fact level rather than the sentence level.

Bunescu and Mooney (2007) observed that given a knowledge base containing facts like `almaMater(Christopher_Walken, Hofstra_University)`, one could reliably train a relation extractor for `almaMater` by looking through a large corpus for all sentences containing “Christopher Walken” and “Hofstra University” and *assuming* they were positive instances for `almaMater`. Negatives could be created in a variety of ways, or most straightforwardly using the closed world assumption.

The term for this method of training relation extractors from knowledge bases came to be known as “distant supervision” (Mintz et al., 2009). This method was good at finding a large number of true positive training examples on account of entity linkers being relatively high recall. But their weakness was the inclusion of false positives: sentences which match up with a fact but don’t commit to that fact *in situ* (e.g. “Hofstra asked Walken to give their commencement speech in 2006.” $\not\Rightarrow$ `almaMater(Christopher_Walken, Hofstra_University)`).

Hoffmann et al. (2011) and Surdeanu et al. (2012) explicitly formulate models which do not assume that every sentence which matches a fact implies that fact is true. These improvements lead to extractors which have much higher precision and recall. Bunescu and Mooney (2007) originally proposed using the multiple instance learning (MIL) framework (Maron and Lozano-Perez, 1998) which also makes weak assumptions, but the algorithms for fitting models like this were too slow at the time (Andrews et al., 2003).

Another line of work, stemming from Riedel et al. (2013), is *generative*: it maximizes the likelihood of observed facts and sentences under the assumption that there are latent features for entities, relations, and features of sentences. These methods also work well, but are not good at making predictions for entities which were not observed when the model was trained. The likelihood of a relation holding in a new sentence is a function of the latent features of the entities described in the sentence, which will be poorly fit if they have not been observed much (or at all).

Work on distant supervision, a method for training textual relation extractors, is related to but distinct from the task of of knowledge base completion (Sutskever et al., 2009; Jenatton et al., 2012; Bordes et al., 2013b; Socher et al., 2013). These approaches are distinct from the focus of this thesis because they are not interested in using text as evidence for knowledge, but rather evidence from either logical inference ($(man(x) \implies mortal(x) \vee socrates(x) \implies man(x)) \implies (socrates(x) \implies mortal(x))$) or statistical regularities (most kings' successor was born in the same country as them).

2.3.4 KB Applications

Previous work has shown that a significant fraction of factoid questions have answers present in publicly available knowledge bases, and question answering systems which explicitly model the relationship between questions and KB schemata can be very effective (Yao, 2014; Yih et al., 2015; Yao, 2015).

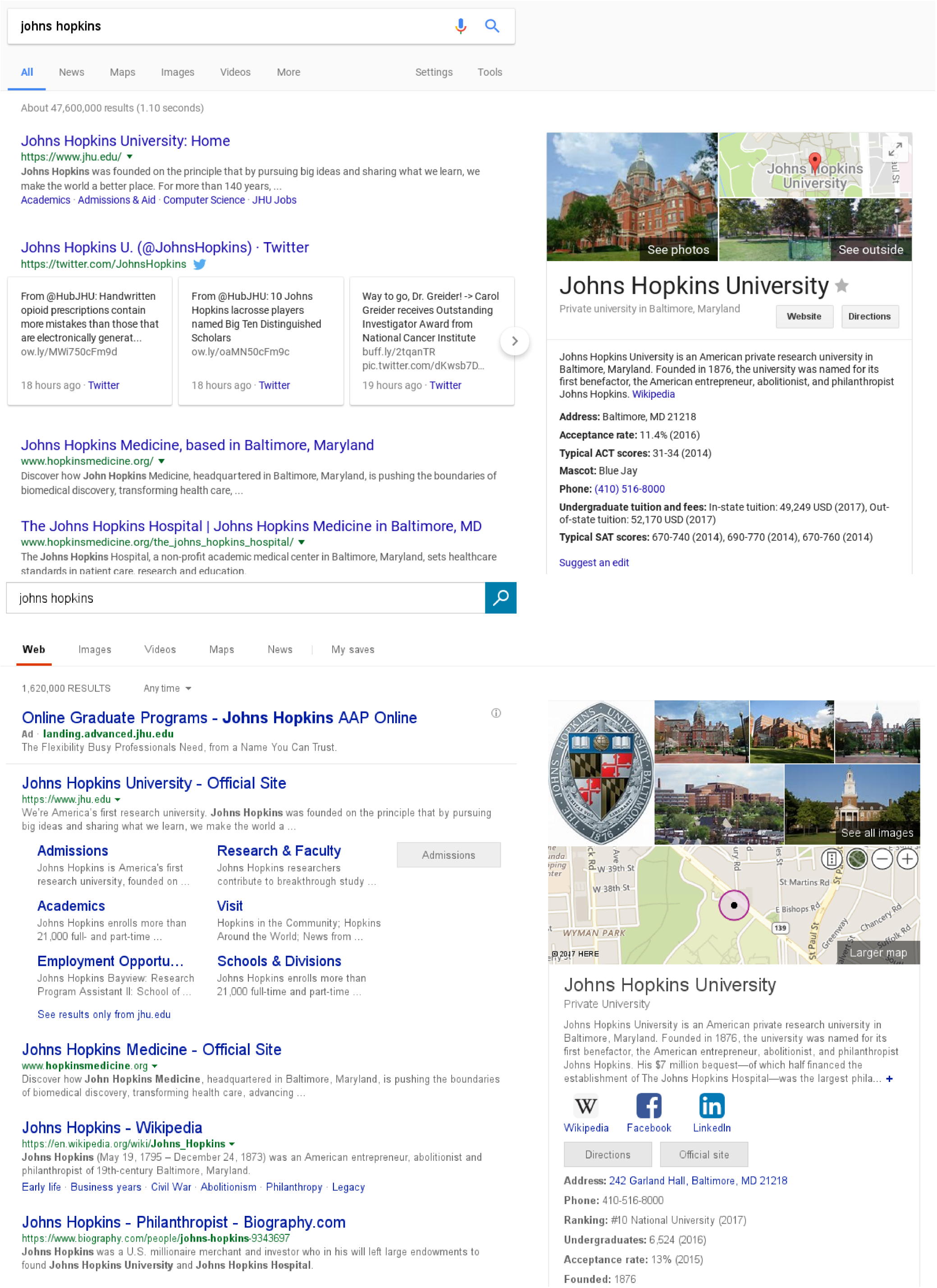
Knowledge bases have also been shown to provide search engine users a view of structured data which provides utility beyond what is provided by the document (e.g. web page) and passage (snippet) retrieval (Dalton and Dietz, 2013; Dietz and Schuhmacher, 2015). Popular search engines today use this type of structured results fetched from a knowledge bases as a part of many web queries which can be unambiguously linked to a node in their knowledge base, see Figure 2.1.

2.3.5 Connection to This Thesis

This thesis has three threads which are KB-centric. In Chapter 4 we present novel work on distant supervision for learning high precision relation extractors from pre-existing knowledge bases like DBpedia. The focus of this work is on learning domain-independent extractors which work well in domains which have lots of text data but no knowledge bases which can be used for inference or reasoning. Lack of a high-coverage knowledge base is the norm for most domains in the long tail which aren't concerned with celebrities, actors, musicians, and athletes which public knowledge bases have good coverage over.

The second KB-centric line in this thesis is the work on generating query-specific small-domain knowledge bases in Chapter 3 and 4. These so called "target KBs" differ from KBs like Freebase and DBpedia in that they are concerned with mapping out the connections between entities discussed in a domain-specific corpora (such as the Panama Papers or the Enron corpus (Klimt and Yang, 2004)). These KBs are concerned with *explaining* the connections between entities, and therefore use lexical relations rather than relations coming from a small schema like DBpedia or TAC KBP's (Ji et al., 2011) schemata.

The third KB-centric contribution of this thesis is the work on entity summarization described in Chapter 7. This work is motivated by the desire to be able to *browse* a knowledge base of entities in a way which supports a natural language based entity view for the fraction of knowledge workers who do not want to learn the semantics of a schema and would prefer textual to structured representations of entities. There is work on "sum-



About 47,600,000 results (1.10 seconds)

Johns Hopkins University: Home
https://www.jhu.edu/
Johns Hopkins was founded on the principle that by pursuing big ideas and sharing what we learn, we make the world a better place. For more than 140 years, ...
Academics · Admissions & Aid · Computer Science · JHU Jobs

Johns Hopkins U. (@JohnsHopkins) · Twitter
https://twitter.com/JohnsHopkins

From @HubJHU: Handwritten opioid prescriptions contain more mistakes than those that are electronically generat...
ow.ly/MWi750cFm9d
18 hours ago · Twitter

From @HubJHU: 10 Johns Hopkins lacrosse players named Big Ten Distinguished Scholars
ow.ly/oaMN50cFm9c
18 hours ago · Twitter

Way to go, Dr. Greider! -> Carol Greider receives Outstanding Investigator Award from National Cancer Institute
buff.ly/2tqanTR
pic.twitter.com/dKwsb7D...
19 hours ago · Twitter

Johns Hopkins Medicine, based in Baltimore, Maryland
www.hopkinsmedicine.org/
Discover how **John Hopkins** Medicine, headquartered in Baltimore, Maryland, is pushing the boundaries of biomedical discovery, transforming health care, ...

The Johns Hopkins Hospital | Johns Hopkins Medicine in Baltimore, MD
www.hopkinsmedicine.org/the_johns_hopkins_hospital/
The **Johns Hopkins** Hospital, a non-profit academic medical center in Baltimore, Maryland, sets healthcare standards in patient care, research and education.

Johns Hopkins University
Private university in Baltimore, Maryland

Johns Hopkins University is an American private research university in Baltimore, Maryland. Founded in 1876, the university was named for its first benefactor, the American entrepreneur, abolitionist, and philanthropist Johns Hopkins. Wikipedia

Address: Baltimore, MD 21218

Acceptance rate: 11.4% (2016)

Typical ACT scores: 31-34 (2014)

Mascot: Blue Jay

Phone: (410) 516-8000

Undergraduate tuition and fees: In-state tuition: 49,249 USD (2017), Out-of-state tuition: 52,170 USD (2017)

Typical SAT scores: 670-740 (2014), 690-770 (2014), 670-760 (2014)

Suggest an edit

1,620,000 RESULTS

Online Graduate Programs - **Johns Hopkins** AAP Online
Ad · **landing.advanced.jhu.edu**
The Flexibility Busy Professionals Need, from a Name You Can Trust.

Johns Hopkins University - Official Site
https://www.jhu.edu
We're America's first research university. **Johns Hopkins** was founded on the principle that by pursuing big ideas and sharing what we learn, we make the world a ...

Admissions
Johns Hopkins is America's first research university, founded on ...

Research & Faculty
Johns Hopkins researchers contribute to breakthrough study ...

Academics
Johns Hopkins enrolls more than 21,000 full- and part-time ...

Visit
Hopkins in the Community; Hopkins Around the World; News from ...

Employment Opportu...
Johns Hopkins Bayview: Research Program Assistant II: School of ...

Schools & Divisions
Johns Hopkins enrolls more than 21,000 full-time and part-time ...

See results only from jhu.edu

Johns Hopkins Medicine - Official Site
www.**hopkinsmedicine**.org
Discover how **John Hopkins Medicine**, headquartered in Baltimore, Maryland, is pushing the boundaries of biomedical discovery, transforming health care, advancing ...

Johns Hopkins - Wikipedia
https://en.wikipedia.org/wiki/**Johns_Hopkins**
Johns Hopkins (May 19, 1795 – December 24, 1873) was an American entrepreneur, abolitionist and philanthropist of 19th-century Baltimore, Maryland.
Early life · Business years · Civil War · Abolitionism · Philanthropy · Legacy

Johns Hopkins - Philanthropist - Biography.com
https://www.biography.com/people/**johns-hopkins**-9343697
Johns Hopkins was a U.S. millionaire merchant and investor who in his will left large endowments to found **Johns Hopkins University** and **Johns Hopkins Hospital**.

Johns Hopkins University
Private University

Johns Hopkins University is an American private research university in Baltimore, Maryland. Founded in 1876, the university was named for its first benefactor, the American entrepreneur, abolitionist, and philanthropist Johns Hopkins. His $7 million bequest—of which half financed the establishment of The Johns Hopkins Hospital—was the largest phila... +

Address: 242 Garland Hall, Baltimore, MD 21218

Phone: 410-516-8000

Ranking: #10 National University (2017)

Undergraduates: 6,524 (2016)

Acceptance rate: 13% (2015)

Founded: 1876

Figure 2.1: Google (above) and Bing (below) will display an infobox (right) with generated from their knowledge base when a query can reliably be linked to a KB node. Yahoo does as well, but shows Johns Hopkins the person, with a limited set of structured information.

marizing" a knowledge base nodes using structured outputs (sets of triples) (Cheng et al., 2011; Gunaratna et al., 2015; Thalhammer et al., 2012), but our work differs in that the output is a natural language description.

2.4 Corpus-centric

Corpus-centric methods organize a corpus as a graph of documents with edges connecting related documents. This sort of graph is useful for performing local exploration of a corpus rather than through search/retrieval involving a query. These edges are what make these methods useful to knowledge workers looking for information. They provide a semantically focused subset of the collection which may contain the information they need.

These methods are more suitable for machine creation as opposed to human creation (e.g. Wikipedia is corpus-centric, discussed later), owing to the scale of the problem and the effectiveness of automatic methods which work off of surface features. On the other hand, these methods may be more difficult for humans to use since they can require reading a lot of text if the link structure alone does not complete the information need.

2.4.1 Conceptual Document Chains

Grouping documents in a large corpus by the concepts that they contain is a significant motivation for topic models (Deerwester et al., 1990; Hofmann, 1999; Blei et al., 2003) *inter alia.* These methods use word co-occurrence methods to infer topics, which are distributions over words but can be thought of as abstract sets of ideas which dictate word choice. Topic models are capable of finding very fine grain distinctions made by authors without anyone actually labeling what the authors' intentions were (other than what they wrote) (Blei, 2012). Topic models assign a distribution over topics to every document in a corpus, and similarity in this distribution indicates a level of similarity in the information contained therein. The inferred topics can be seen as a basis to view a large corpus of documents. A knowledge worker with an abstract information need can look at a topic and often determine how relevant the entire topic is to what they are looking for, providing a way to ignore large numbers of documents (Zou and Hou, 2014). Within a topic of interest, graphs connecting documents according to even finer grain similarity can be formed by creating edges for documents whose topic distributions are close enough (Chuang et al., 2013).

While topic models can be used to map out a large corpus of documents, the idea of using actual maps of concepts has been proposed (Brner, 2010). These maps are a way of organizing information (in a scientific field for example) in a way which a domain expert would find efficient. In this category, there is work on building these maps over documents, such as the work on "metro maps" in Shahaf et al. (2012, 2013).

2.4.2 Chains of Documents in Time

One means of organizing documents in a large collection is through temporal patterns of document similarity. The clearest example of this is the tracking of a news story, prototypically, Topic Detection and Tracking (Allan et al., 1998) project. They were interested in grouping news stories into "topics" which was their term for a temporally and semantically coherent group of articles. These topics, or sequences of news stories, were a good way of discovering the complete set of information related to a developing story.

Kleinberg (2002) developed a method for detecting bursts of activity in a stream of documents. These bursts can be used to group documents in cliques or to identify events, which is useful for users who want to explore collections which exhibit these bursts. TimeMines (Swan and Jensen, 2000) proposes a similar method, but with the added ability to consider content features of the document stream. Petrović et al. (2010) describes an efficient and scalable algorithm for first story detection (Allan et al., 1998) which similarly finds clusters of documents (tweets) which are a part of a temporally coherent news event. Finally, Shahaf and Guestrin (2010) proposes an ILP-based model for detecting chains of news stories which constitute a story based on document similarity and temporal coherence.

Organizing documents by time has also been hybridized with topic modeling approaches (Blei and Lafferty, 2006; Wang and Mccallum, 2006). These topics can be seen as higher level abstractions than TDT topics. Topic models tend to have at most thousands of topics used across a collection, whereas there may be 10 or 100 times as many more news stories (TDT style topics). Graphs of documents are not viewed as the primary goal of topic modeling work like this, but there are a variety of ways to construct them from the inferred variables (e.g. document-topic distributions) such as putting an edge between any two documents which have a topic loading in the top k in the collection.

2.4.3 Connection to This Thesis

The work in this thesis which concerns building target knowledge bases (TKBs) is corpus-centric. TKBs are built from mentions of entities and situations and are therefore implicitly graphs over documents (which contain these mentions). This document structure is another way to explore a corpus, grouping documents by an entity, situation, or entity co-occurrence of interest. This view is orthogonal to the work described above which groups documents by news stories or other domain-specific concepts (e.g. metro maps).

2.5 Report-centric

In the previous category, corpus-centric approaches, information stored in natural language is distributed across *many* different documents in a corpus. Report-centric approaches index and store information in a *single* document called a report. These approaches are employed when information needs are relatively large and fall into a fairly small number of buckets. Large information needs (e.g. "malaria treatment in the 3^{rd} world") call for long form reports which are consumed largely as a whole, while small information needs (e.g. "Tom Cruise's first movie") do not warrant report generation and can be satisfied by extracting information from an arbitrarily-organized corpus. Report-centric methods' primary concern is language *synthesis* used to generate reports.

Report-centric methods are not limited to automatic methods. In fact this category is currently dominated by humans, often knowledge workers, who generate these reports. We are interested in natural language reports, but conceptually they include any synthesized information-dense formats used to communicate between knowledge workers such as slide decks, technical reports, or tabular reports. Report generation sometimes correlates with organizational decisions made by employers of knowledge workers. We have in mind cases like assigning a reporter to a beat (e.g. "violent crime in the St. Louis metro area") or an analyst to a topic (e.g. "natural gas production in Russia").

In the case of natural language reports, these methods have *conceptual scala-*

bility. By this we mean that there is nothing about the form which limits expression at various levels of granularity. This is a hallmark of natural language. You can use it to describe minute details about the process by which a hole in an airplane must be drilled (and enumerate the technical consequences of not doing so), or you can use it to describe the sentiment of the populace leading up to the French revolution. The concepts needed to express this information always have names that knowledge workers already use, and natural language can always accommodate these concepts. This is often not true for other methods described earlier. For example, DBpedia contains the entity `http://dbpedia.org/resource/Boeing_747` and various facts about it, but has no details on how it is manufactured. This is not simply because these details are private, it is because the schema has no way to express the information contained in the thousands of technical reports which have been written on this topic. The cost of the knowledge engineering required to capture these details in a formal representation is enormous. Methods which have the ability to express a wide variety of information, including report-centric methods, have conceptual scalability.

In the rest of this section we will lay out some important work which falls into the category of report-centric approaches, and touch on their relevance to this thesis. We will follow the order of work done by “most human, least machine” to “least human, most machine”.

2.5.1 Wikipedia

Wikipedia is probably the most well known example of a report-centric method for organizing and distributing information. It contains millions of reports on a range of topics from Taylor Swift to Hindustani grammar to childhood obesity in Australia. The reports have non-trivial structure which varies across concepts or topics. For example reports which describe people often use sections dedicated to “early life”, “education”, various time periods related to what the subject is famous for (e.g. for bands, time periods between the release of their albums), and “death” (when relevant). Reports on places often break up the material by history and governance, demographics, geography, economy and industry, transportation. This high level structure is ad-hoc and varies greatly, but provides a general and useful index on the information contained in these reports. Research on automatic methods for generating this type of structure is limited, but some efforts have been made (Sauper and Barzilay, 2009).

2.5.2 Knowledge Base Acceleration

The TREC Knowledge Base Acceleration (KBA) track (Frank et al., 2013) ran from 2012 through 2014 and focused on developing automatic methods for aiding, and in limited cases, creating, reports from a large stream of news. KBA systems were expected first and foremost to be able to classify news stories as being relevant to particular reports on entities in Wikipedia and Twitter.[8]

> TREC KBA is a stream filtering task focused on entity-level events in large volumes of data. Many large knowledge bases, such as Wikipedia, are maintained by small workforces of humans who cannot manually monitor all relevant content streams. As a result, most entity profiles lag far behind current events. KBA aims to help these scarce human resources by driving research on automatic systems for filtering streams of text for new information about entities.
> (Frank et al. (2013))

[8] Twitter does not contain a report for its entities, but you could imagine creating one as is already present in Wikipedia. Some entities appear on Twitter and already have reports in Wikipedia, but many do not.

The first task that KBA systems must address is vital filtering where documents in a news stream must be marked as vital to maintaining up-to-date information in a set of pre-specified reports. Annotators judge a news stories relevance to a report as either vital (contains information which would motivate changing text in a report), useful (contains information relevant to the report which can be used for evidence/citation but is not novel), neutral (contains information technically relevant to the report but not important in any sense), or garbage (doesn't refer to the report's subject).

The second task is streaming slot filling where systems attempt to predict mentions which serve as fillers for a variety of fields. Slots for people: Names, PlaceOfBirth, DateOfBirth, DateOfDeath, CauseOfDeath, AwardsWon, Titles, FounderOf, MemberOf, and EmployeeOf. Slots for buildings and facilities: Names, DateOfConstruction, and Owner. Slots for organizations: Names, DateOfFounding, FoundedBy, and Members. These fillers are not required to be entities in a knowledge base, but are judged as correct or not as strings. The point of this task is to determine when new fillers appear, cumulatively reporting all possible values for these slots over time. These slots are useful pieces of information to include in Wikipedia and other reports.

While KBA in general is motivated as a report-centric program designed to help create reports, the streaming slot filling aspect of the program is KB-centric.

2.5.3 Text Summarization

A major thread of work on automatic report-centric methods is in text summarization (Luhn, 1958; Nenkova and McKeown, 2012). There have been numerous different approaches to text summarization but the task is essentially one of automatically synthesizing a short summary or report given some source material. This source material could be a scientific paper (as was the goal of producing automatic abstracts in Luhn (1958)), newspaper articles (for which, by journalistic practices, the first paragraph is often a good summary), or in general a larger collection of reports or natural language which addresses a coherent subject.

Text summarization is justified in at least two ways. First, summaries allow knowledge workers to absorb the most important details first, which allows them to better spend their time where their attention is warranted and ignore large collections of text discussing things not relevant to their information needs. Second, text summarization methods explicitly address the issue of redundancy. When multiple documents are used as the source material to summarize, there may be a lot of overlap in the events and facts described across these documents. Text summarization systems recognize this redundancy and only present one version of the relevant information.

Text summarization work is sometimes broken down into *abstractive* and *extractive* systems. The former works in a way akin to the non-document-centric categories in this chapter and attempt to understand (build a latent representation which explains the observed texts) the sources and then generate a text summary from this latent form. Extractive systems simply find which sentences or words to cut and paste from the source materials into a summary. The distinction is often not clear in practice because most work is neither purely abstractive or extractive, but some form of hybrid. A good example of a hybrid model related to narrative chains (§2.2.2), is Barzilay and Elhadad (1997).

A lot of work in extractive summarization uses a basic framework of summary creation via source sentence selection, e.g. Gillick and Favre (2009). Within this, an important observation is that sentences need not be selected as a whole, but pieces which do not contain information relevant to a summary can be excluded. These approaches are called deletion models (Knight and Marcu, 2002; Cohn and Lapata, 2009; Napoles et al., 2011).

There has been a lot of work addressing the question of what makes a piece of text salient, interesting, or useful to include in a report or summary. Solutions to this problem have used machine learning (Mani and Bloedorn, 1998; Chuang and Yang, 2000), lexical frequency (Gillick and Favre, 2009), and models of text or conceptual centrality (Erkan and Radev, 2004).

For extractive methods which rank sentences to include in a summary by some criterion, it is important to remove redundant text instead of just putting the most relevant pieces of text into a summary. This observation has been put into practice using both greedy (Carbonell and Goldstein, 1998) and exact methods (Gillick and Favre, 2009).

2.5.4 Connection to This Thesis

This thesis makes a contribution in this area in the entity summarization work described in Chapter 7. This work addresses the problem of synthesizing reports which cover the domain of an entity. The entity summarization work in this thesis also makes connections to the KB-centric line of work through our summarization model which explicitly uses relation extractors trained with KB-level supervision (§4.3). Lastly, the predicate argument linking work described in Chapter 6 offer a way to augment existing reports by adding structure linking claims to evidence and related claims (Salton et al., 1991, 1997).

2.6 Conclusion

There has been an enormous amount of work on how to extract, represent, retrieve, and explore information collected from natural language. In this thesis we take the view that there is no one solution which will work very well for more than a narrow set of information needs. In light of this, the work in this thesis is towards the goal of methods which work at as many levels of abstraction as possible: entities, events, and documents. Especially in cases where information *exploration* is needed, what is most important is providing many different views and ways of structuring information which is comprehensible to knowledge workers.

Chapter 3

Entity Detection and Linking

Report linking is about automatically linking claims to textual sources and finding new information about a given topic. The ability to recognize and disambiguate entity mentions plays a central role in report linking methods. For instance, finding out whether someone was involved in a particular type of event, or searching for facts about a given entity, or inferring the relationship between two entities all require the ability to spot referring mentions in text.

This chapter is about taking a collection of source text containing information relevant to some reports and finding entities discussed in both the reports and sources. We assume that our methods have access to the output of a named entity recognition (NER) system and are faced with the challenge of discovering which mentions refer to the same entities.

As explained in Chapter 1, there has been extensive work on these problems under the tasks of coreference resolution and entity linking. We draw on this work in this chapter, but we apply them in a slightly different way towards our goals in report linking. We are primarily concerned with the ability to enumerate all mentions of a given entity, and this chapter explains how this is done.

3.1 Problems To Tackle

We start by discussing some challenges in entity discovery and linking for report linking, defining two important problems which we address in this chapter.

Dependence on a Knowledge Base of Entities Entity linkers can be fast and accurate, but they only work if you have a knowledge base (KB) of entities to link against. This assumption is problematic in report linking where the text sources can cover domain-specific and/or long-tail entities which will not appear in public hand-crafted KBs like Wikipedia.[1] Constructing KBs for these domains comes at a prohibitively high cost since it cannot be shared across many interested parties and because in many professional settings there is the option of falling back on simpler information retrieval tools and human effort.

In some cases there are good proxies for KBs. For example one could consider every unique email address as an entity and then link mentions of “Bill” and “Mary” in first person messages against this ad-hoc KB (Gao et al., 2017). Other times there are many specialized KBs which might hold linkable entities (Gao and Cucerzan, 2017). In general however, constructing KBs to support entity linkers is a costly process.

A key challenge for the methods in this chapter is the ability to work without a knowledge base, from text input alone. The ability to create entities on the fly is a basic

[1] 70% of the query entities in the TAC 2013 Slot filling task did not appear in Wikipedia.(Surdeanu, 2013)

skill that human readers posses and a largely unsolved research challenge.

Computational Efficiency When KBs are not available, the problem is often cast as coreference resolution. There is a long line of research on mention-pair models of coreference resolution (Bagga and Baldwin, 1999; Soon et al., 2001) which work by considering compatibility functions between pairs of mentions. These methods require $O(n^2)$ time to compute the score function and finding an optimal coreference configuration is in general an NP-complete problem (Bansal et al., 2002). These methods and their associated approximation algorithms tend to work well in practice when n is small, say the number of mentions in news article. For larger corpora like Wikipedia, TDT-style topics (Allan et al., 1998), or textbooks, these methods are prohibitively expensive.

Singh et al. (2011), Wick et al. (2012), and Wick et al. (2013) introduced methods for scaling the coreference resolution problem up to millions of mentions. Their methods work by storing mentions in a tree and having an edge-factorized scoring function which requires linear time to compute. They propose MCMC sampling procedures for inference and the worst case runtime is not well understood. These methods distribute inference over many machines but at great communication cost between nodes.

Rao et al. (2010) introduced a streaming model for coreference resolution which is appealing but has a couple drawbacks. For one it is greedy and cannot re-visit mistakes. For another they do not explain how to distribute the computation to scale beyond cases where all of the data can fit in the memory of one machine. It is not clear that a distributed version of their algorithm is possible.

A key challenge for this work is to create methods which run in time linear in the amount of text and constant for each resolution or disambiguation query and which can be distributed across many machines to scale up to large text collections.

3.2 Just-in-time Coreference Resolution

We describe a hybrid method which uses the advantages of both coreference resolution (no need for a KB) and entity linking (constant time) which we call *just in time coreference resolution*, or JITCR. JITCR is a scalable entity disambiguation technique which splits work up into two phases: **ingest**, during which a sketch/index of all the entity mentions to be disambiguated is created and **search**, where an entity is "named" by providing a mention of it which is used to find others. Ingest is highly parallelizable and takes time and space linear in the amount of text indexed. Search is roughly linear time in the size of the desired set of mentions, which is often small. JITCR enables optimizations which improve both speed and accuracy in information extraction tasks which depend on an entity (such as TAC slot filling (McNamee and Dang, 2009) and sentiment analysis (Godbole et al., 2007)) by only retrieving mentions which may play some useful role in the IE task. In this chapter we show the vanilla usage of JITCR for enumerating the mentions of an entity. Later in the thesis we show other possible usages in inferring open-vocabulary labels for pairs of related entities (Chapter 4), detecting event coreference via common arguments (Chapter 6), and exploration and summarization (Chapter 7).

JITCR is not entity linking and does not reify entities. We use an information-extraction-centric definition of entities: a set of mentions. Other attributes and facts about an entity in this view are derivable from the mentions constituting the entity, and therefore not central to the definition. JITCR retrieves mentions of a given entity given some way of "naming" the entity. In the context of report linking, the primary way of naming an entity is by specifying a mention of an entity in a document. To a human who can understand the document and reference, this is enough to pick out a particular entity, and therefore this name contains just as much information as an entity id in a knowledge base.

JITCR is similar to streaming coreference resolution, but with lazy decision making instead of greedy. Being lazy has a few advantages. First, if a mention is not a part of any entity which is queried for, JITCR does not waste time inferring its referent, which can lead to huge saving in a large corpus. Similar to batch methods, if a query is likely to be issued multiple times, its result can be cached. Second, JITCR supports anytime inference, meaning it can produce some predictions quickly, and if more time is available continue to improve its predictions using more refined scoring functions. Streaming algorithms by their definition cannot be anytime.

Finally, the decoupling of the ingest and search steps means that new mentions can be added to the index without affecting most entities.

In this section we will explain how the JITCR can be applied to entity mention search. In the next section we will show how the JITCR method can be extended to handle queries on pairs of entities, and how these extensions can actually improve accuracy.

3.2.1 JITCR Ingest

We begin by describing the process of building a JITCR index, which can be used for a variety of different queries. This step involves building an index of the text over which we are expected to find coreferent mentions. A JITCR index is an inverted index of mentions (Manning et al., 2008). The keys in this index are called **triage features**. The features we use are all case insensitive and capture word unigrams, bigrams, the headword (if parses are available at ingest time), and an acronym. These features are a form of string match which can backoff to fragments of a mention. For example the mention "Association of Computational Linguistics" would be featurized as[2]

```
u:association u:of u:computational u:linguistics
b:BBBB\_association b:association\_of b:of\_computational
b:computational\_linguistics b:linguistics\_AAAA
h:association
a:acl
```

We build the inverted index on a distributed key-value store such as Accumulo,[3] DynamoDB,[4] or Bigtable.[5] These databases can scale to text datasets which are large enough that performing annotations like NER and parsing become the limiting factor. If the corpus to be ingested is instead very small, an in-memory hashmap can provide the same functionality even faster.

In the inverted index, a form of multi-map (a dictionary where a key can be associated with many values), the keys are the triage features and the values are mention ids. We require a special property of mention ids: that they encode the id of the sentence that they appear in. This property is used later in §3.3 when searching for pairs of entities. A simple way to implement this property is to make a sentence id the prefix for all mention ids contained in that sentence.

The index only stores mention ids (and implicitly sentence ids). We store sentences (along with any annotations of them like parses and NER taggings) in another table and

[2] `BBBB` is a dummy token for "before" used at the boundaries of a mention and `AAAA` is for "after".

[3] `https://en.wikipedia.org/wiki/Apache_Accumulo`

[4] `https://en.wikipedia.org/wiki/Amazon_DynamoDB`

[5] `https://en.wikipedia.org/wiki/Bigtable`

Dataset	Num. Mentions	Bytes for sentence id	Index size
Roth and Frank (2012)	2.0×10^3	2	17.6 KB
EECB (Lee et al., 2012)	8.0×10^3	2	70.5 KB
Ontonotes 5 (Weischedel et al., 2011)	2.0×10^5	3	1.7 MB
English Wikipedia (Ferraro et al., 2014)	5.1×10^8	4	4.3 GB
English Gigaword 5 (Ferraro et al., 2014)	8.9×10^8	4	7.5 GB
FACC1 (Gabrilovich et al., 2013)	5.1×10^9	5	42.9 GB

Table 3.1: Size of compressed JITCR indices for various datasets.

only retrieve them as needed. We require a similar property of sentence ids as we did with mention ids: sentence ids should encode the document id. This allows the document context to be looked up regardless of what sort of id is held (mention, sentence, or document id).

To give some indication of the size needed, consider Table 3.1, which lists the size of indices for various datasets. All of these indices can fit on a single machine, often within memory.

3.2.2 JITCR Search

The first step of JITCR search is a triage step designed to find plausible mentions which will be re-ranked. At retrieval time we are given a set of triage features and retrieve mentions with similar triage features. We search for mentions with the highest tf-idf dot product similarity with the query (Salton and McGill, 1986).[6] In the tf-idf scheme, triage features are weighted by the product of two terms, their term frequency (tf) and their inverse document frequency (idf). Term frequency is how many times a particular (triage) feature appears in a mention. Inverse document frequency of feature i is $1 + \log \frac{D}{d(i)}$, where

[6] Cosine similarity can be computed later when the mentions are retrieved. Only then can we compute the full triage feature vector for them and normalize the dot product to compute cosine similarity. In practice we found the dot product to perform well as a triage step without the need to store indexed mentions' vector norms separately.

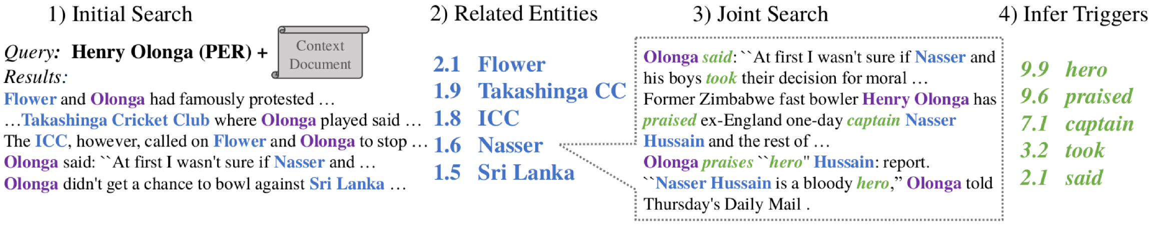

1) Initial Search
Query: Henry Olonga (PER) +
Context Document
Results:
Flower and Olonga had famously protested …
…Takashinga Cricket Club where Olonga played said …
The ICC, however, called on Flower and Olonga to stop …
Olonga said: ``At first I wasn't sure if Nasser and …
Olonga didn't get a chance to bowl against Sri Lanka …
2) Related Entities
2.1 Flower
1.9 Takashinga CC
1.8 ICC
1.6 Nasser
1.5 Sri Lanka
3) Joint Search
Olonga said: ``At first I wasn't sure if Nasser and his boys took their decision for moral …
Former Zimbabwe fast bowler Henry Olonga has praised ex-England one-day captain Nasser Hussain and the rest of …
Olonga praises ``hero'' Hussain: report.
``Nasser Hussain is a bloody hero,'' Olonga told Thursday's Daily Mail .
4) Infer Triggers
9.9 hero
9.6 praised
7.1 captain
3.2 took
2.1 said

Figure 3.1: High-level overview of the steps involved in search used to build a topic knowledge base (TKB) using the entity discovery and linking methods described here. The final step, inferring triggers, is discussed elsewhere in Chapter 4.

D is the number of documents in the collection and $d(i)$ is the number of documents in the collection which contain feature i. Since we are retrieving mentions rather than documents, we instead use inverse *mention* frequency, replacing D and $d(i)$ with counts of mentions rather than documents. For efficient representation of $d(i)$, we use a count-min sketch (Cormode and Muthukrishnan, 2005) built at ingest time. This is a compact way to represent approximate mention frequencies without having to query a database. For triage features which are queried for, the exact mention frequency can be computed which can increase the idf term for some features.[7]

The method we use for retrieving mentions is best first: the most selective (highest $\hat{idf}$) triage features and associated mentions are retrieved first. For each retrieved mention we add an entry to a hashmap storing the running estimate (lower bound) of the tf-idf vector dot product between that mention and the query. This hashmap can grow quite large, so we stop adding entries when it reaches a pre-specified size. Mentions which do not make it into the hashmap before it stops growing are pruned. We stop querying the inverted

[7] If $\hat{d}(i)$ is the approximate counts given by a count-min sketch, we are guaranteed that $\hat{d}(i) \geq d(i)$ and therefore $\hat{idf}(i) \leq idf(i)$. When the inverted index is queried, we receive a list of all mentions which contain a particular feature i. The length of this list is $d(i)$. As a post-processing/re-scoring step to the triage stage, we can replace $\hat{d}(i)$ with $d(i)$ for all mentions found, increasing (or leaving the same) every dimension in the tf-idf triage feature vector, and thus potentially increasing their dot product with the query.

index when we have covered at least 75% of the query's triage features (by L_2 norm). For example, if

$$t(m) = \{a \to 2.7, b \to 3.3, c \to 1.9\}$$

We would query the inverted index in the order $[b, a, c]$. After the first step of finding all mentions which contain b, we would have covered $\frac{||\{3.3\}||_2}{||\{3.3,2.7,1.9\}||_2} = 0.7069$ of the query triage feature vector, so we'd continue to search for a, after which we'd stop because $\frac{||\{3.3,2.7\}||_2}{||\{3.3,2.7,1.9\}||_2} = 0.9134 \geq 0.75$.

At the end of this triage process we have a set of mentions which likely belong to the entity named by the query. The next step is to refine the ranking (score) of these mentions by considering richer families of scoring functions.

The triage score is one factor which captures name similarity well, but it does not account for the context the entity is mentioned in, and therefore will lump distinct entities together. We continue in the vector space model (Bagga and Baldwin, 1998) and introduce new features for splitting these instances.

We define **context features** as a word unigram tf-idf vector which characterizes the language used near a given entity. We implement a compromise between Cucerzan (2007) (used sentences before, after, and containing a mention) and Bagga and Baldwin (1998) (used any sentence in a coreference chain). We do not assume we have coreference chains for documents in the index, but we do want to aggregate information over mentions within a document as if we did. Instead of running a coreference resolver, we use the high-precision heuristic of linking mentions with the same headword and NER type. This heuristic will fail to merge nominal and pronominal anaphors, but it has a precision of over

90% measured on Ontonotes (Lee et al., 2013). Words in the context feature vector are weighted as $\frac{2}{1+d}$ where d is the distance in sentences to the nearest mention. The only word normalization we perform for context features is mapping digits to `0`.

To further refine our scoring model, we introduce a generalization of the features defined in Mann and Yarowsky (2003) which we call **attribute features**. Mann and Yarowsky (2003) train a few ad-hoc relation extractors like `birth_year` and `occupation` from seed facts.[8] If the extracted values match for any two mentions in two clusters, this is high-precision signal indicating the two clusters may be merged.

In the Mann and Yarowsky (2003) case, one might have two clusters of entity mentions, one containing "[John] was born in Youngstown" and another containing "[John], a Youngstown native, ...". In both cases, these sentences match a `birth_place` extractor for "John" and extract the value "Youngstown". In cases where the document context for these two "John" mentions is not high, other evidence like knowing that both of these "Johns" were born in "Youngstown" can be the marginal evidence needed to conclude that they are the same person.

Our generalization is that this property holds for relations other than the handful chosen by Mann and Yarowsky (2003) like `birth_year`. Dependency path relations between nearby proper nouns are also good evidence for merging entity mention clusters. In the previous example, it is enough to know that there is some relationship between "John" and "Youngstown" (without knowing that it is a `birth_place` relation). But with this generalization, we do not need to ensure that these two mentions match a pre-trained extractor,

[8] You can train extractors for relations like `birth_year` by listing seed facts like (Wolfgang Mozart, 1756), (George Washington, 1732), (Donald Trump, 1946), etc, and then looking for precise context features appearing with sentences which mention these values. For more discussion of these methods, see §4.3.

they can cover much more general cases. For example from "[John] left Harvard to join the Obama administration" we could extract R("John","Harvard") and R("John","Obama") and match these relations to mention in other sentences like "Obama reached out to [John], a Harvard economist, ...". Inferring the relation is both difficult and of secondary importance in these cases. Knowing that this "John" has anything to do with "Harvard" is informative, and most "John"s will not have this attribute.

We call values like "Harvard" *attributes.* We collect all attributes which are either an `NNP*` or capitalized `JJ*` word within 4 edges of an entity mention. We union these attributes across mentions found by the headword and NER type coreference heuristic to build a fine-grain tf-idf vector. We use the same $\frac{2}{1+d}$ tf re-weighting for attributes, except where d is the distance in dependency edges to the nearest entity mention head. The closest attributes are descriptors within a noun phrase like `HEAD-nn-Dr.`. We include the NER type of the headword to distinguish between attributes like `PERSON-nn-American` and `ORGANIZATION-nn-American`.

There are (at least) two caveats in using string equality to check for attribute matches. First, only attributes which are *functional* relations, having exactly one output for every input (entity), can be used as evidence of a mismatch. This holds for things like `birth_year` and `mother` but not for relations like `occupation` and `daughter`. The other caveat is that what constitutes a mismatch is not clear, and string equality is not rich enough. Someone's `birth_year` being "57" is not a mismatch with "1957", and neither is someone's `mother` being "Mary Jackson" vs "Dr. Jackson". Unifying these values requires natural language understanding beyond the scope of this work.

Even accounting for these caveats, we find that adding attribute features as another similarity factor helps more than it hurts. During development we found that, when used in conjunction with the name-match coreference heuristic, these attribute features allow for inferring properties like first names when only a last is used in a mention, greatly improving absolute recall.

Coreference Score Given the triage features $t(m)$, context features $c(m)$, and attribute features $a(m)$, we define the coreference similarity score for mentions m as:

$$coref(m_{query}, m) = (1 + \alpha_t \cos\theta_t)(1 + \alpha_c \cos\theta_c)(1 + \alpha_a \theta_a) \tag{3.1}$$

where $\cos\theta_t$ is the cosine similarity between $t(m_{query})$ and $t(m)$. We only consider the subset of mentions that have $\cos\theta_t > 0$ which were found during the triage step. Any mention with a score higher than τ is considered coreferent with the query.

In §3.4 we measure the accuracy of JITCR's coreference predictions. In the next section we explain how JITCR can be augmented to perform richer queries that return more than just a list of entity mentions.

3.3 JITCR^2: Joint Searches and One Sense per Entity Co-location

In this section we show another usage of a JITCR index: performing joint entity disambiguation. In the previous section we used a mention in context to "name" an entity for JITCR to enumerate mentions for. Here we show how to use JITCR to disambiguate an entity in the context of a query and *related* entity (a joint search). There are a couple reasons for doing this.

First, as we will discuss at greater length in Chapter 4, one may need to list mentions of a pair of entities discussed in the same context. For example, in the context of a report, there may be an indication that two entities have some relation, but the exact nature is not known.[9] A search for both in context could find sentences which clarify the relationship between these entities.

Another reason to do joint search is for disambiguation purposes. Similar to attribute features from the last section, entities themselves can be seen as binary features for disambiguating other entities. If we have to disambiguate a mention of "Michael Jordan", the features for Scottie Pippen and David Blei are very discriminative features. In the context of reports, which often describe a small cluster of related entities, these related entities can be used to disambiguate mentions when enumerating an entity.

This is related to the one sense per collocation assumption. Yarowsky (1993) found that during word sense disambiguation, there were certain word co-locations which were only observed with one word sense with high probability. Similarly, entity co-locations tend to have one sense (pair of entities). For example, there may be some ambiguity over a single referent (e.g. "Jordan" may refer to the shooting guard for the Chicago Bulls, the ML researcher from U.C. Berkeley, or the country in the Middle East), but the entropy of the referent distribution is basically zero (one sense) when you condition on some co-located entities (e.g. Scottie Pippen).

[9] "Known" could mean understood with respect to the reading comprehension of either a human (e.g. "John has something to do with ACME") or a machine (which cannot infer a relation due to insufficient model complexity or training data).

This property has been used before in collective entity linking, (Han et al., 2011) where graph properties of the knowledge base being linked against are used to find one-sense entity pairs (and cliques). However, we do not need a knowledge base populated with semantic relations to find these pairs: we can take them from reports being linked.

3.3.1 Inferring Related Entities

In principle any mention in a query's discourse (i.e. report document) could be considered a related entity, but this gives us only coarse/binary signal to their relatedness. For this work we use a query-expansion method for inferring what entities are related to a query, and thus which entities are suitable disambiguation features to use during joint search.

We first perform a single JITCR query using the query. The top mentions retrieved typically have very high precision, and we use the top 100 documents as the source of related entities. The number of documents is chosen to be as small as possible (for speed) while reliably finding at least one mention of the top related entities.

Our method computes the skeleton of a topic KB (TKB) comprised of all of the entities mentioned in these top documents.[10] We process each mention in these documents and make a ternary decision to either link it to a mention in the TKB, promote this mention to an entity to add to the TKB, or ignore the mention. We link to an existing mention if its coreference score s (computed using Equation 3.1) is greater than τ. If not, we promote the mention to an entity and add it to the TKB with probability $1 - \frac{s}{\tau}$. Mentions with a score near τ may be coreferent, so we prefer low scoring mentions to avoid over-splitting

[10] This skeleton does not include edge labels between entities which is discussed in Chapter 4.

entities. Dropping mentions which are near τ is undesirable, but a small concern compared to ensuring cluster purity (Manning et al., 2008) in the TKB.

Entities' relatedness to the query is a function of how often entities are mentioned together. We model it as the sum of the joint entity linking probabilities:

$$related(q, e) = \sum_{\substack{D \in JITCR_{100}(q) \\ m \in D}} \text{logit}^{-1} \frac{coref(m, e)}{\tau} \tag{3.2}$$

3.3.2 Joint Search Algorithm

The joint search algorithm is similar to the best first algorithm for single mention search. The joint search method accepts triage features from a query and a related entity and produces a set of sentences which contain a mention of both.[11] Instead of retrieving a single key (a triage feature for a mention), we retrieve pairs of keys (one triage feature from each mention), and take the intersection of the sentences retrieved.

For joint searches, we measure the coreference score as the product of the coreference scores for the query and related entity pairs.

When retrieving from an inverted index requires an remote procedure call (RPC),[12] we find that caching the result (list of sentence ids) is both feasible and helpful. Taking the intersection of these lists is still time consuming, and best-first ordering can be used to generate high-scoring results quickly. We consider pairs of keys in best first (highest

[11] For this work we required that both entities be mentioned together in a sentence, but this can be generalized to appearing in the same document. Since document ids are a function of sentence ids, ensuring this property is no more difficult for documents than sentences.

[12] An RPC may be used in cases where the inverted index (database) is stored on a machine other than the machine performing search (application). This pattern of running the database and application on separate machines and having them communicate via RPC is common when the resource requirements for the database and application are different, as they are in this case.

idf first) by the average of the idfs for the two triage features. Similarly to in the single mention case, we populate a hashmap of sentences' tf-idf vector dot products and stop it from growing when it hits a limit, but this limit is less often hit due to the requirement that sentences in this set contain some triage feature from both entities.

3.4 Experiments

In this section we validate the accuracy of our JITCR^2 method. We want to show two things: first that the coreference predictions made using the one sense per entity co-location assumption are correct and that the entities we find most related (appear together in the most joint search results) are judged to be related to the query.

We use the TAC 2013 Slot Filling (Surdeanu, 2013) query entities to evaluate our methods; 50 person and 50 organization entities mentioned in a document (each) are used as queries. 70 of the 100 query entities were NIL (26/60 PER and 44/50 ORG), meaning that they do not appear in the TAC KB. We use annotated versions of Gigaword 5 (Parker et al., 2011; Ferraro et al., 2014) and English Wikipedia (February 24, 2016 dump) as the source material to index and retrieve from.[13] We use Amazon Mechanical Turk workers as annotators. We run JITCR with $\tau = 15$, $\alpha_t = 40$, $\alpha_c = 20$, and $\alpha_a = 10$. These constants were tuned by hand and are not sensitive to small changes. For each query we take the 15 most related entities and all entity mentions our system found which exceeded τ for judging. We call these instances where each is a sentence with two labels: a mention of the query m_q and a mention of the a related entity m_r.

[13] We do not use the coreference annotations provided by Annotated Gigaword

For each instance, we asked the annotators: COREF: Does the query mention refer to the same entity as m_q? RELATED: Is the query entity meaningfully related to the referent of m_r? These annotations are not done by the same annotators to avoid confirmation bias. Worried annotators might be lulled into thinking all COREF instances were true, we made the task ternary by adding an intruder entity (randomly drawn from SF13 queries). Annotators were shown m_q and could choose coreference with the query, the intruder, or neither.[14] We drop annotations from annotators who chose an intruder[15] because we know these to be incorrect, and compute accuracy as proportion of the remaining annotations which chose the query.

RELATED was posed as a binary task of whether m_r is more related to the query or the intruder (without highlighting m_q). In positive cases, the annotator should observe that sentence shown contains a mention of the query entity and explains why they are related. Only m_q and m_r were highlighted for COREF and RELATED respectively. The results are in Table 3.2.

Our system retrieves coreferent and related mentions with high accuracy. For coreference, mistakes usually happen when there is significant lexical overlap but some distinguishing feature that proves too subtle for our system to doubt the match, like Midwest High Speed Rail Association vs U.S. High Speed Rail Association or [English] Nationwide Building Society vs Irish Nationwide Building Society.

For relatedness, the biggest source of errors are news organizations listed as related entities because it is common to see sentences like *"**Mohammed Sobeih**, Moussa's deputy,*

[14] The order of the intruder and the query were randomized.

[15] This affected 6.1% of COREF annotations.

	PER	ORG	All
COREF	94.6	91.5	93.1
RELATED	90.7	88.2	89.5
COREF and RELATED	86.6	80.9	83.9

Table 3.2: Accuracy of the extracted TKBs for SF13 queries. Columns denote query type.

*told **The Associated Press** on Monday that...".* We may be able to address this problem by using normalized measures of relatedness like PMI or tf-idf rather than raw co-occurrence counts.

Query Entity		Related Entity	
Marc Bolland	PER	Dalton Philips	PER
Marc Bolland	PER	Stuart Rose	PER
Marc Bolland	PER	Marks & Spencer	ORG
Henry Olonga	PER	Givemore Makoni	PER
Henry Olonga	PER	England	LOC
Henry Olonga	PER	Harare	LOC
Mohammad Oudeh	PER	Munich	LOC
Mohammad Oudeh	PER	Fatah Revolutionary Council	ORG
Mohammad Oudeh	PER	Gaza Strip	LOC
A123 Systems LLC	ORG	Fisker	ORG
A123 Systems LLC	ORG	Watertown, Massachusetts	LOC
A123 Systems LLC	ORG	Obama	PER
United Steelworkers of America	ORG	Curt Brown	PER
United Steelworkers of America	ORG	Wayne Fraser	PER
United Steelworkers of America	ORG	Jerry Fallos	PER
BNSF	ORG	Santa Fe	LOC
BNSF	ORG	Chapman	ORG
BNSF	ORG	Robert Krebs	PER

Table 3.3: Example entity queries and inferred related entities used during joint search. Each entry in this table is backed by a positive coreference and relatedness judgment, but we have not listed the provenance of these judgments. For example, A123 Systems LLC is related to Obama because they were a battery maker which went out of business (following the Solyndra bankruptcy) after being supported by Barack Obama and a Department of Energy grant. In each case a snippet of the back-story is available, and was shown to annotators, derived from the mentions used to support inclusion in the TKB.

3.5 Related Work

Most of what we call entity linking today stems from the work of Bunescu and Paşca (2006) and Cucerzan (2007). They posed entity resolution as a classification problem over entities in a knowledge base like Wikipedia and trained fast and precise models. Our JITCR models share some of the same features as these entity linkers, but work on pairs of mentions instead of mentions and entities.

There are a variety of tasks which are called “entity search” in the information retrieval community. One subset of this work focuses on retrieving semantic web entities (e.g RDF triples) given keyword searches (Pound et al., 2010; Balog et al., 2010a; Blanco et al., 2011; Neumayer et al., 2012). While this work is motivated by the desire to find entities, we do not assume that structured representations for queried entities exist and building them from text and information extraction techniques. Another type of “entity search” is the work of Chang (2007); Cheng et al. (2007) which focuses on retrieving “entities” like phone numbers, prices, and (book) cover images.

Perhaps most similar to this work, Blanco and Zaragoza (2010) study the information retrieval problem of finding *support sentences* which explain the relationship between a query and an entity, which is similar to joint mention search with JITCR. Our work additionally addresses how to automatically find related entities, which are assumed given in that work.

Chen and Van Durme (2017) describe a discriminative retrieval framework originally intended as a passage retrieval step for question answering systems. In that work they did an experiment where they configured their system to retrieve entity mentions instead of passages and evaluated how well their system performed at cross document coreference resolution. This is related to our JITCR system, but where feature weights are learned rather than determined through distributional properties as they are in this work.

3.6 Summary

In this chapter we introduce scalable methods for searching for entity disambiguation and mention search. These methods are distributable to many machines, dynamic (do not require re-computation upon adding more mentions), and efficient (search requires sublinear time). We also discuss one entity sense per co-location, a method of improving entity disambiguation by searching for pairs of related entities. This chapter provides methods which will be useful later in inferring open-vocabulary labels for pairs of related entities (Chapter 4), detecting event coreference via common arguments (Chapter 6), and exploration and summarization (Chapter 7).

Chapter 4

Relation Situation Detection

4.1 Introduction

In the previous chapter we discussed methods for finding mentions of entities discussed in reports, and putting them in the topic knowledge base (TKB). In this chapter our goal is to label the edges between entities in the TKB which explain the *relationships* between them. In the last chapter we described how to enumerate sentences containing related entities, and in this chapter we work on methods for extracting situations (events and relations) from these sentences. These situations will serve as edge labels in the TKB and allow knowledge workers to explore a domain discussed in a report at an abstract level (the graph) as well at a concrete level (the mentions linked to the graph).

In §4.2 we discuss an unsupervised method which finds *trigger* words indicative of the relationship between two entities. In §4.3 we describe how to put more fine grained labels on the edges corresponding to relations like the ones in Wikipedia infoboxes. This is done using distant supervision to train relation extractors for relations like birthPlace, director (of movie), and ceo (of company).

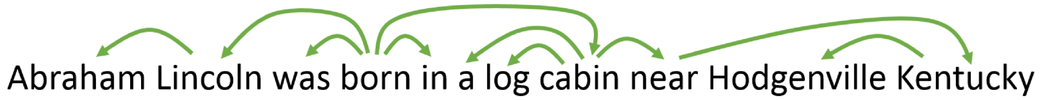
4.2 Unsupervised Trigger Word Extraction

4.2.1 Proposed Method

In this section we describe a method for choosing trigger words which explain the relationship between two entities. We choose triggers from lemmas contained in the sentences which mention two entities. We have a few prior beliefs about what makes for a good trigger word, which we state here.

Predicate (triggers) and arguments are syntactically close together. We compute the probability of two random walks in a dependency tree, one starting from each of the related entities, ending up at the same node, which is presumably the trigger word. This serves as a weak syntactically informed prior. This measure assigns some credit to any word in a sentence mentioning two entities, but prefers ones which are close to both. Probability is distributed uniformly across all of the dependency edges leaving a node, with a self-loop to ensure ergodicity. We use the power method (Mises and Pollaczek-Geiringer, 1929) to compute each random walk's probability and take the product (walks are independent) to compute a score for each node.

Information is conveyed via surprisal under a background distribution (codebook). Very common words which are close to related entities do not convey much meaning, even if they frequently appear near both entities. We compute a unigram distribution over words which are likely under the syntactic prior we just described and condition this distribution on the NER type of the two arguments. This distribution characterizes what trigger words


Abraham Lincoln was born in a log cabin near Hodgenville Kentucky

	Word	Rand. Walk log Prob	Codebook	log Repetitions	Score
	Abraham	$-\infty$			
q	Lincoln	$-\infty$			
	was	-3.196	-1.029	+0.349	-3.876
	born	-1.639	-0.969	+0.336	-2.271
	in	-2.688	-1.199	+0.414	-3.472
	a	-3.607	-1.038	+0.536	-4.110
	log	-3.607	-0.194	+0.179	-3.623
	cabin	-1.951	-0.430	+0.179	-2.202
	near	-1.970	-0.773	+0.239	-2.504
r	Hodgenville	$-\infty$			
	Kentucky	$-\infty$			

Figure 4.1: Example of trigger id algorithm in practice. Note that the log Repetitions column depends on how many times a given word was observed in $m(e)$, not just from this sentence.

to expect without knowing anything about the entities. We use this marginal distribution as a codebook. We divide out this codebook probability in every pair of related entity mentions in the TKB giving a cost in bits (log probability ratio) of each trigger word.

Repetition indicates importance. Chances are that if two entities have been predicted as related, we will have more than one sentence which might explain why they are related. We sum the costs we pay for each word in all the sentences. This corresponds to the likelihood in a generative model where all the sentences are generated independently from one another. Independence between entity mentions is too strong of an assumption (Church, 2000), so we average the measure under independence (sum of costs) with a simple max (over sentences).

This process yields a score for every trigger word type, and we use the top k triggers as edges in our TKB. Figure 4.1 demonstrates the terms involved in scoring triggers in a sentence. Table 4.1 provides a slice of a few knowledge bases built with the methods in this section.

4.2.2 Experiments

To evaluate our approach for finding trigger words, we construct TKBs for TAC KBP 2013 slot filling queries (Surdeanu, 2013). In the slot filling task, queries are given in the form of an entity (e.g. Marc Bolland) and a slot (e.g. `per:country_of_birth`), and a system must return a filler (e.g. Holland). While our system does not address this task directly because we do not model slots, we can build TKBs for the query entities and see what related entities we find and if they tend to be fillers.

After building a TKB for each query entity, we present Amazon Mechanical Turk annotators with a sentence with the query, a related entity, and two potential triggers highlighted. One trigger is chosen according to the method proposed above and the other is an intruder. We choose the intruder from all `NN*|VB*|JJ*|RB*` words in the sentence which are in the smallest sub-tree which includes both entity mentions (excluding the mentions themselves).[1] In the Lincoln example, the potential triggers are *born, log, cabin.* The annotator may choose either trigger as a good characterization of the situation involving the query and the related entity, or label neither as sufficient. Note that this baseline is strong: it shares the entity linking trigger sentence selection, and dependency parse tree as our method. We report the results in Table 4.2.

Our method performs about twice as well as the baseline, though it does not find a “sufficient” explanation about half of the time. These examples can be broken down into

[1] If node nodes match this, we walk up the tree until we find a matching node.

a few categories.

Query entity		Related entity		Triggers
Marc Bolland	PER	Dalton Philips	PER	*appointed, departure, following, move*
Marc Bolland	PER	Stuart Rose	PER	*replace, 50, Briton*
Marc Bolland	PER	Marks & Spencer	ORG	*departure, CEO, become, following*
Henry Olonga	PER	Givemore Makoni	PER	*club, president, done, played*
Henry Olonga	PER	England	LOC	*cricketer, asylum, hiding, quit*
Henry Olonga	PER	Harare	LOC	*hiding, armbands, wore*
Mohammad Oudeh	PER	Munich	LOC	*massacre, briefed, defended*
Mohammad Oudeh	PER	Fatah Revolutionary Council	ORG	*faction, belonged, return*
Mohammad Oudeh	PER	Gaza Strip	LOC	*allows, asked, host*
A123 Systems LLC	ORG	Fisker	ORG	*supplier, struck, recall, owns*
A123 Systems LLC	ORG	Watertown, Massachusetts	LOC	*produces, batteries, company*
A123 Systems LLC	ORG	Obama	PER	*plant, opening, Granholm*
United Steelworkers of America	ORG	Curt Brown	PER	*spokesman, rejected, contracts*
United Steelworkers of America	ORG	Wayne Fraser	PER	*negotiator, spokesman, union*
United Steelworkers of America	ORG	Jerry Fallos	PER	*boss, broke, shut, local*
BNSF	ORG	Santa Fe	LOC	*asked, vote*
BNSF	ORG	Chapman	ORG	*venture, help, transition, joint*
BNSF	ORG	Robert Krebs	PER	*Burlington, chairman*

Table 4.1: Examples of slices of TKBs for the three most related entities for six queries and the best triggers for each pair. Supporting sentences for related entities and trigger words are not shown.

Related Entities vs Slot Fillers There is no fair way to evaluate systems without a common schema, but we offer some extraction statistics. On SF13 queries our system generated 17.6 relevant entities/query,[2] each having 4.6 trigger words/pair, 2.1 mentions/trigger word, and 9.8 mentions/pair. In extractions from *all* systems in the SF13 evaluation (pooling answers, filtering out incorrect), they filled 6.0 slots/query with 14.2 fillers/query and 38.3 mentions/query as provenance.

[2] This is given a cap of 20 relevant entities per query.

	System	Intruder	Neither
Person	29.4	12.4	58.2
Organization	29.1	17.3	53.7
All	29.2	14.7	56.1

Table 4.2: Related entity trigger identification.

Some slots have string-valued fillers, but many could be related entities in the TKB sense. In these cases, we found 2.2 entities/query overlapping, 1.7 fillers not in their corresponding TKB and 10.8 related entities which weren't fillers.

4.2.3 Analysis

Here we discuss some cases which our trigger id model gets correct which are difficult for the frame-based extraction methods (Chapter 5) to correctly extract. The methods in this chapter are purely information theoretic and syntactic in nature, and do not involve any training data or theories of frames or roles.

Mistakes The most common reason is that the trigger is a salient event, but either the query or related entities are not a core semantic argument of it. In the following example, Harare is not a core argument to *hiding*, though it is involved in the situation (the reason for the *hiding* has to do with Harare):

> [**Olonga**]$_q$ was *hiding* in a "safe house" after Zimbabwean secret police officers traveled to East London, South Africa, to "escort" him to [**Harare**]$_r$, where he could face treason charges.

When the related entity is wrong, the triggers chosen tend to be poor too. For example:

> "Nasser Hussain is a bloody *hero*," [**Olonga**]$_q$ told Thursday 's [**Daily Mail**]$_r$.

Discourse Often a trigger is found with the two related entities a single clause. When this doesn't happen, our method prefers triggers which lie on a chain of predicates in a discourse relation. In the example below, the *wearing* event is causally related to the *mourning* event, and the query is an argument of the former and related entity of the latter.

> Flower and [**Olonga**]$_q$ *wore* the armbands during the team's World Cup victory over Namibia on Monday and said they would continue to wear them to "mourn the death of democracy" in [**Zimbabwe**]$_r$.

One can debate which is the best trigger that could be chosen in this case (*wore*, *mourn*, *death*, *democracy*, or even that no single trigger is appropriate), but this sentence is likely the best way to support the relation between Olonga and Zimbabwe. The frame-based methods from Chapter 5 do not model the discourse relations between the frames in a sentence, and would therefore not lead to an analysis connecting Olonga and Zimbabwe.

Coreference Our method tends to choose many appropriate triggers which would ordinarily require accurate coreference resolution. The examples below require at least one coreference decision to prove that the related entity is an argument of the trigger. Our model does this implicitly: in both cases there was more than one sentence containing the query, related entity, and this trigger word, supporting the idea that any coreference decisions required to unify the related entity with the trigger are warranted since it would be a coincidence to see the trigger co-occur multiple times without the related entity being an argument.

> In 1977, France arrested [**Oudeh**]$_q$, then *released* him a few days later and expelled him to [**Algeria**]$_r$, angering Israel.
>
> Mourners threw flowers at [**Maria Kaczynska**]$_q$'s cortege after her body was flown home following their deaths in an air crash in Russia on Saturday , with officials saying they would be *buried* alongside kings in [**Krakow**]$_r$ castle .

Nominal Triggers Our syntactic prior favors nodes high up in the dependency tree since they are more likely to lie on the shortest path between a query and related entity. These nodes are most often verbs, but when they are nouns, they tend to be high quality triggers. For example:

> [**Gary Hubbard**]$_r$, a *spokesman* for the [**United Steelworkers**]$_q$, confirmed on Tuesday that the international had received an e-mail from Cirri.
>
> Another suspect held along with him, [**Arkaitz Aguirregabiria del Barrio**]$_q$, is the group's "number two" and would have been [**Karrera Sarobe**]$_r$'s future *replacement* had he not also been arrested , Alfredo Perez Rubalcaba said .

4.3 Distantly Supervised Relation Extraction

While unsupervised methods provide a general way of inferring an open class of relations, they may fail to be informative or clear. Relation extraction methods which describe a pre-defined set of relations make stronger gaurantees about the utility of the relationships they find. In this section we turn to more supervised methods for training relation extractors to use as labels on entity-entity relations in TKBs.

Knowledge bases like DBpedia (Auer et al., 2007), Freebase (Bollacker et al., 2008), and Yago (Suchanek et al., 2007) offer a rich source of facts about the world. One of our goals in report linking is to find textual sources for facts about entities of interest. Learning mappings from text to facts in public knowledge bases offers an informative method for labeling the edges in our KBs which can be used on task-specific text collections.

While this data is more volumous than any text-based relation extraction data sets like ACE (Doddington et al., 2004), it is not as easy to train models because the data is only partially annotated. Previous work on distant supervision (Bunescu and Mooney, 2007; Mintz et al., 2009; Riedel et al., 2013) has shown that it is possible to learn even when the labeled data does not include links to text. In this section we make contributions on the problem of training relation extractors from KB (distant) supervision.

4.3.1 Goals

In this work we have specific goals and perspectives about how to train relation extractors using distant supervision. We will begin by explaining our assumptions and goals and what consequences they have on the methods we will eventually choose.

Explainability For the purposes of report linking, we want our model, or at least its predictions, to be introspectable. If we predict that a fact is true, for use of explaining the relationship between two entities, we want the ability to show a user *why* we believe this fact is true by offering up a claim made in a report as justification. There are many latent feature models (e.g. (Riedel et al., 2013)) which cannot offer this information, since their conclusions are affected by all mentions of entities involved in a given fact. Methods that use neural attention (Bahdanau et al., 2014; Verga et al., 2016) mostly satisfy this goal because they can show which mentions were most influential in a prediction. But methods which do not use textual evidence like Socher et al. (2013) are not of interest to us here because they cannot “show their work”. There is currently interest in models which show their work (Gunning, 2017) because they are more useful to consumers of predictions (Ribeiro et al., 2016) and modeling rationales can improve model quality (Zaidan et al., 2007).

Linguistic Plausibility We are interested in linguistically plausible representations for relation extractors. Many semanticists build their models atop a syntactic representation, and we do the same, using the Universal Dependencies (UD) representation (Nivre et al., 2016). Some NLP practitioners opt for models which do not incorporate syntax for practical concerns like computational requirements and availability of high accuracy parsers which work in their particular language or domain. Computationally, modern parsers can operate at anywhere from 600 to 15,000 tokens per second (at and near state-of-the art accuracy respectively) (Honnibal, 2016). In terms of accuracy, in the CoNLL 2017 shared task on dependency parsing, there were 10 languages which had a parser capable of achieving LAS of 90% or higher, and 48 languages with LAS of 80% or higher (Hajič, 2017). While dependency parsing is not a solved problem, high quality and fast parsers are available in many cases. Worrying that a relation extraction system will not work well because it depends on a parser which doesn't work well is a negative way of viewing the issue. To the extent that it is true, the following statement is true: the more progress is made on parsers, the better that syntax-based relation extractors will get.

Previous work has used shortest paths connecting mentions of a subject and object (Riedel et al., 2013), but this is too weak of a class of features. Consider the statement "X received a degree from Y".[3] This implies `almaMater(X,Y)`, but the shortest path `nsubj*(received,X)-prep(received,from)-pobj(from,Y)` is insufficient in entailing `almaMater(X,Y)`, since it would match statements like "X receieved an ice cream cone from Y". It is clear that a family of features at least as rich as dependency sub-graphs is needed, which would be able to additionally predicate on the edge `dobj(received,degree)` appearing in this example. Extractors which match dependency sub-graphs, for both presence and absence of specific edges, can naturally handle phenomenon like negation, reporting/belief verbs, and modals. For example we might look for dependency sub-graphs which

[3] This is similar to the patterns in Fader et al. (2011), but their confidence estimates are only derived from fully supervised sentences and therefore do not include lexicalized features, which are necessary to conclude that "receive a degree from" is an acceptable extractor.

contain nsubj*(received,X)-prep(received,from)-pobj(from,Y) and dobj(received,degree) but not neg(received,n't) to rule out statements like "X didn't receive a degree from Y".

High Precision There are multiple phenomenon which are relevant to statistical models of infering facts from text. You can imagine these phenomenon as lying on a continuum ordered by the precision, or conditional likelihood, of a fact being true given a phenomenon as evidence. At one end of this continuum we have the strongest predictors, things which we can strongly entail a fact being true. If we are concerned with facts of the form birthPlace(X,Y), an example of this would be the direct statement "X was born in Y". There are however a variety of weaker indicators, like "X went to high school in Y" or "X returned to Y". Statistically, these patterns may lead to a correct inference, but a listener cannot claim to know a fact given only them as evidence.

It is a reasonable goal to mine patterns which indicate some degree of correlation between an expression and a fact being true, but this is not our goal. We are only interested in addressing the problem of "when can we be sure that an author intended to express that a particular fact of interest was true?"

Most work on evaluating relation extraction rewards both precision and recall, though recall may be less important in practice. For example, mean average precision (MAP) (Manning et al., 2008) gives credit for precision at all levels of recall, even if those precision levels are unusably low.[4] In most applications (e.g. any user-facing sort of question answering), the utility of predictions which fall below effective certainty is very low. MAP

[4] If facts are to be used in a downstream application like question answering, there is a precision threshold at which adding noisy facts is more likely to cause a wrong answer than a right one. Other than error propagation, some systems would rather not use any prediction if they cannot be reasonably sure it is correct, such as in Jeopardy or web search.

is not the only metric which rewards precision levels which are of very low utility, MRR also rewards any system which can boost the correct answer up in the rankings, even if that answer has little or no chance of getting to the top of the list. These metrics reward methods which are capable of detecting and storing a wide variety of phenomenon which have precision well below certainty and are of questionable utility. Receiever opererating characteristic (ROC) curves offer an alternative way to report performance at various levels of precision. F_β where $\beta < 1$ is another option for evaluation where precision is more important that recall:

$$F_\beta = (1+\beta^2)\frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

4.3.2 Proposed Method

In light of the goals above, we propose a new model for distant supervision. Our method is based around precision-ranked extractors. An extractor is a set of terms, all of which must be satisfied. A term is a predicate requiring that a lexicalized dependency edge be either present or absent in a sentence.[5] Extractors have a relation r, and if they fire on a sentence containing a mention of entity s in subject position and entity o in object position, then we add the fact $\langle s, r, o \rangle$ to the output.

Because these extractors only need one observation to conclude a fact is true, they need to make very few mistakes. To accomplish this, we collect only extractors which have very high empirical precision with respect to our training data. This can be computed by just running each extractor on every mention in a large corpus of mentions which are linked

[5] We only implement terms which prohibit the presence of `neg` dependency relations, as covering `aux` edges in general is too computationally expensive.

into a knowledge base containing facts (distant supervision). Given many extractors, each of which may be very precise but seldom fire, they can be combined into a model which has higher recall but comparable precision by taking the disjuction of many extractors. The idea of ranking rules by precision to form an ensemble decision rule has been employed successfully in coreference resolution (Baldwin, 1997; GuoDong and Jian, 2004; Haghighi and Klein, 2009; Raghunathan et al., 2010; Lee et al., 2013), named entity recognition (Chiticariu et al., 2010), and part of speech tagging (Brill, 1992). After sorting extractors by precision, the number which should be included can be tuned on development data to optimize a few objectives like F_β or maximum recall under a minimum precision constraint. For this work we report an ROC curve to show the performance at various levels of precision, but we are most interested in models with high precision, at least 80%.

Where do we get the extractors? We build them from sentences which contain a fact in the KB (positive instances). We acknowledged that shortest paths are not strong enough to check if a relation is entailed by a sentence, but they are a good place to start. We start by instantiating an extractor with terms for each edge in the shortest path connecting a subject and object mention in a positive instance.

Then we perform a refinement step k times which grows new extractors with $i + 1$ terms from existing ones with i terms. At the beginning of a refinement step, we collect the top d extractors according to a heuristic score which we will discuss later. Each of these d extractors is considered “active”. We make another pass over the positive instances, and if an instance matches an active extractor, we instaniate all extractors which are a superset of the current active extractor and contain one more term satisfied by the current positive instance. The process of growth whereby only d extractors are "active" per pass is needed to ensure that only a tractable number of extractors are instantiated (this is helped by the fact that we only instantiate extractors which satisfy positive instances). In our experiments we used $d = 2^{16}$ and $k = 3$.

Extractors are naturally stored in a trie where the edges are terms. This makes lookup of extractors which satisfy an example efficient. The number of operations needed for lookup is upper bounded by the number of terms satisfied by the example, but in practice much less since the first term checked is the shortest path between subject and object mentions, which is $O(\mathit{depth}(\text{subject}) + \mathit{depth}(\text{object}))$.

Entity Types Entity types are important in relation extraction because relations have selectional preferences. For example, the subject of an almaMater fact is always a person. In general, these selectional preferences may be fine-grained: the relation director requires the subject to be a *movie*, which is a type that is not offered by coarse-grain named entity recognition (NER) models. Knowing the types of a relation's subject and object rules out many facts without needing to condition on further evidence. Further, some knowledge bases like the ones we use in this work provide entity types for some entities. The question is: how should types be incorporated into the objective when trying to learn relation extractors?

Conditioning on the entity types at test/prediction time is appealing in theory but problematic in practice. There are two ways to get the subject and object entity types: by performing entity linking into a knowledge base populated with types or by predicting the types from one or more mentions of the subject or object. The former is only appropriate when both a high performing entity linker and a high recall knowledge base are available, which is a very strict and often unreasonable assumption. The latter case requries a model for assigning types to entities given a set of mentions and requires finding these mentions, which is computationally costly.

Leaving aside these difficulties, conditioning on entity types also means storing many more statistics introduced by features which involve entity types. To get an idea of how costly this can be, imagine that every relation can have 10 different plausible typings for the subject and object entities,[6] each with their own features and statistics indicating the probability of a fact being true given these features. A model with 10 types per relation will be 10 times larger than one with no types. The subject and object types can be checked independently, saving space and pooling statistics, but this is still a large price to pay. Future work may address this problem with efficient methods for storing these statistics, but for this work we do not model types in our extractors by conditioning on them.

However, we have the types available at training time, and we should be able to utilize this information to find better extractors and models. Instead of conditioning on types, We use them as another type of observation used to score extractors. Our model is not generative, but this is related to the notion proposed in Andrews et al. (2017), where a gazeteer is generated rather than conditioned on. We describe how types are used in scoring in the next section after first discussing simpler scoring methods.

Extractor Scoring To begin, we propose a baseline method for computing precision, which we will use in ordering the extractors in our ensemble. A sensible estimator, given

[6] Entity types nest and fine grain types may be necessary to properly type a relation. For example the relation `currentClub` may be "applicable" to `athelete` subjects and `organization` objects, finer grain types like subjects who are `footballer` can only be paired with objects of type `footballTeam`. When we talk about the "types" for a relation, we may be talking about any fine grain combinations which do (e.g. (`footballer`, `footballTeam`)) or do not (e.g. (`footballer`, `baseballTeam`)) type check.

that our extractors propose facts which hold at the fact type (two entities and a relation) rather than token level (two entity mentions), is the following:

$$\hat{p}_e(f) = \frac{n_e^+(f) + \alpha \hat{p}_e(b(f))}{n_e(f) + \alpha} \tag{4.1}$$

where

- f is an extractor
- $\hat{p}_e(f)$ is an estimate of the precision of f with respect to its entity statistics (hence the subscript)
- $n_e(f)$ is the number of entities (subject and object arguments) contained in f's relation r
- $n_e^+(f)$ is the number of entities contained in f's relation r which were mentioned in at least one sentence matched by f
- $b(f)$ is the *backoff* extractor of f, its parent in the trie, an extractor with one fewer term than e[7]
- $\hat{p}_e(root) = 0$
- α is a smoothing parameter for how conservative we want our $\hat{p}(e)$ estimates to be. Because we are sorting extractors by precision on a finite sample, our precision estimates of the top extractors is going to be biased high, and α shrinks the estimate back towards 0.

[7] In future work, we may address more linguistically plausible forms of structured backoff, for example by pooling statistics over extractors which contain lexical entries which are synonyms.

This estimate is already a little more sophisticated than naive distant supervision methods which project the fact/entity level labels (i.e. knowing that a fact is true because it is in the KB) to the mention level (i.e. assuming that every mention of a subject and object which appear in a fact in the KB with a relation express that relation). This would correspond to a related precision estimate which we denote:

$$\hat{p}_m(f) = \frac{n_m^+(f) + \alpha \hat{p}_m(b(f))}{n_m(f) + \alpha} \tag{4.2}$$

Where the counts $n_m(f)$ and $n_m^+(f)$ now count positive sentences rather than entities (subject and object arguments). It should be noted that the implemenation of $\hat{p}_m(f)$ requires signficantly less memory than $\hat{p}_e(f)$ since it can be implemented by counting mentions in a stream rather than maintaining the cardinality of the sets $n_e(f)$ and $n_e^+(f)$.

Returning to the issue of using entity types as supervision, we can ask: is there any difference between an extractor which finds facts which have many types versus extractors which only fire on a small number of types? Extractors which only work on a small number of types (relative to the number of types which are possible according to the relation in question and the types provided by the KB) are suspect. First, this could be a sign that an extractor is rare, and perhaps has high precision (on a small number of facts) *by chance*, and will fail to generalize to new data. Extractors which fire on many types are less likely to have high precision by chance.

It could also mean that the semantics of this extractor are not general, and only work in special cases dictated by the types of entities which its firing is correlated with. For example, suppose we are scoring extractors for the relation currentClub. The extractor nsubj*(bowls,SOURCE)-prep(bowls,for)-pobj(for,TARGET) will have fairly high precision and work very well on (cricketPlayer, team) facts, but it will fail to generalize to (baseballPlayer, team) facts. Holding $\hat{p}_e(f)$ constant, extractors which find *more types of facts* are more likely to have more general semantics. For example we should prefer nsubj*(plays,SOURCE)-prep(plays,for)-pobj(for,TARGET) over "bowls for" if it has the same precision. The reason to use number of distinct types rather than the number of distinct entities (arguments) as a measure of generality is that the number of types is more robust in cases where a KB has a distribution over facts which differs from the distribution of mentions of it's facts. In the previous example, if there are 100 times as many currentClub facts in the KB concerning cricketPlayers than baseballPlayers, then the precision on (cricketPlayer, team) extractions will matter much more to $\hat{p}_e(f)$ than extractions of (baseballPlayer, team) facts.

Returning back to how to integrate this insight into our objective function, our "generality-regularized precision" score should be monotonically increasing in the number of types of facts the extractor covers (on positive facts), $n_t^+(f)$. But this introduces the question, shouldn't we also favor extractors which fire on many positive types *and no other types*? As long as the closed world assumption is reasonable *at the type level*, then it is good to reward extractors for being general *and precise* at the type level.

After all of this discussion, we come to a simple notion: we want precision at both the entity and the type level. Just like we can project entity labels *down* to mentions by giving *all* mentions which map to a fact a positive label, we can project entity labels *up* to types by giving *any* types which map to a fact a positive label. We can therefore define a third measure of precision:

$$\hat{p}_t(f) = \frac{n_t^+(f) + \alpha \hat{p}_t(b(f))}{n_t(f) + \alpha} \tag{4.3}$$

To integrate all of our precision estimates into a single estimate, we simply take a weighted average.

$$\hat{p}(f) = \frac{w_t \hat{p}_t(f) + w_e \hat{p}_e(f) + w_m \hat{p}_m(f)}{w_t + w_e + w_m} \tag{4.4}$$

Active Heuristic We can apply the same "generality" insight to the heuristic which selects extractors to extend. In this case however, precision does not make for a good heuristic because it does not reward extractors which fire a lot. Given that we are enumerating refinements, we do not want to refine extractors which rarely fire. It is not even clear that a modified precision estimate, such as $\frac{n_e^+(f)}{\sqrt{n_e(f)}}$, which rewards selecting common extractors to be active is desirable. We therefore choose extractors which have a high PMI (Church and Hanks, 1990) with the relation (positive facts), which is analogous to performing a statistical independence test (Minka, 2003).

$$PMI(X, Y) = \log \frac{p(X = 1, Y = 1)}{p(X = 1) \cdot p(Y = 1)}$$

where X and Y are both random variables defined over pairs of entities, $X = 1$ indicating that an extractor fires and $Y = 1$ indicating that a relation holds. PMI selects extractors which fire on the relation more than we would expect by chance. The extractors are diverse in terms of frequency $n_e^+(f)$. Selecting by PMI is equivalent to selecting by precision divided by the likelihood of the extractor firing.[8] We use the equation 4.4 for precision and estimate

[8] The other term in PMI is the probability of the relation, which is constant with respect to the extractors.

the marginal probability using the same backoff scheme.

4.3.3 Experiments

Data We validate our methods on the FACC1 corpus (Gabrilovich et al., 2013) which contains entity links from mentions in the ClueWeb09 and ClueWeb12 datasets (Callan et al., 2009). This data has over 340m documents and 5.1b entity mentions.

We use Wikipedia infobox facts as our knowledge base, provided by DBpedia (Auer et al., 2007). The Freebase MIDs in FACC1 can be mapped to DBpedia entities and aligned with infobox facts. There are 18M infobox facts which have an entity as the subject and object that we considered in this work. We consider the subset of these facts (17.6M) which appear in one of the 902 relations which have at least 1000 facts. This threshold was used to remove relations which did not have enough training data, though the threshold value itself was chosen arbitrarily. Figure 4.2 plots how many facts are observed for each relation in the KB.

This setup has about the same number of facts as the largest collection of this kind, Bordes et al. (2013a), but about 100 times more mentions due to the large text collection provideded by FACC1 and ClueWeb with additional pre-processing by Yao and Van Durme (2014). This is important for collecting statistics on our text-based extractors.

Evaluation To evaluate we slice the KB facts into train, dev, and test sets. For each relation, arguments (a pair of a subject and object entity ids) whose hash modulo 20 are 0 constitute the test set, 1 the dev set, and the rest are for training.[9]

[9] We use `murmur3_32` (Appleby, 2017) on the subject and object entity ids separated by a dash. We strip off the `/m/` prefix from Freebase MIDs.

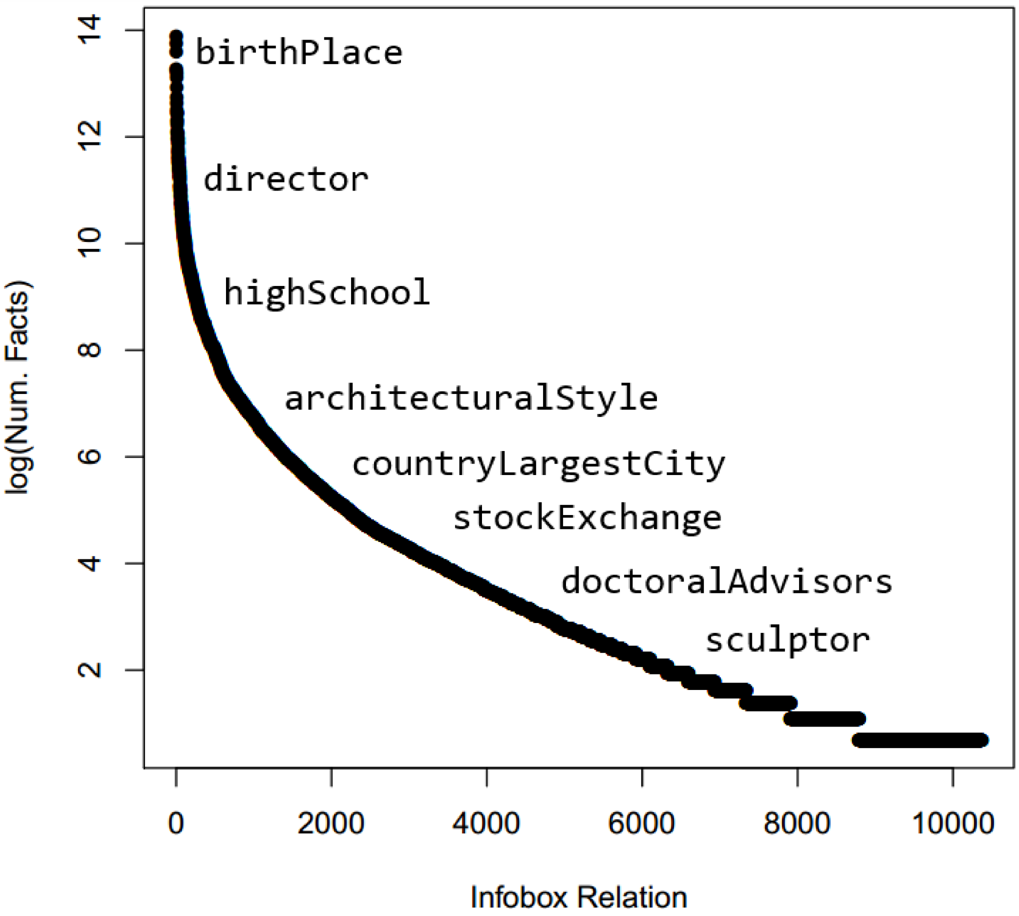


Figure 4.2: The (natural log of) number of facts available for (non-singleton) infobox relations. Relation names are appear at a height matching their frequency (most frequent: birthPlace, lease frequent: sculptor).

In the following we will refer to models which optimize "mention", "entity" and "entity-type" objectives. These correspond to assignments of $\{w_m, w_e, w_t\}$ as described above. Mention models set $w_m = 1$ and $w_e = w_t = 0$, so they optimize for mention level precision when ranking extractors and mention level PMI when expanding the set of extractors. Entity models set $w_e = 1$ and $w_m = w_t = 0$ and enitity-type models set $w_e = w_t = 1$ and $w_m = 0$. F_β is always computed at the entity level, where the two sets being compared are entities predicted by the ensemble vs entities in the held-out slice of the KB. All models are ensembles of up to 2^{20} precision-ranked extractors.

Precision Ranking Our first experiment compares different methods of computing precision. In Figure 4.3 we plot the difference in F_β performance for held out facts between the baseline method which optimizes according to entity measures like $\hat{p}_e(f)$ (this is similar to the objective used by Hoffmann et al. (2011) and Riedel et al. (2013)) and other measures of precision. Each point on the graph is a relation and we have jittered the points in the x-axis to make the values easier to see. The red dots are the results of training the model to optimize mention-level precision, similar to methods which naively project fact level labels down to mentions like Mintz et al. (2009). These dots are usually below 0, indicating that they performed worse than the baseline method of using entity measures. The blue dots are the differences in performance achieved by entity-type models. These dots are usually above 0, indicating that optimizing for type-level measures of precision and PMI are helpful in finding extractors which generalize better. In the first case, it could be that the training objective (mention-level) not matching the evaluation measure (entity-level) is what caused performance to worsen. However this cannot be true in the second case: adding the "gen-

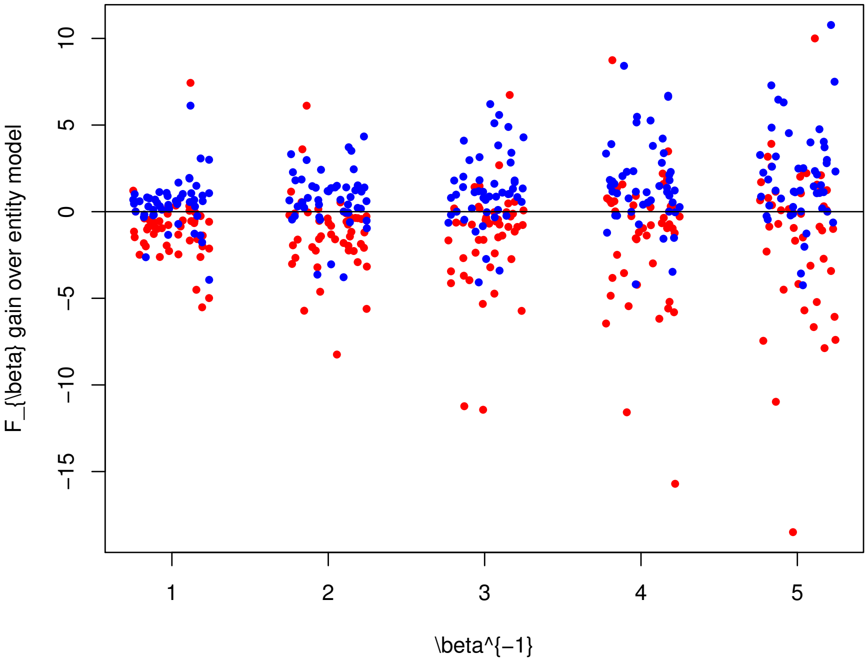
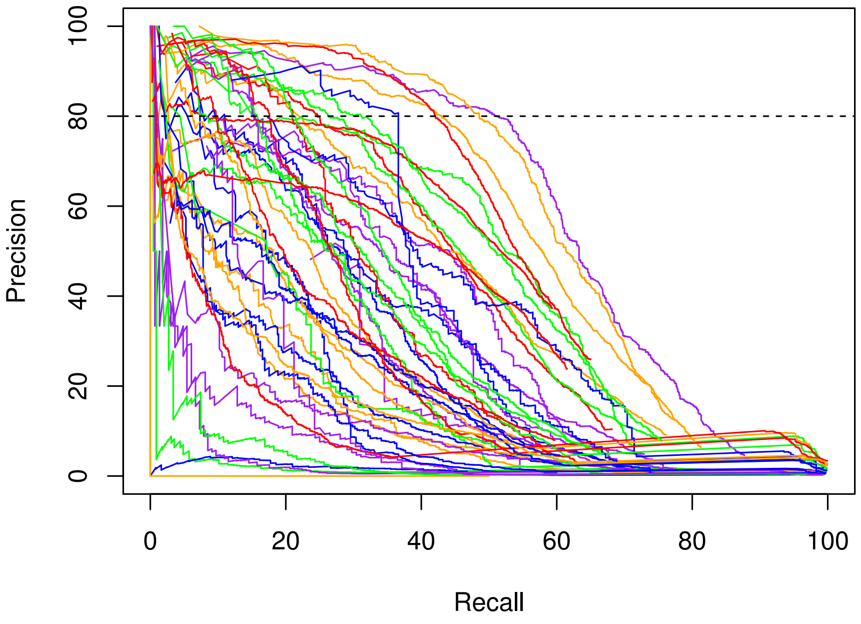


Figure 4.3: Red dots are F_β gains for relations over the entity baseline achieved by mention models. Blue dots are gains achieved by an entity-type model.

erality" *regularizer* described earlier improves held-out performance measured at the entity level. Note that both of these effects tend to grow as β gets smaller (to the right), indicating that this phenomenon is most noticeable at the high-precision end of the ROC curve.

In Figure 4.4 we plot the ROC curve for a subset of the KB relations where the colors indicate a measure of how much training data is available. In the first plot we use color to indicate how many *facts* are available for training, assigning one color to each quintile, which in order of most facts to least is: red, orange, green, blue, purple. In looking at the first plot, there is no clear relationship between the number of facts and the area under the ROC curve. There seems to be a slight trend of more facts resulting in better performance



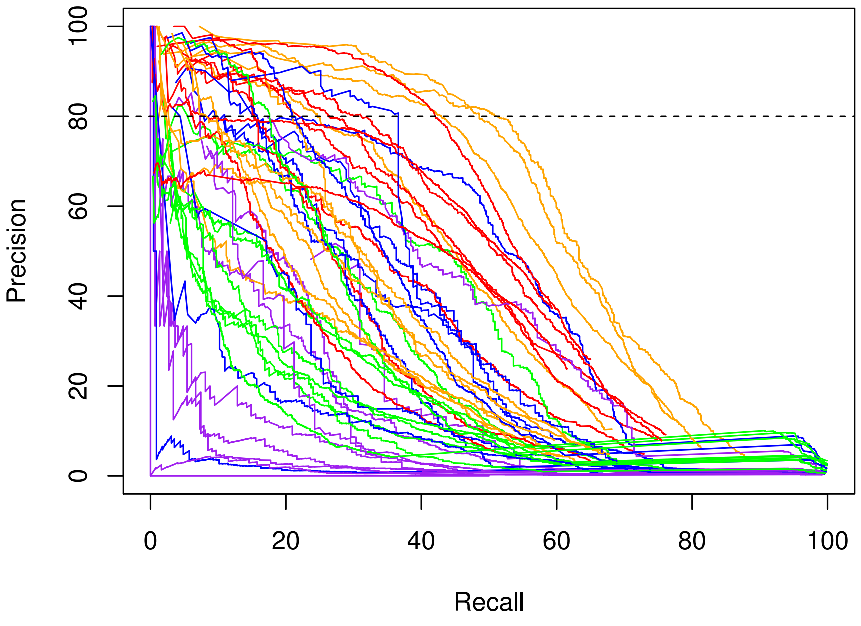


Figure 4.4: ROC curves for various relations with color indicating a measure of amount of training data. Above: number of facts in the KB for each relation. Below: number of mentions in FACC1 matching a fact in the KB. In both cases colors indicate the quantile of amount of training data. In decreasing order of amount: red, orange, green, blue, purple

Relation	Rank by mentions	Number of mentions	Rank by facts	Num. facts	$F_{\frac{1}{2}}$
starring	9	1971336	7	338051	66.96
awards	37	59919	37	53311	48.64
title	8	1689870	8	309358	29.09
language	41	89876	28	78200	10.17

Table 4.3: Sample relations and some statistics about their training data.

at the extremes (best and worst relations), but there are many counter-examples of the trend. The second plot indicates the number of *sentences* which mention a fact, and here we see a much more clear trend. The top four relations are all in the top two quintiles and the bottom six relations are all in the bottom two quintiles. There are some third quintile relations which perform well and some which perform poorly, but there are few counter-examples to the trend near the extremes. This indicates that good performance using distant supervision may be contingent on the frequency of which people talk about facts in that relation. Large collections like FACC1 may make training models easier, but smaller corpora which have a bias towards discussing the facts of interest, such as newswire may do just as well.

Next we plot four relations which exhibit a range of performances which we single out for qualitatitive description. We chose two relations which obey the trend we observed (more mentions correlates with better performance) and two which do not for further investigation. The relations and training statistics are listed in Table 4.3 and their ROC curves are listed in Figure 4.5.

In Figure 4.5 we can see a few points which need to be explained:

- awards works very well even though there are not that many training examples

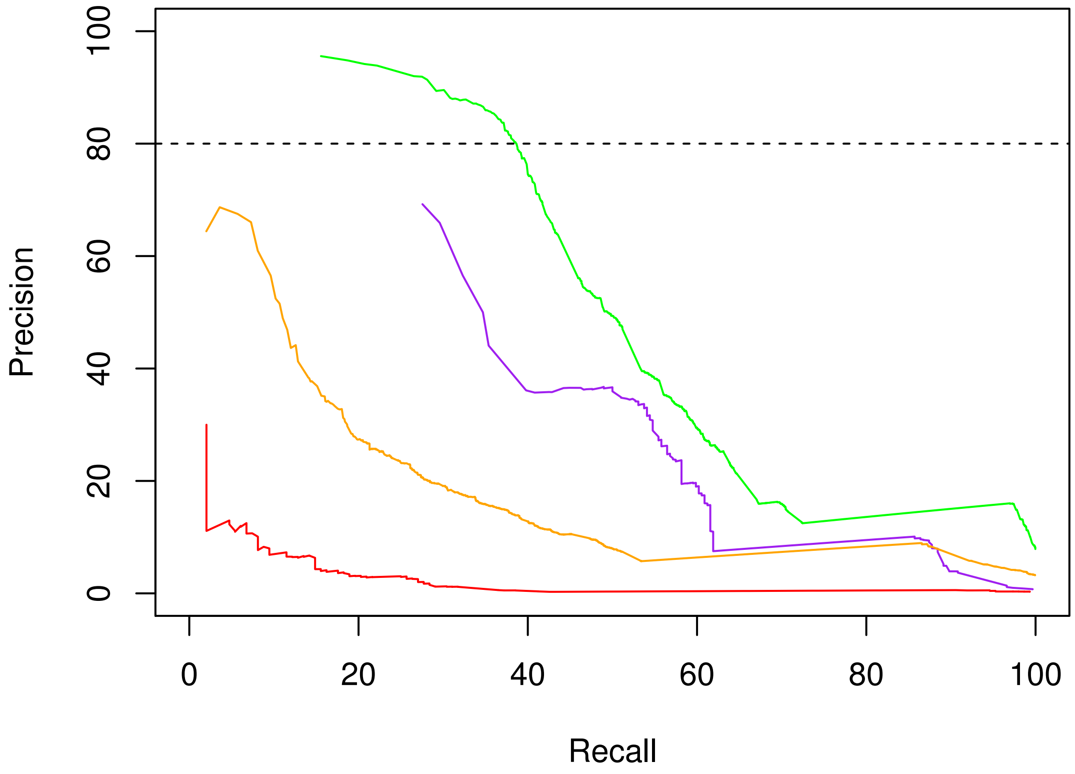


Figure 4.5: ROC cuves for select relations: green is starring, purple is awards, orange is title, and red is language.

- title works poorly even though it has almost as much training data as starring
- language just doesn't work at all, despite having a reasonable amount of training data.

In Figure 4.4 we list a sample of the facts used as supervision for the four relation extractors, and in Figure 4.5 we list the top extractors learned for each relation. The extractors for starring and awards are very sensible, occur frequently in the training data, and have high precision in held out data, all as we would expect. title seems to be an oddly defined relation which has a few usages. It seems to be a mix of song and TV *titles* and official *titles* (for a person). It is no wonder that our method cannot learn how to predict this relation, since it's facts were clearly pulled from either a mixed source or corrupted in some other way. The extractors confirm there is at least two senses with examples like `nsubj*(gained,SOURCE)-dobj(gained,title)-prep(title,of)-pobj(of,TARGET)` and `appos*(album,SOURCE)-pobj*(from,album)-prep*(single,from)-nsubj(single,TARGET)` . It is at least reassuring that even on these poorly defined relations, our method can still pick out extractors which are apprpriate for a subset of the facts which match a particular sense of the relation. Future work might analyze the bipartite graph connecting extractors and facts to find clusters of facts which belong to different senses, towards the goal of fixing the training data.

Finally, we return to language, which did not work well at all, despite having a reasonably large amount of training data. It illustrates another difficulty in this type of knowledge extraction: there are things which people never bother to talk about or write down (Gordon and Van Durme, 2013). In the case of language, it is a very well defined concept (the language a particular piece of art is rendered in), but it is rarely attested to. It has an average of 1.15 mentions per fact, compared to 5.83 for starring. From looking at the extractors, it seems that it is never directly attested to, but only through presuppositions involved in noun phrases and appositions like "Samooham, a Malaysian drama film, ...". The top extractors, while not keying off of verbal predications, appear to be high precision though, perhaps higher than our closed-world assumption evaluation lets on. Because the KB is incomplete, our extractors are likely over-penalized for proposing true facts which are

Subject	Relation	Object
Solo (TV series)	starring	Stephen Moore (actor)
The Lawrence Welk Show	starring	Ralna English
The Stranger (1946 film)	starring	Edward G. Robinson
Boat (2007 film)	starring	Emily Stofle
Tobacco Road (film)	starring	Gene Tierney
Kit Carson (1928 film)	starring	Nora Lane
Hollywood's Magical Island: Catalina	starring	Peggy Moran
Paromitar Ek Din	starring	Aparna Sen
Taking Lives (film)	starring	Gena Rowlands
The Forbidden Dance	starring	Barbra Brighton
John C. Pappy Herbst	awards	Purple Heart
Sarath Munasinghe	awards	Uttama Seva Padakkama
Otto Heidkämper	awards	Knight's Cross of the Iron Cross
Orson Leon Crandall	awards	Medal of Honor
Friedrich Beckh	awards	Knight's Cross of the Iron Cross
George Raymond Dallas Moor	awards	Military Cross
Alfred William Robin	awards	Mentioned in dispatches
Ralph Francis Stearley	awards	Commendation Medal
Dilwar Khan	awards	Ekushey Padak
Cyril Martin (GC)	awards	Military Cross
Francis Janssens	title	Roman Catholic Archdiocese of New Orleans
The Happening (song)	title	List of Billboard number-one singles
Jago Eliot	title	Earl of St Germans
Don't Go Now	title	List of number-one singles in Australia during the 1990s
John Leake	title	Rear-Admiral of the United Kingdom
Load (album)	title	Until It Sleeps
Frederick VI, Count of Zollern	title	House of Hohenzollern
Grzegorz Schetyna	title	Ministry of Interior and Administration (Poland)
Raimondo Del Balzo Orsini	title	Principality of Taranto
List of Real Time with Bill Maher episodes (2010)	rtitle	Reihan Salam
Dipu Number Two (film)	language	Bengali language
Women of Twilight	language	English language
The Great American Snuff Film	language	English language
Children of the Open Road	language	German language
Samooham	language	Malayalam
Street Knight	language	English language
White Fungus (magazine)	language	English language
Stephen Leacock Collegiate Institute	language	English language
Dreams of Speaking	language	English language
Virato Social News	language	German language

Table 4.4: Facts in the KB used for distant supervision on select relations.

Relation	Train Freq.	Held-out Prec.	Extractor
starring	3820	0.94	pobj*(in,SOURCE)-prep*(portray,in)-nsubj(portray,TARGET)
starring	1589	0.92	pobj*(in,SOURCE)-prep*(starred,in)-nsubj(starred,TARGET)
starring	6195	0.89	partmod(SOURCE,starring)-dobj(starring,TARGET)
starring	1207	0.90	rcmod(SOURCE,starred)-dobj(starred,TARGET)
starring	741	0.92	pobj*(in,SOURCE)-prep*(starred,in)-prep(starred,with)-pobj(with,TARGET)
starring	885	0.90	rcmod(SOURCE,stars)-dobj(stars,TARGET)
starring	1102	0.90	dep*(film,SOURCE)-partmod(film,starring)-dobj(starring,TARGET)
starring	1138	0.89	dep*(movie,SOURCE)-partmod(movie,starring)-dobj(starring,TARGET)
starring	633	0.92	pobj*(in,SOURCE)-prep*(starred,in)-rcmod*(TARGET,starred)
starring	1701	0.88	nsubj*(starring,SOURCE)-dobj(starring,TARGET)
awards	1033	0.71	nsubjpass*(awarded,SOURCE)-dobj(awarded,TARGET)
awards	149	0.70	rcmod(SOURCE,awarded)-dobj(awarded,TARGET)
awards	609	0.62	nsubj*(received,SOURCE)-dobj(received,TARGET)
awards	128	0.63	nsubj*(recipient,SOURCE)-prep(recipient,of)-pobj(of,TARGET)
awards	98	0.73	appos(SOURCE,recipient)-prep(recipient,of)-pobj(of,TARGET)
awards	106	0.54	pobj*(to,SOURCE)-prep*(awarded,to)-nsubjpass(awarded,TARGET)
awards	100	0.58	rcmod(SOURCE,received)-dobj(received,TARGET)
awards	54	0.83	nn(SOURCE,recipient)-nn(recipient,TARGET)
awards	95	0.62	nsubj*(awarded,SOURCE)-dobj(awarded,TARGET)
awards	42	0.71	iobj*(awarded,SOURCE)-dobj(awarded,TARGET)
title	169	0.95	nsubj*(gained,SOURCE)-dobj(gained,title)-prep(title,of)-pobj(of,TARGET)
title	61	0.87	pobj*(from,SOURCE)-prep*(single,from)-nsubj(single,TARGET)
title	146	0.79	nsubjpass*(created,SOURCE)-dobj(created,TARGET)
title	51	0.96	appos*(album,SOURCE)-pobj*(from,album)-prep*(single,from)-nsubj(single,TARGET)
title	42	0.95	dep*(album,SOURCE)-pobj*(from,album)-prep*(single,from)-nsubj(single,TARGET)
title	61	0.78	pobj*(from,SOURCE)-prep*(song,from)-dep(song,TARGET)
title	139	0.68	pobj*(by,SOURCE)-prep*(conducted,by)-partmod*(TARGET,conducted)
title	200	0.60	nn(SOURCE,champ)-nn(champ,TARGET)
title	69	0.74	pobj*(against,SOURCE)-prep*(title,against)-nn(title,TARGET)
title	160	0.60	nsubjpass*(crowned,SOURCE)-xcomp(crowned,TARGET)
language	45	0.36	nsubj*(newspaper,SOURCE)-nn(newspaper,TARGET)
language	23	0.36	nsubj*(channel,SOURCE)-nn(channel,TARGET)
language	17	0.18	appos*(channel,SOURCE)-nn(channel,TARGET)
language	6	0.67	appos(SOURCE,network)-nn(network,language)-amod(language,TARGET)
language	7	0.58	nsubj(SOURCE,newspapers)-nn(newspapers,TARGET)
language	8	0.40	appos(SOURCE,daily)-nn(daily,TARGET)
language	23	0.25	pobj*(daily,SOURCE)-prep*(TARGET,daily)
language	16	0.17	nn(SOURCE,channel)-nn(channel,TARGET)
language	10	0.19	nn(SOURCE,language)-amod(language,TARGET)
language	12	0.19	dep(SOURCE,channel)-nn(channel,TARGET)

Table 4.5: Top extractors in a shortest-path ensemble, sorted by PMI.

not in the KB. Future work could correct for this by paying human annotators, but perhaps after correcting for other irregularities in the data, such as mixed relations like title.

Dependency Relations We argued that syntax was a natural and precise way to capture the semantics entailed in a sentence. Our representation is based on Universal Dependencies (Nivre et al., 2016). While there is no easy way to compare our syntax-based approach to a related syntax-free model,[10] we can make a slight modification to see how important the syntactic structure is. In parsing it is common to report labeled and unlabeled attachment scores, and in other work on data mining with depenency syntax it is common to use paths without edge labels. To see how much information these edge labels provide, we train pairs of models, one without edge labels and one with. We report the differences in scores in Figure 4.6. Most of the time, the model which has access to edge labels performs better than the model without (dots above the horizontal line at 0). The gap in performance is fairly large too, starting at an average of about 2 F_1 points and going up as β decreases (favoring precision). This result indicates that even lexicalized dependency paths (which include direction) can be improved upon by conditioning on the type of syntactic edge being crossed.

Dependency Subgraphs vs Shortest Paths Finally we return to examine the effectiveness of our expansion method for refining extractors which only look at a shortest path into extractors which can check an entire dependency subgraph. We use shortest path ex-

[10] The author tried many different model variants which were purely lexical and based on surface features like linear distance, all with very little success. Models like this have to deal with problems of enormous vocabularies (there are more than 1M words which appear within 3 words of a subject or object for more than half of the relations tried) and sparse statistics. Syntax nicely solves this problem by limiting the set of features/extractors down to a plausible set.

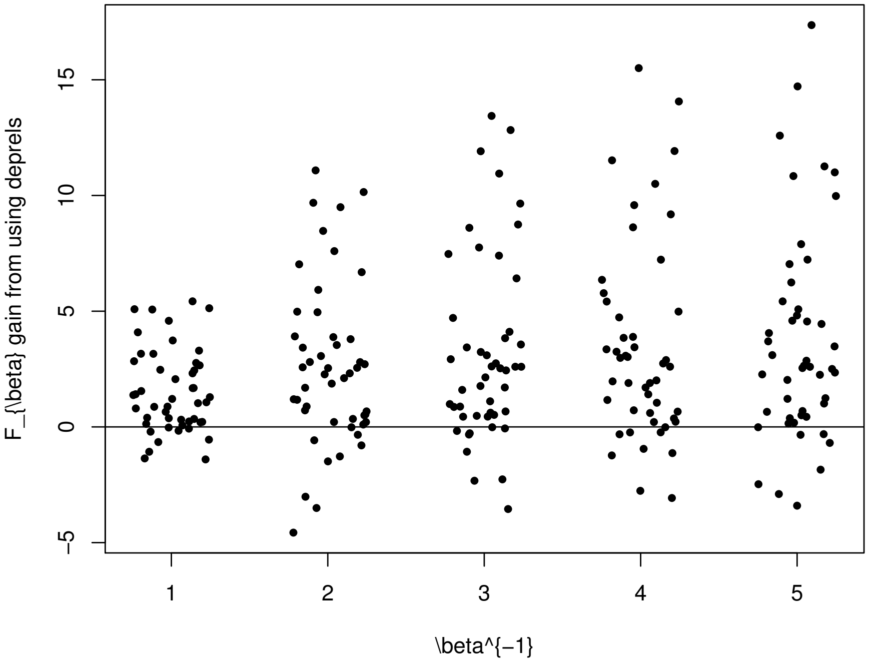


Figure 4.6: Gain for using dependency relations over untyped (but directed) paths. Each dot is a relation. Dots above the horizontal line are relations where a model which uses dependency relations outperforms one which doesn't. The left measures F_1 (precision and recall weighed equally) and the right measures $F_{\frac{1}{5}}$ which emphasizes precision more than recall.

tractors as the baseline and perform two rounds of refinement. In Figure 4.7 we plot the gain in performance for ensembles which have undergone one round of refinement (in black) and two rounds (in green). In general the refinement does improve over shortest paths resulting in an average improvement of about 1.5 F_1 points, with the benefit growing larger as β gets smaller (favoring precision). Adding two dependency edges to a shortest path extractor doesn't seem to improve much over adding just one edge. This could be because of sparsity issues involved in estimating the precision of these very selective extractors or it could be that these extractors do not improve much at the given size constraint for our models.

The most common types of additional edges that refined extractors check are:

- `ROOT(ROOT,<verb>)` edges which check that a main verb at the crest of a shortest path between subject and object is not in an embedded clause.
- `auxpass(<verb>,was)` which is the most common type of verbal modifier in cases which involve `nsubjpass` edges.
- `dobj(<verb>,<noun>)` is used to qualify cases where one of the arguments is not in object position such as the "receieved a degree from" example given earlier.
- `det(TARGET,the)` is an indirect way to type objects as proper nouns.
- `dobj(<verb>,who)` can be used to match questions which presuppose a relation such as "Who did X play in Y?" entailing `starring(X,Y)`.

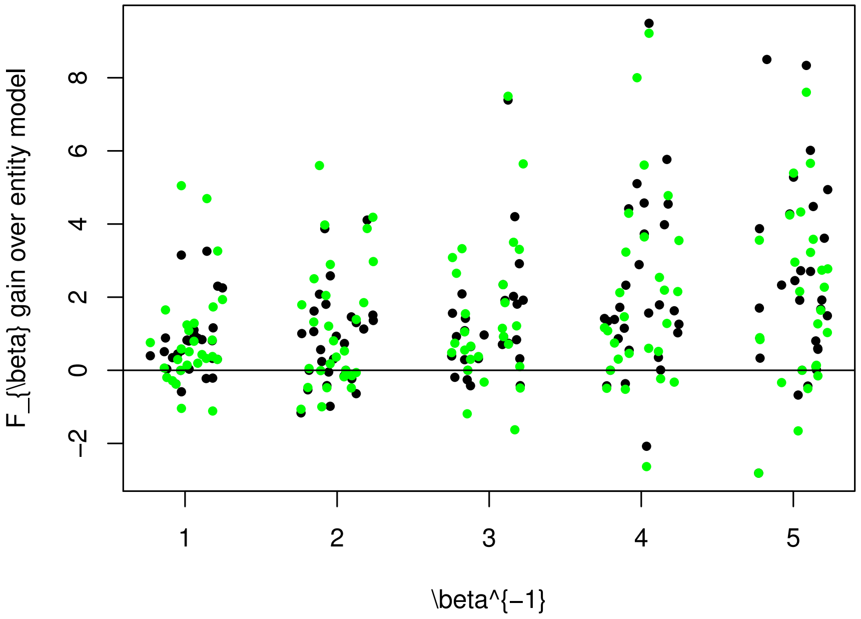


Figure 4.7: Gain for using dependency sub-graph extractors with one additional edge (black) and two (green) over shortest path extractors. While both one edge and two edge extractors perform better than the baseline (shortest path), there is little gain from adding the second edge.

4.4 Related Work

Slot Filling This chapter has largely been about populating KBs with facts extracted from text. We discussed both unsupervised §4.2 and distantly supervised §4.3 methods for labeling the edges in a KB. The most relevant line of work to this chapter is the TAC Knowledge Base Population track (McNamee and Dang, 2009) which has run since 2009. The track has included tasks for entity linking, slot filling, and cold start KBP, the last two being very related to this chapter. Slot filling (SF) is the task of finding core facts about a query entity like where a person works or who are the leaders of a company. Cold start (CS) is like SF but where no KB of entities is given at the start, so systems must induce their own entities (similar to Chapter 3).

Distant Supervision Slot filling is related to the distant supervision methods in §4.3 because most slot filling systems use some form of distant supervision to train extractors. One way to do this is to directly train slot extractors from the limited training data given (Surdeanu et al., 2010; Garrido et al., 2011; Sun et al., 2011). This has advantages which come with simplicity, in terms of ease of implementation and the avoidance of cascading errors which come with more complex pipelines.

Another way to use distant supervision for slot filling is learn extractors for another set of relations $\mathcal{R}_z$ and then map the facts found in $\mathcal{R}_z$ to the slot filling schema. $\mathcal{R}_z$ is usually taken to be Open IE patterns (Banko et al., 2007), which can have either rule-based/unsupervised extractors (Banko et al., 2007; Fader et al., 2011; Angeli et al., 2015) or supervised (Mausam et al., 2012) extractors. Mapping from Open IE patterns to the slot filling schema can be done either manually (Soderland et al., 2013; Finin et al., 2015) or automatically (Angeli et al., 2015; Singh et al., 2013).

This relation extraction aspect of this work is similar to wrapper induction (Kushmerick, 1997), who created extractors for factual information like what time movies are playing from tabular or tree-structured data on the web. The facts that they extract are different from the facts which appear in infoboxes, but they search over a hypothesis class of extractors for a high-precision and low-complexity element is the same.

Snow et al. (2005) proposed a distant supervision method similar to our work in §4.3 but for hypernym detection. They did not use precision-ranked extractors, but rather logistic regression trained on instances generated from the naive distant supervision assumption (all instances express the given relation). However, their extractors were similar in their use of syntax, although they did not consider arbitrary edges in the dependency graph.

Downey et al. (2010) studied the problem of estimating the probability that a fact is true conditioned on how many times it was mentioned in a large corpus. This is related to the work in §4.3 where we estimate the probability that an extractor produces a true fact. In §4.3 we found that a fact's mention frequency was not a good way to estimate an extractor's precision (both entity pair and entity and type pair methods produced better extractor ensembles). However, the work of Downey et al. (2010) did not involve ranking a mix of good and bad extractors, they only looked at the facts extracted by high-quality extractors produced in previous work (Etzioni et al., 2005). Our work does not speak to this restricted setting. Finally, the model of Downey et al. (2010) requires joint inference over all training instances which limits it application to batch settings. In the batch setting, this information can be used in conjunction with the work in §4.3 to refine the confidence estimates from the extractor to the fact level.

Hasegawa et al. (2004) described a method for doing relation label propagation using a method similar to our distant supervision method. They build context vectors for pairs named entities based on the words observed between them and then cluster named entity pairs. Relation labels on seed named entity pairs can be propagated to all unlabeled pairs in the cluster. Both our methods use set similarity measures and use named entity pairs (arguments). Their method has the limitation of only supporting one relation type between two argument types (e.g. a person and a university can only be labeled as `almaMater` or `professorAt`).

Trigger Extraction Blanco and Zaragoza (2010) study the information retrieval problem of finding *support sentences* which explain the relationship between a query and an entity, which is similar to this work. Our work addresses two new aspects of this problem: 1) how to automatically find related entities, which are assumed given in that work and 2) how to find the salient parts of support sentences (trigger words) by aggregating evidence across sentences.

Raghavan et al. (2004) investigated open vocabulary characterization of entities. They found intersecting entity language models for a pair of related entities yields common descriptors. Their notion of similarity (e.g. Ronald Reagan and Richard Nixon are both *presidents*) is different from our notion of relatedness (e.g. Alexander Haig and Princeton, NJ are related via *Meredith* – Haig's sister).

KBs for Information Retrieval Our work is concerned with building KBs for a query from raw text. There is related work on finding a relevant subset of an existing KB which is related to a query such as Dalton and Dietz (2013) and Dietz and Schuhmacher (2015). They create "knowledge sketches": distributions over documents, entities, and relations related to a query. They use Freebase for relations and Wikipedia for anchor text and links. Additionally their queries are richer than in this work (entity mentions), such as "scope and scale of child labor in Turkey", rather than queries of specific entities.

4.5 Conclusion

In this chapter we discussed methods for inferring relations between entities in text. Our goal was to come up with meaningful labels for the edges connecting related entities in a topic knowledge base (TKB) using textual claims made in reports. In Section 4.2 we discussed an unsupervised method which, given a set of related entities, can infer trigger words which meaningfully characterize the relationship between two entities. These methods worked twice as well as a syntactically informed baseline and are very flexible due to their open vocabulary. In Section 4.3 we discussed distantly supervised methods for inferring entity relations. These methods have the advantage of tying into a well know schema like DBpedia (Wikipedia infobox facts) without requiring costly direct supervision via annotated mentions. Specifically we showed how to find high-precision extractors which are most likely to be useful to professional analysts. All of the methods in this chapter "show their work", and can trace back claims about a TKB to claims in a report.

Chapter 5

Frame-based Situation Detection

5.1 Introduction

In this chapter we discuss new models for identifying events or situations described in text, and the relationship the participants play. “Situation” is the term we will use in this chapter to describe a wide range of events, states, and facts which may be discussed in a report. The most common case of situations we would like to identify and reason about are verbs which describe events, such as “John *stole* $10 from Mary” or “ACME *fired* their CEO, Alex Sanchez”. This chapter will describe a few different conceptions of what situations are before going on to describe general methods for identifying them in text. What is common to all of these conceptions are **frames** and **roles**. Frames characterize the type of situation which is occurring (e.g. Theft or Employment_end) and roles characterize the relationship between the participants and the event (e.g. “John” is the Perpetrator and “Mary” is the Victim). What differs are the names, definitions, and granularities of the frames and roles which are part of a situational schema, as we will see in the next section.

One reason frames and roles are important to report linking is because they offer a way of searching for situations. Frames specify a type of situation such as a company buying another company (of interest to a financial analyst) or a suspect traveling to a certain location (of interest to a law enforcement officer). Roles specify a relationship between a participant and a situation and are sometimes said to correspond to the wh- questions of "who", "what", "when", "where", and "why". If a biologist wanted to understand the consequences of a gene mutation, they might search for situations describing that gene *regulating* another gene (role of Agent) rather than that gene *being regulated* by another gene (role of Patient), since only the former are relevant to the mutation.

All of the methods described in this section are based on the theory of linking textual descriptions to a human-designed schema of situations and roles. This is not necessary in principle and there is unsupervised work on event schema induction (Chambers, 2013) frame induction (Modi et al., 2012; Cheung et al., 2013), and role induction (Titov and Klementiev, 2012; Titov and Khoddam, 2015), as well as the work in Chapter 4.

These unsupervised methods are often also referred to as distributional approaches and characterize the meaning of words, frames, or roles (or any linguistic concept) through a distribution over lexico-syntactic observations. Commonly, but more specifically, these methods treat meanings (distributions) as latent variables which can be learned by optimizing the likelihood of observed (but unlabeled) data. Would these approaches lead to frame and role representations which are "just as good" as supervised approaches?[1] Leaving the question in general aside, for semantic roles it does not seem like this is currently the

[1] Supervised approaches treat the mapping from lexico-syntactic observations to semantic representations as a a thing to be learned from a target (e.g. a linguist or native speaker).

case. One of the best unsupervised methods for semantic role labeling, Titov and Khoddam (2015), achieves an F_1 score of 82.8, while supervised methods perform considerably better under more difficult test setups: 85.93 (Johansson and Nugues, 2008a) and 87.9 (Roth and Lapata, 2016).[2] Given that inter-annotator agreement is near perfect, 99% (Palmer et al., 2005), this performance gap can be taken as evidence that there are significant divergences between distributional representations and the representations which annotators (and by extension practitioners) seek to infer.

It is possible that the annotators are seeking the wrong representations, but we will not entertain this idea in this work. For this chapter we adopt supervised methods for inferring frame and role representations. A secondary benefit of the supervised paradigm is that it allows research to proceed along two paths, one on theory and methods related to annotating frames and roles and the other concerned with general purpose methods for training statistical models to predict them. This work is on the second path.

It is perhaps worth pointing out that in this chapter we pursue models which use a centralized set of frames and roles, whereas in the previous chapter we used decentralized set of entities. Centralization, as we are using it here, is whether there is a single source of labels and label definitions. Labels for situations are frames and roles which can be centralized in schemas like Propbank and FrameNet or decentralized as clusters over mentions in an arbitrary set of text. Labels for (entity) mentions are entities, which can be centralized (a knowledge base of known entities) or decentralized (coreference annotations

[2] This comparison under-estimates the size of the gap between the approaches. The unsupervised evaluation of Titov and Khoddam (2015) uses purity and collocation rather than precision and recall. The former measures are related to the latter under the assumption of the best-possible mapping between cluster labels and true labels, so these numbers can be interpretted as maximally generous to the distributional representations. Additionally, Titov and Khoddam (2015) uses gold syntax and gold argument identification which must be predicted in the supervised case.

which reference clusters not entities). This difference is driven by the fact that entities are known to be domain specific, heavy-tailed, and effectively an open class (Jin et al., 2014). It is not clear whether frames and roles are a "closed class", which would make the choice of a centralized schema appealing. This chapter assumes that this is true and in Chapter 4 we describe other approaches which do not rely on this assumption.

5.2 Resources

We now turn to resources which define frames and roles and examples of them used in text. We use these resources to train models for identifying situations which we believe will be useful for report linking. Each of these resources has slightly different notions of frames and roles which we discuss below.

5.2.1 FrameNet

FrameNet (Baker et al., 1998) is a semantic resource based on frame semantics (Fillmore, 1982). Fillmore's frame semantics is built on the idea that words are used to evoke events or situations in the real world (or peoples' perceptions thereof) called frames, and that there is meaning tied to frames rather than just the words themselves. For example, there is a frame for Purchasing, and this frame may be evoked by many words like "buy", "sell", "sale", "shop", or "receipt". The frame which is common to all of these words has some common categories of participants, like the buyer, the seller, and the charge. These categories are called frame elements in Fillmore's work. Frames, Fillmore argued, are related to how people perceive and describe the world, and are thus a key aspect of meaning that we should attempt to extract from language.

A difficulty in putting frame semantics into practice is that there is a large divergence between frames, frame elements, and syntax. Frame semantics, unlike θ-roles (Chomsky, 1981; Carnie, 2006), is not a part of a generative theory of language, and has no theory relating it to syntax.[3] This break with many more syntactic theories of semantic roles leaves frame semantics free to model the semantics of non-verbal predicates like nouns and adjectives[4], but it also means that inferring frame labels from language is a difficult problem which inevitably draws on world knowledge.

FrameNet uses the term "frame element" instead of "role" to avoid confusion with any syntactic theories which use the term "role" (Baker et al., 1998), but we will revert to "role" in accordance with the now popular NLP task of Semantic Role Labeling (SRL), popularized by Gildea and Jurafsky (2002).[5] FrameNet has core roles, which are central to the meaning of a frame (e.g. the Speaker in the Communication frame) and peripheral roles which are more commonly used for roles like time, place, and manner. Core roles are often specific to a frame (do not appear as roles for many frames). Killer and Victim are core roles for the frame Killing. Killer appears as a role for no other frames but Victim appears with 15 other frames including Attack, Robbery, and Arson.

In FrameNet terminology a frame is "triggered" (evoked) by a lexical unit (LU), which is a lemma (or multi-word expression) with a part of speech tag. LUs are written

[3] Fillmore (1967) is an early generative theory which unified frames and syntax and had an influence on construction grammar (Goldberg, 1995).

[4] Efforts like NomBank (Meyers et al., 2004) and AMR (Banarescu et al., 2012) cover non-verbal predicates as well.

[5] Gildea and Jurafsky (2002) coined the term Semantic *Role* Labeling as such even though they based their work on FrameNet which didn't use that term; as early or before other resources like Propbank (Kingsbury and Palmer, 2002) started using the term "role" instead of "frame element".

as `kill.V`. LUs are intended as a coarse filter for determining what frames are possibly being described before statistical analysis can be used to disambiguate the senses (e.g. determining if `run.V` refers to Leadership as in "run a company" or Self_motion as in "run a race").

As a part of the FrameNet lexicon, there are 1019 frames in FrameNet 1.5, used in this work, covering situations like Killing (e.g. `genocide.N`, `behead.V`, `slaughter.V`, `infanticide.N`, etc), to Change_of_quantity_of_possession (e.g. `lose.V`) to Fall_asleep (e.g. `fainting.N` and `faint.V`). There are 1167 frame elements, which are supposed to be frame independent (Ruppenhofer et al., 2006), though most work on FrameNet parsing has shown that these frame elements are better treated as frame-specific (Das et al., 2010) (their distributional properties are not invariant across frames) in which case there are 9633 frame elements.

FrameNet also includes annotated sentences which evoke frames and frame elements which come in two varieties. The first type are called *lexical* examples which are a list of sentences for every frame which are intended to show as many variations on frame element realization as possible. The other variety is the *full text* annotations which are meant to be complete annotations of short documents (3044 sentences for training total), which are suitable for training statistical parsers. Das et al. (2010) has pointed out, and was confirmed during the experiments which lead to this work, the lexical examples can actually hurt the performance of statistical models when evaluated on data like the full text data. The lexical examples lead to a very poorly estimated base rate for most frames. Rare frames have about as much representation as common frames in the lexical data, but not in full text data, leading to poorly calibrated predictions.

5.2.2 Propbank

Propbank (Kingsbury and Palmer, 2002; Palmer et al., 2005) is another SRL resource based on extending the syntactic annotations of the Penn Treebank (Marcus et al., 1993). Propbank is concerned with annotating syntactic frames rather than Fillmore-style frames. Propbank annotates verb frames only, and is interested in teasing apart different usages like transitive vs intransitive ("we ate fish and chips" vs "we ate at noon") and causative vs inchoative ("he chilled the soup" vs "the soup chilled"). The goal was that if these syntactic frames could be identified, verb classes such as those identified by Levin (1993) could be used to determine paraphrases (e.g. "A will [meet/visit/debate/consult] (with) B" all mean roughly the same thing) which would be useful in a variety of settings like machine translation and information extraction (Kingsbury and Palmer, 2002).

The Propbank lexicon includes 9,209 different verb senses covering 6,902 different surface forms. For computational purposes, verb senses are roughly comparable to FrameNet's LUs, since they map surface forms to frames, and Propbank has fewer of these mappings (9,209 vs 13,064).

Propbank defines 30 different roles,[6] (considerably less than FrameNet's 1167) each of which has frame-specific meaning, meaning there are as many as 207,060 possible roles, even though only 31,123 are observed in the annotated data (this is considerably more than FrameNet's 9,633 possible and 2,947 observed frame elements). If you consider only core

[6] ARG0, ARG1, ARG2, ARG3, ARG4, ARG5, ARGA, ARGM-ADJ, ARGM-ADV, ARGM-CAU, ARGM-COM, ARGM-DIR, ARGM-DIS, ARGM-DSP, ARGM-EXT, ARGM-GOL, ARGM-LOC, ARGM-LVB, ARGM-MNR, ARGM-MOD, ARGM-NEG, ARGM-PNC, ARGM-PRD, ARGM-PRP, ARGM-PRR, ARGM-PRX, ARGM-REC, ARGM-TMP, LINK-PCR, LINK-SLC

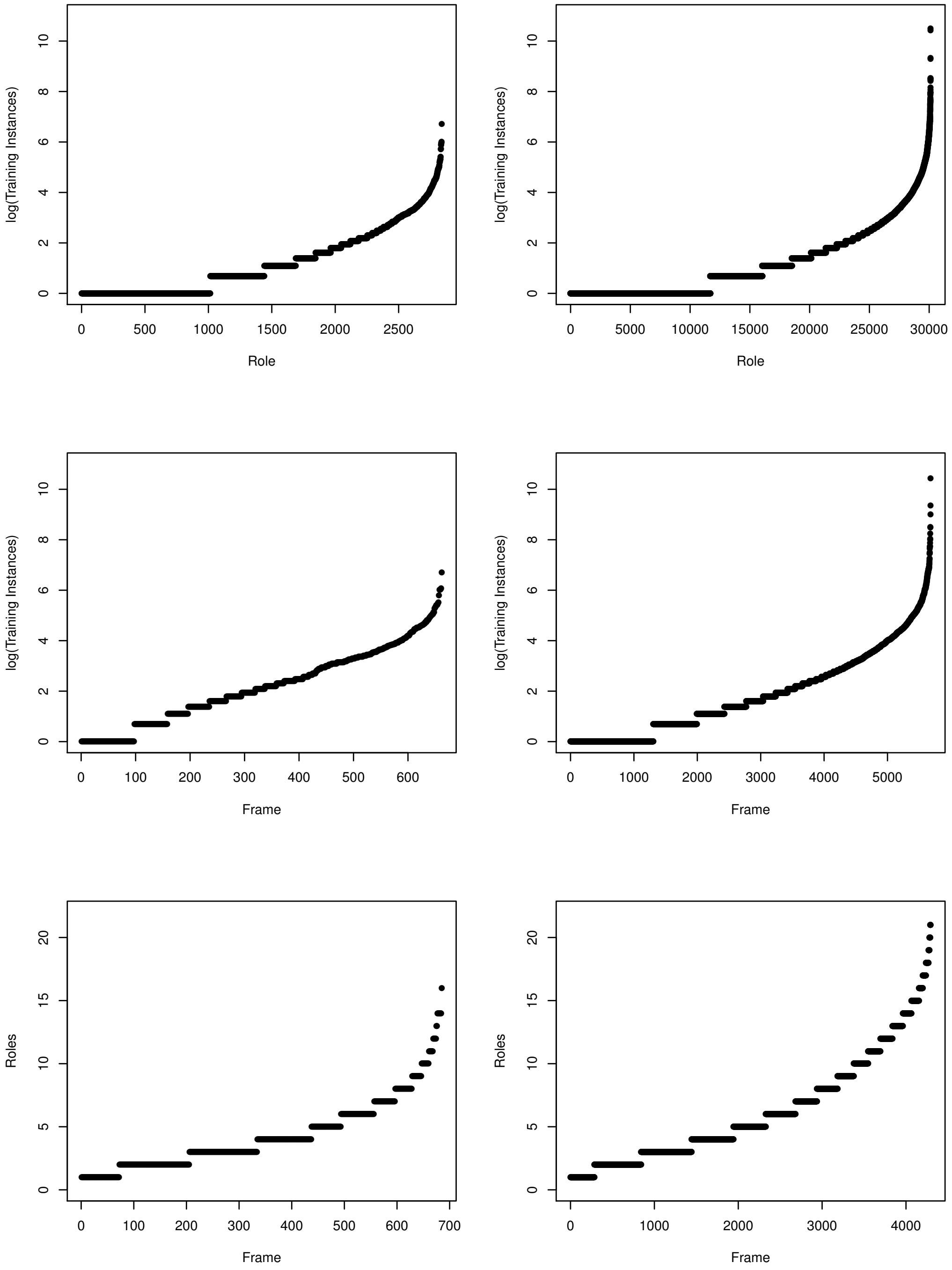


Figure 5.1: Statistics about the FrameNet (left) and Propbank (right) datasets. The first row plots the number of training instances (y-axis) available for each frames' roles (x-axis). The second row is similar but aggregating over roles to show the number of training instances for each frame. The third row is concerned with only the schema rather than training instances, plotting number of roles (y-axis) per frame (x-axis).

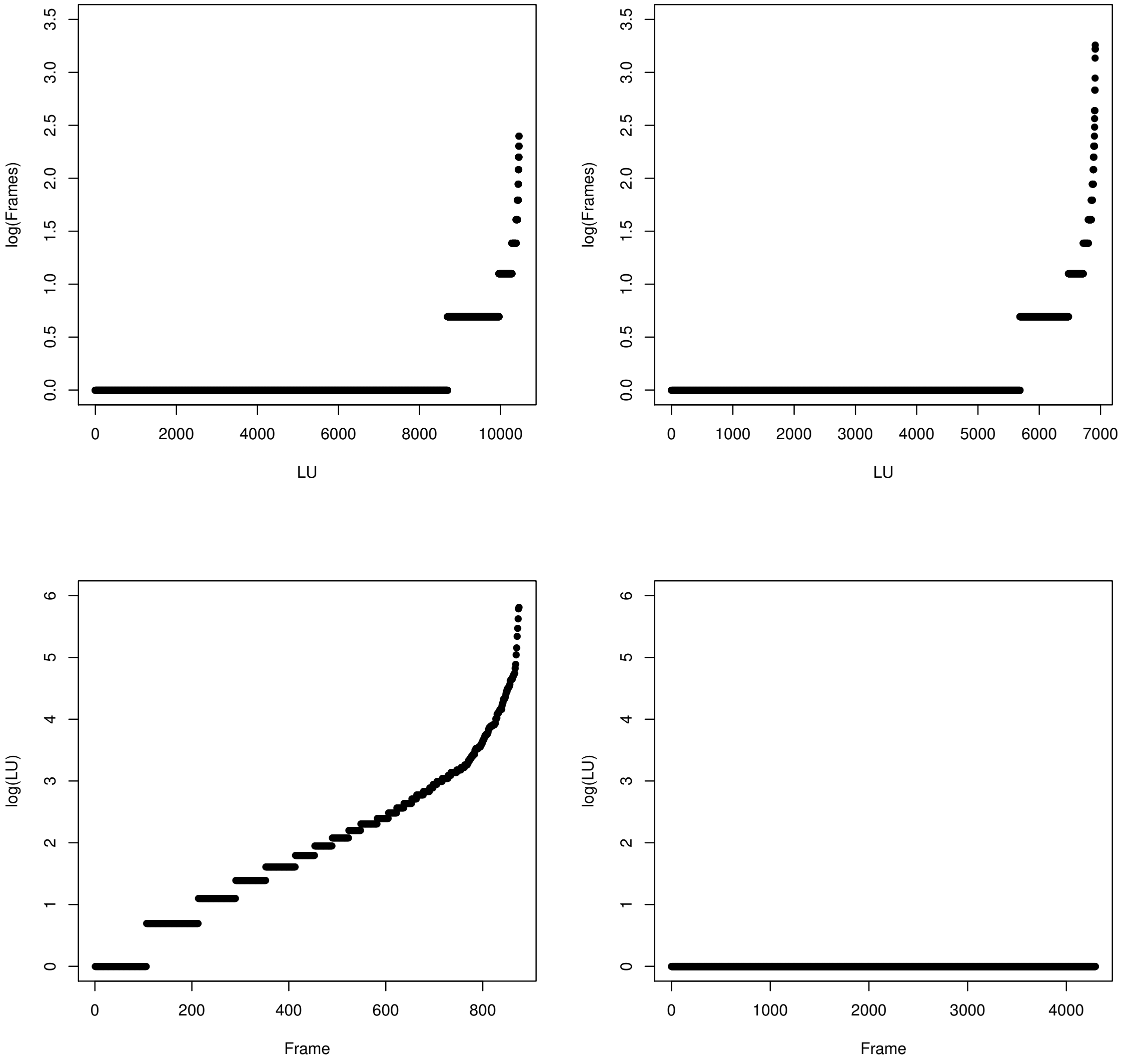


Figure 5.2: The relationship between LUs and frames in FrameNet (left) and Propbank (right). The top row plots how ambiguous LUs are, y-axis being the number of frames which correspond to an LU. The bottom row plots how many ways there are express a given frame, which is defined to be exactly one in Propbank. In the top row we see that most LUs are unambiguous (FrameNet 8691/10457 = 83.1%, Propbank 5686/6916 = 82.2%) and only a few have many senses (3 or more: FrameNet 496/10457 = 4.7%, Propbank 438/6916 = 6.3%).

roles, there are 9,956 roles.

Propbank also has a notion of continuation and reference roles. A continuation role is used when a role's filler does not form a contiguous span in a sentence. For example, in the following sentence, the $Arg1$ role is filled by a split constituent, and the second one is assigned a continuation role ($Arg1 - C$).

> [$_{Arg1}$ By addressing those problems], [$_{Arg0}$ Mr. Maxwell] said, [$_{Arg1-C}$ the new funds have become "extremely attractive to Japanese and other investors outside the U.S."] (wsj 0029)

Reference roles are explained in Täckström et al. (2015) and are used in cases where a pronoun refers to an overt argument. For example, "who" in the sentence below is not $Arg0$, but a reference to the true $Arg0$:

> [$_{Arg0}$ The spy] [$_{Arg0-C}$ who] knew [$_{Arg1}$ me]

5.2.3 Other Resources

There are a variety of somewhat related resources which target both different phenomena and representations such as Abstract Meaning Representation (Banarescu et al., 2012), Entities, Relations, and Events (Song et al., 2015), Automatic Content Extraction (Doddington et al., 2004), and the Message Understanding Conferences (Grishman and Sundheim, 1996). For a paper comparing many of these approaches, see Aguilar et al. (2014) or Abend and Rappoport (2017)

Palmer (2009) describes Semlink, a project designed to link the resources of FrameNet, Propbank, and VerbNet. This project has improved type coverage on some corpora, but no systems with competitive performance use this type of resource due to the inexactness of some of the mappings (leading to noisy features).

Another conception of semantic role labeling is the CoNLL 2008-2009 joint task on dependency-based syntax and semantics (Surdeanu et al., 2008). The dependency representation used there differs from the span (FrameNet) and constituent (PropBank) based representations used in this work. The task derives its labels from the constituency labels of Propbank with head rules applied to convert to a dependency tree.[7] The challenge focused on joint methods for parsing syntax and semantics.

Finally there is the semantic proto-roles work of Reisinger et al. (2015) and White et al. (2016) which is based on Dowty (1991). The idea is to factorize roles into properties corresponding to questions about the role such as "did this entity intend for this situation to happen?". From a modeling point of view, using a variety of binary questions to describe a role rather than a discrete set of categories can lead to richer and more statistically efficient models.

5.3 Transition-based Semantic Role Labeling

In this section we will introduce some statistical models for making SRL predictions for frames and roles given by either FrameNet or Propbank. The core contribution of this chapter is the implementation of and experiments on transition based models for SRL. The transition based models explored in this chapter are related to pipeline models, but generalize them in a) the ability to add non-deterministic inference through the use of either look-ahead or a beam and b) the inclusion of global factors which consider joint variable assignments (whereas pipeline models are locally scored).

[7] Aspects of the dependency-based work have extended to span-based work as well. For example (Täckström et al., 2015) used a span pruning heuristic derived from dependency syntax in a fashion inspired by a dependency SRL model.

In order to train these transition-based models we turn to imitation learning (Schaal, 1999). Imitation learning is related to reinforcement learning (Sutton and Barto, 1998), but where the learner does not have to solve either the exploitation vs exploration problem or the credit attribution problem. Put another way, imitation learning is appropriate when there is no delayed reward (every step incurs a cost or gain) and there exists an oracle or teacher which can give supervision for what action is most profitable to take in every state. When these conditions are met, imitation learning is more statistically and computationally efficient to train (Schaal, 1999).

5.3.1 Problem Formulation

We begin by describing the problem formulation which is a sentence-level structured prediction problem. Let x refer to a sentence and its POS tags and dependency parse. We are given x and a set of triggers which are spans in x which evoke a frame. We are concerned with predicting a binary tensor $Y(t, f, s, k)$ called an assignment. $t \in T$ is the given set of triggers, $f \in F$ is the set of frames given by the schema, $s \in S(t)$ is a set of spans which can be role fillers, and $k \in K(f)$ is the set of roles for a given frame given by the schema. $S(t)$ is a subset of all spans in the sentence derived using the heuristics described in Xue and Palmer (2004) which has a special value $\varnothing$ denoting a role is not filled (or null-instantiated in FrameNet terminology). Frame predictions are defined with special dummy indices for s and k called †. For example, in the sentence "$_0$ John $_1$ bought $_2$ vegetables $_3$ for $_4$ dinner $_5$", $Y\big([1,2), \textsf{Commerce\_buy}, \dagger, \dagger\big)$ means the frame Commerce_buy is evoked at the trigger located at $[1, 2)$ (saying nothing about its roles).

There are constraints on the values in $Y(t, f, s, k)$.

1. If w is the LU appearing at t and w has not been observed with f in the training set, then $Y(t, f, \dagger, \dagger) = 0$.

2. Frame assignments must come before the corresponding role assignments.

   That is $\forall s, k\ Y(t, f, \dagger, \dagger) = 0 \implies Y(t, f, s, k) = 0$.

3. Argument spans do not cross or overlap for the same trigger.

   That is $\forall t, f, k_1, k_2\ Y(t, f, s_1, k_1) = 1 \land \texttt{overlap}(s_1, s_2) \implies Y(t, f, s_2, k_2) = 0$.

4. Some roles are either mutually exclusive or require each other.

   In the first case:

   $\forall t, f, s_1, s_2\ E(f, k_1, k_2) \land Y(t, f, s_1, k_1) = 1 \implies Y(t, f, s_2, k_2) = 0$

   and the second case:

   $\forall t, f, s_1, s_2\ R(f, k_1, k_2) \land Y(t, f, s_1, k_1) = 0 \implies Y(t, f, s_2, k_2) = 0$

   The relations E and R are provided by the schema.

Enforcing these constraints while maintaining the flexibility to vary transitions systems like we will in this work is non-trivial. Our approach, which we describe in §5.3.3, is to enforce these constraints through a few global features which have efficient implementations. Our models learn weights which softly enforce these constraints.

Transition Systems A $Y \in \mathbb{Y}$ specifies which variables are true or false. Let $\mathbb{A}$ be a space of the same size and dimensions as $\mathbb{Y}$, but with different semantics: for an $A \in \mathbb{A}$, $A(t, f, s, k) = 1$ means that assigning that index (t, f, s, k) may be considered as an action at the current state. An A tensor specifies the set of indices, or *actions* which are allowed.

A transition system T is a function $\mathbb{Y} \to \mathbb{A}$. Elements of the domain are states (a partial assignments) and elements of the co-domain (range) are a sets of actions out of the current state.

A model M chooses an action in a state and has type $\mathbb{Y} \times \mathbb{A} \to \mathbb{Y}$. We are concerned with deterministic models which have the form:

$$(s, a) \in \mathbb{Y} \times \mathbb{A} \mapsto s \diamond \operatorname*{argmax}_{a \in T(s)} \theta \cdot f(s, a) \tag{5.1}$$

where $\diamond : \mathbb{Y} \times \mathbb{A} \to \mathbb{Y}$ the the action application function which takes a state and a chosen action and returns the state resulting in taking that action (in this case this is a pointwise addition of the state and action tensors).

A transition system and a model composed together have the type $M \circ T : \mathbb{Y} \to \mathbb{Y}$. Inference is the process of computing the fixed point of $M \circ T$ (finding a final state) starting from the initial state $Y(t, f, s, k) = 0 \; \forall t, f, s, k$.

In this formulation we have factorized the transition function (mapping states to states) into $M \circ T$. The purpose of this is twofold. First, the number of actions possible at every step is one factor in inference runtime. Conservative transition systems might consider many actions at every state, giving the model the chance to choose from as many possible actions which are consistent with the gold label. More aggressive transition systems consider fewer actions per state, forcing assignments to be completed quickly. The second reason is explained further in §5.3.4 but is related to the relationship between the order that actions are taken and the global features which fire. These differences can lead to models which instantiate more or less global features and have access to more reliable predictions earlier.

Probabilities This transition framework is deterministic, not probabilistic like related frameworks like (PO)MDPs. Probabilistic variations have some nice properties like a smooth objective function. Deterministic models have piece-wise constant objectives because of the max at each state. Small changes in parameter settings therefore cannot lead to small probability masses moving from lossy states to less lossy states, making gradient based optimization impossible. However, there are a couple advantages of deterministic transition systems. First, their objective functions are closer the objective function we actually care about. Most of the time one would use an MDP to train a structured prediction model the desired output is a one-best prediction (which we would like to be right) rather than a probability distribution over labels (the objective in MDP settings).

The second reason to prefer a deterministic framework like this is compatibility with beam search. When the model of computation for inference involves a beam, the notion of "probability of being in a state" is not well defined and not similar to how inference is conducted. As will be discussed in §5.3.5, it is possible to promote a deterministic transition system to one with a beam, and define related objective functions which are beam-aware. With probabilistic models, adding beam decoding to a locally trained model can actually hurt performance (Vaswani and Sagae, 2016).

5.3.2 Experimental Design

Before proceeding to methods, we describe how we evaluate. We measure performance on two data sets, the Propbank annotations (Kingsbury and Palmer, 2002) available in the Ontonotes 5.0 corpus (Pradhan et al., 2012) and FrameNet 1.5 (Baker et al., 1998).

For all learning methods we average the weights across all iterations of training

(Freund and Schapire, 1999). This is explicitly called for as a part of LOLS (c.f. §5.3.6) (Chang et al., 2015) and is also a standard trick used with the structured perceptron (Collins and Roark, 2004).

We use the local features described in Hermann et al. (2014) for argument and frame identification, but we did not use their feature embedding method since it performed about as well as the sparse feature method and was slower. We use the best refinements using the process described in §5.3.3.

We are studying the fully greedy case of inference in this work (i.e. a beam size of 1). As far as we know, efficient beam search and easy first inference are mutually exclusive goals, and we focus on the latter. Our implementation uses a heap to store actions in a manner similar to Goldberg and Elhadad (2010). This way actions can be generated once, instead of once per transition, and global features perform sparse updates to the actions on the heap. For beam search, states cannot share a heap (since their histories, and thus global features, would be different), so actions generation, global features, and action sorting would have to occur at every transition.

All performance values shown here are measured for the task of frame semantic parsing (FSP), meaning that we measure precision, recall, and F-measure where every index in f and k are considered predictions. Predictions in k are not correct unless the frame that they correspond to are also correct. We show two scenarios: gold f refers to the case where the frame labels are given and auto f refers to when they are predicted by the model. All figures and plots are on dev set performance.

5.3.3 Global Features

In this work we explore the use of global features for our SRL models. A global feature is one which can inspect the value of more than one assignment/action. Put another way, global features condition on the state and the action rather than just the action (variable assignment out of context). Global features are important for a couple reasons. First, a variety of insights and statistical regularities from previous work (Punyakanok et al., 2004; Toutanova et al., 2008; Täckström et al., 2015) can be described using global features on states and actions. Our definitions will not be fully equivalent to the formulation in previous work, but will draw on the same set of information. Second, global features are by their nature very expressive, and may help our models avoid myopic variable assignments. Next we will list our global features and their motivations.

`numArgs` is a global feature template which counts how many arguments a given predicate has realized in a sentence. This is perhaps the simplest type of information which is expressable in a global model but not a local one. This is useful because it serves as a dynamic or contingent intercept. Normally an argument is predicted if its score exceeds 0 (or the score of the action corresponding $\varnothing$), but with this global feature that threshold also depends on how many arguments have already been labeled.

The remaining global features are pairwise features, meaning they can be expressed as templates of the form $h(a_i, a_t)$ where a_i is any action in the history s_t and a_t is the current action to be scored.

`roleCooc` is a feature template which expresses which roles co-occur with each other in a predicate argument structure. There are some hard role co-occurrence constraints in the annotation guidelines for both Propbank and FrameNet which this feature aims to learn. For Propbank, continuation and reference roles may not appear without their base counterpart. FrameNet does not have this distinction between base, continuation, and reference roles, but instead has some mutual exclusion relationships between frame elements (roles) such as the Entities, Entity_1, and Entity_2 roles for the Similarity frame. Entity_1 and Entity_2 require each other's realization and both are mutually exclusive with the Entities role. These roles exist so that there is a sensible way to annotate sentences like "[The two painters]$_{\mathsf{Entities}}$ were [alike]$_{\mathsf{Similarity}}$" as well as "[Our economy]$_{\mathsf{Entity\_1}}$ is [like]$_{\mathsf{Similarity}}$ [a healthy plant]$_{\mathsf{Entity\_2}}$"

If $R(a_t)$ is a function which returns the role of an action a_t (assuming a_t assigns a value in k), then the pairwise definition of this feature is $h(a_i, a_t) = (R(a_i), R(a_t))$.

`argLoc` is a feature template which describes the linear relationship between argument spans. This relationship $pos(s_1, s_2)$ is the all-pairs relationship between the starts and end indices of the two spans, where two indices are said to be either "left", "left and bordering", "equal", "right and bordering", or "right". If $E(a_t)$ is a function which returns the span of an action a_t (assuming a_t assigns a value in k), then $h(a_i, a_t) = pos(E(a_i), E(a_t))$ This can encode overlap, nesting, or boundary relationships between argument spans.

`roleCoocArgLoc` is the pointwise product of `roleCooc` and `argLoc`. This feature can capture regularities like "a continuation role is to the non-bordering left of the base role" which depend on information from both `argLoc` and `roleCooc`.

Finally `full` refers to all templates together.

Refinements We designed the features in a way as to be overly general. For example, consider numArgs and its effects for various frames. A value like 4 may be very unlikely for a frame like see-v-3 which was instantiated with exactly one argument in each of the 24 times it appeared in Propbank. But, a value of 4 is below average for a frame like afford-v-1 which was observed 43 times with an average of 4.2 realized arguments.

While numArgs seems like it should depend on the frame, there are other cases like the FrameNet role exclusion and requires relationships which should hold regardless of frame. For example, the frames Amalgamation, Becoming_separated, Cause_to_amalgamate, and Separating all have the same pattern concerning the Parts, Part_1, and Part_2 roles. These frames were seen only 2, 2, 9, and 12 times in training data respectively, so generalizing this rule by pooling training data is crucial.

To choose the right granularity for the global feature templates, we consider multiple refinements. A refinement of a template is the result of taking the pointwise product of the template with one or two label features templates. The label feature templates we consider are constant (a backoff feature), frame, role, and frame-role. For each global feature template, we try each refinement and use the one with the best dev set F-measure when trained with LOLS (c.f. §5.3.6).

5.3.4 Action Ordering

An important factor in the effectiveness of global features is the order in which frames and roles are assigned.

Easy First The first motivation is related to easy first inference (Shen et al., 2007; Raghunathan et al., 2010) inter alia. The idea is that the "easiest" decisions should be made first because there is less risk that they are wrong and may be more safely conditioned on in making future decisions than any other action. To implement this heuristic, we define two variants of the **easyfirst** meta action ordering. **easyfirst-dynamic** chooses the variable index corresponding to the highest scoring action. **easyfirst-static** chooses variable indices sorted by the dev set F-measure of the local model (most accurate visited first).[8]

Baselines The **freq** ordering sorts actions by how frequently their role appears in the training set, most frequent first. This be seen as a very naive version of **easyfirst**, but with the nice property that it is independent of the local model.

From a model (estimator) bias and variance point of view, we should expect dynamic orderings to have higher variance (whether they have lower bias is a somewhat related but empirical question). In our case, we could track this variance by proxy and look at the number of nonzero global features, as is common in the sparsity-inducing regularization literature. Consider training a model with the `roleCooc` global feature on single example, a frame with K roles. With **easyfirst-dynamic**, there are K^2 possible `roleCooc` nonzero features, whereas with **easyfirst-static** and **freq** the maximum is $\frac{K(K-1)}{2}$ since the order is fixed at training time.

To see if increased variance is responsible for potential differences in the **easyfirst** variants, we construct a parallel situation with random orderings: **rand-static** and **rand-dynamic**. The first chooses a random ordering over roles which is used throughout training

[8] F-measure is computed from MAP estimates of precision and recall under a $\beta(1, \frac{5}{4})$ prior, slightly rewarding frequency.

and testing, and the second chooses a random ordering every time inference is run.

Results In Figure 5.3 we have plotted models trained with each global feature type and each action ordering. The first thing to notice is the variance across different action orderings is generally larger than the variance across different global features (for the best action ordering). This indicates that action ordering is important, perhaps more so than the global features used. This is an important result considering that most previous work on transition based inference has not addressed automatic ordering.

Next, there is little consistency between Propbank and FrameNet. We believe the major reason for this is the amount of training data (Propbank has 20.7 times more instances and 1.58 more instances per type), causing overall accuracy to be higher and **easyfirst** inference to work better.

Looking at the number of non-zero global features, we see virtually no correlation between that measure of capacity and performance, on either data set. While this metric is often used in static (local) models to describe capacity, we believe this metric is less meaningful with global features.

Note that **rand-dynamic** works well on FrameNet, only losing to a non-random ordering once (**easyfirst-static** on `argLoc`). Given the overall worse performance of our model on FrameNet, and the dearth of training data, we hypothesize that **rand-dynamic** is actually providing a regularizing effect similar to dropout (Hinton et al., 2012). Since both **rand-static** and **rand-dynamic** are random, they offer no real signal they could differ on (bias is the same), and using the standard bias-variance argument we should expect **rand-static** to do no worse since **rand-dynamic** introduces additional variance into the

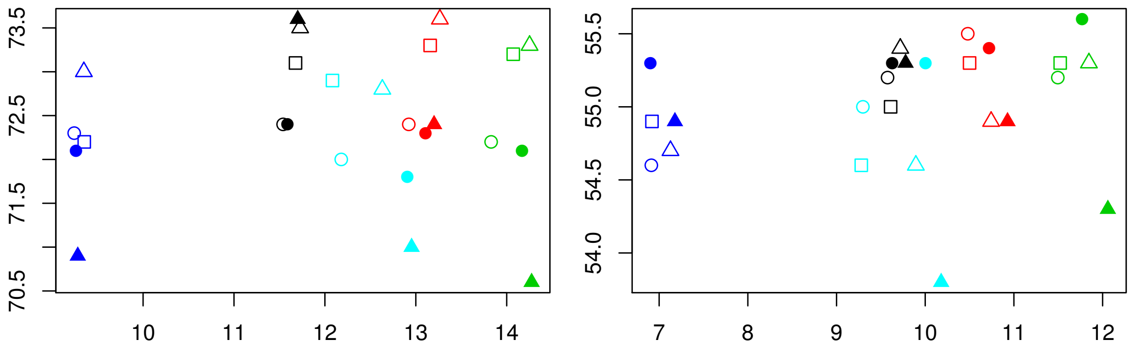


Figure 5.3: Model performance (y) by log number of non-zero global features (x). Propbank (left) and FrameNet (right). Global feature type by color: numArgs, roleCooc, argLoc, argLocRoleCooc, and full. **easyfirst** is triangle, **freq** is square, **rand** is circle. Filled in means dynamic, hollow is static.

model estimate. Our only explanation for the results is that **rand-static** is overfitting in a way which **rand-dynamic** isn't capable of.

Consistent with overfitting, we see that on both data sets **easyfirst-static** usually does as well or better than **easyfirst-dynamic**. In the opposite fashion of the random orderings, here the dynamic version is more expressive and likely to overfit.

5.3.5 Violation Fixing Perceptron (VFP)

At this point we have specified a transition system and features (both global and local), and all that is left is to choose a method for training the weights. Violation Fixing Perceptron (VFP) (Huang et al., 2012) is a family of perceptron updates which are intended to train machines which operate using beam search (greedy search being the trivial case of a beam with size one). The beam holds states, and at every step an action is appended to each state to reach a successor state which is put on the next beam.

In VFP, the core concept is a violation. A tuple (x, y, z), where x is a sentence as defined earlier and y is a string of correct actions (having zero cost/loss), and z is a string of predicted actions, is a *violation* if $\theta \cdot f(x, z) > \theta \cdot f(x, y)$ and z is "incorrect". There are multiple ways of defining incorrect which yield different algorithms in the VFP family. In all variants y and z must be the same length and if there is more than one incorrect (x, y, z), the one with the largest difference in score is chosen. In the early update variant, first described by Collins and Roark (2004), z is incorrect if it differs from y *only* in the *last* position. In max violation z is incorrect if it differs *any* position. In latest update z is incorrect if it differs in the *last* position (but can include other differences, unlike early update).

VFP Results In Figure 5.4 we plot the difference in performance between a model which includes a particular global feature type and the baseline model which only uses local features. Almost across the board the values are negative, indicating that the global model performs worse, even though the local model is nested within the global model (i.e. there exists a parameter setting in the global model such that it is equivalent to the local model). This result is at odds with previous results which have successfully used max violation perceptron to train models with non-local features. We hypothesize that the reason performance goes down is due to the expressivity of our global features and the inconsistency problem described in Chang et al. (2015).

Briefly, the inconsistency comes from the fact that the weights derived from VFP training simultaneously, and ambiguously, reflect what to do conditioned on being in a state arrived at by the *oracle* or the *predictor*. These two distributions over states are different if the predictor cannot perfectly mimic the oracle (the beam separability assumption). At

	Gold f		Auto f	
Global Feature	PB $\Delta\ell$	FN $\Delta\ell$	PB $\Delta\ell$	FN $\Delta\ell$
`numArgs`	-0.4	-0.1	-1.3	+0.3
`roleCooc`	-0.4	-0.3	-0.1	+0.6
`argLoc`	-1.2	-0.4	-1.9	+0.2
`roleCoocArgLoc`	-2.0	-0.2	0.0	+0.2
`full`	-1.5	-0.7	-2.0	+0.2

Figure 5.4: Global model advantage using max violation VFP and **freq**.

	Gold f		Auto f	
Global Feature	PB $\Delta\ell$	FN $\Delta\ell$	PB $\Delta\ell$	FN $\Delta\ell$
`numArgs`	0.0	0.0	+0.2	+0.7
`roleCooc`	-0.6	-0.3	-0.1	+0.5
`argLoc`	-0.4	+0.1	-0.1	-0.4
`roleCoocArgLoc`	+0.4	+0.4	+0.1	-0.1
`full`	+0.6	+0.4	-0.1	+0.3

Figure 5.5: Global model advantage using LOLS and **freq**.

test time, all of the states will be reached from the *predictor*'s actions, so the contribution of what to do by possibly incorrectly assuming the state/history was created by the *oracle* is misleading. This can be very bad when the global features are expressive and the predictor makes a significant number of mistakes.

VFP Inconsistency To validate that inconsistency is responsible for this poor performance, we setup another experiment where we artificially make the task easier. If the model is more accurate, then the predictor will necessarily be closer to the oracle, meaning that the inconsistency will shrink towards 0. To make the task easier, we added a binary feature to the local features which was either 1 or -1 based on whether the action has cost 0. We flip the sign of this feature with probability $1 - \alpha$. A model with $\alpha = 1$ should get perfect accuracy and $\alpha = \frac{1}{2}$ offers no extra information.

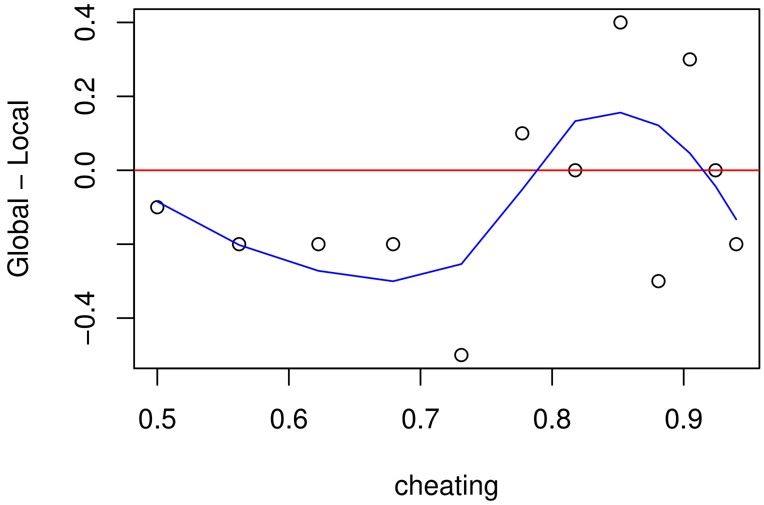


Figure 5.6: Benefit of `roleCooc` global features as a function of inconsistency in the model.

Figure 5.6 shows the difference between a global model using `roleCooc` and a local model (both receiving the "cheating" feature) for various values of α. This experiment used FrameNet data and max violation VFP. The local model does better than the global model (below the red line) where the inconsistency is high ($\alpha < 0.75$) and worse where it is low. Though the plot is noisy, when $\alpha = 1$ the two models have the same performance.

This result explains why max violation training has been shown to be successful in tasks like POS tagging and shift-reduce parsing, where the accuracy of the model is $> 90\%$. VFP with global features improves over local models on these tasks because the inconsistency is small, and the benefit from global features is great. In §5.3.7 we will return to this problem.

5.3.6 Locally Optimal Learning to Search (LOLS)

Learning to search (L2S) is a family of imitation learning algorithms including early update perceptron (Collins and Roark, 2004), LaSO (Daumé III and Marcu, 2005), SEARN (Daumé III et al., 2009), DAgger (Ross et al., 2011), and LOLS (Chang et al., 2015). The unifying feature of these algorithms is that they all are a reduction of training transition based models to a cost-sensitive classification problem over (s_t, a_t) pairs.

L2S methods are generally online algorithms which proceed by "rolling-in" to a state by running model which assigns scores to (s_t, a_t) pairs. For each state visited, a cost of taking each action is computed by "rolling-out", running another model to completion and observing the loss incurred by taking each action. The cost for every (s_t, a_t) pair is added to a running set of examples and the next iterate of the model is optimized to minimize the cost residuals.

Chang et al. (2015) showed that when the reference (oracle) policy is optimal, which we can guarantee in our case,[9] the oracle can be used during roll-outs to estimate costs, which can be computed in constant time. Given reference cost estimates, the only way to distinguish within this family is with respect to the roll-in distribution. The LOLS algorithm prescribes using the current policy for rolling-in, which does not always work well, which we will return to in §5.3.7.

LOLS Results In Figure 5.5 we plot global model advantage LOLS training. There are mixed results; some global features are actually improving over the local model (something which was not achieved by VFP training). We will return to why this is in §5.3.7, but first

[9] Every action fills in a label and we can say whether it is right or wrong, thus the reference policy is the one which always fills in a correct label.

we will analyze an orthogonal aspect of the model.

5.3.7 Error Analysis

Neither VFP nor LOLS worked for our transition transition systems out of the box. Here we discuss problems encountered with each algorithm and offer some solutions for fixing them. We do not claim these solutions are general, but hopefully offer insight into potential difficulties in training models like this.

VFP

The max violation version of VFP dictates that the violation to be corrected is the solution to

$$\underset{(x,y,z)\in C, z\in\bigcup_i\{\mathcal{B}_i[0]\}}{\operatorname{argmin}} \quad w_t \cdot \Delta\Phi(x, y, z) \tag{5.2}$$

Where $\mathcal{B}_i$ is the beam holding actions at step i^{th} and C is the beam confusion set as defined in (Huang et al., 2012). With only local features, Φ and $\Delta\Phi$ decompose into a sum over actions and and the argmin can be pushed inside that sum. This is equivalent to an (unstructured) perceptron update for every step in the trajectory. When global features are added, the update to the local features ceases to match the unstructured perceptron update and both global and local features are only updated with respect to a prefix of the oracle and predicted trajectory.

This prefix update may mean that mistakes at the end of the trajectory will not be corrected until the mistakes at the beginning are fixed.[10] Skipping training data puts

[10] This is the intended behavior under the beam separability assumption, but this may lead to very poor performance in general.

global models at a disadvantage over local ones, and we attribute the poor performance of the global models to this issue.

This problem was one of the motivations of max violation over the early update strategy introduced by Collins and Roark (2004). Huang et al. (2012) described an update called "latest update" which chooses the longest prefix which was still a violator, presumably to address the problem of skipping training data. While this may help, it is still possible to construct examples where a classification update would be made but a "latest update" would not.

For example, let $s(y_i) = w \cdot \phi(x, y_i)$ and $s(y_{[1:i]}) = w \cdot \phi(x, y_{[1:i]})$, such that $w \cdot \Delta\Phi(x, y_{[1:i]}, z_{[1:i]}) = s(y_{[1:i]}) - s(z_{[1:i]})$. Assume local scores $s(y_i)$ are derived from one-hot vectors indexed by (i, y_i). Assume a global model with the form: $s(y_{[1:i]}) = \sum_{k<j} w \cdot f(y_k, y_j) + \sum_j s(y_j)$ Take a sequence of binary decisions over the alphabet $\{a, b\}$ with mistakes at indices i and j such that $i < j$. Assume greedy search.

$$y_i = a, y_j = b, z_i = b, z_j = a$$

$$w \cdot f(b, a) = -3, w \cdot f(x, y) = 0 \ \forall (x, y) \neq (b, a)$$

$$s(y_i) = 0, s(y_j) = 0$$

$$s(y_{[1:i]}) = 0, s(y_{[1:j]}) = 0$$

$$s(z_i) = 1, s(z_j) = 1$$

$$s(z_{[1:i]}) = 1, s(z_{[1:j]}) = 1 + 1 + -3 = -1$$

$$s(y_{[1:j]}) > s(z_{[1:j]}) \Leftrightarrow \Delta\Phi(x, y_{[1:i]}, z_{[1:i]}) < 0$$

$(x, y_{[1:j]}, z_{[1:j]})$ is in the confusion set, but is not a violator, even though $y_j \neq z_j$, and the classification update would change $s(y_j)$ and $s(z_j)$.

	Gold f		Auto f	
Training	PB $\Delta\ell$	FN $\Delta\ell$	PB $\Delta\ell$	FN $\Delta\ell$
max violation	-3.5	-0.9	-1.3	-0.4
latest update	-1.4	-0.7	-1.4	-0.3
max violation +CLASS	-3.0	+1.8	-2.2	+2.4
latest update +CLASS	-2.4	+1.2	-2.4	+2.4

Figure 5.7: Global model advantage using `roleCooc` and **easyfirst-dynamic** across VFP variations and +CLASS.

Both max violation and latest update would choose to update on $(x, y_{[1:i]}, z_{[1:i]})$ in hopes of fixing it before moving on to the mistake at j. This happens consistently in our experiments (on the FrameNet data with `roleCooc`, by the end of training more than 10% of violators contain a mistake in the suffix not chosen by max violation).

Results In Figure 5.7 we show the performance of max violation and latest update variants of VFP along with an augmentation (+CLASS) intended to fix the issue of missing suffix mistakes. Global models were trained with the `roleCooc` feature template and **easyfirst-dynamic** action ordering. +CLASS adds an unstructured perceptron update for every index in the trajectory. This modification always helps on FrameNet, leading to global models which outperform local models, but consistently hurts on Propbank. Remember that all of these deltas are measured against a local only model, which is a pure CLASS update, so you can think of the +CLASS variants as a linear interpolation between a global and local objective.

LOLS

LOLS performs a roll-in with the current policy. This causes many updates which are derived from mistakes during frame identification. Once the wrong frame is predicted, in argument identification the model's cost incentives flip towards trying to predict $\varnothing$ for all roles so as not to incur false positives. The roles in FrameNet are defined based on the frame[11] and in Propbank they are not consistent across frames.[12] This is arguably a pathological property of a transition system: action costs strongly depend on state.

Using LOLS (model roll-in), there is a strong bias towards choosing $\varnothing$ for all roles, leading to high precision, low recall, and overall sub-optimal models. We found that when training the argument identification parameters of the model it was better to perform a hybrid model/oracle roll-in whereby the frame identification actions were chosen by the oracle. This may not be the fault of LOLS, but the of Hamming loss for action costs, which is a bad surrogate for F-measure.

Another important component of LOLs is the choice of cost in the cost-sensitive classification reduction. We found that defining costs based on the Hamming loss of an action performed very poorly. We found much better results with the multiclass hinge encoding described in Lee et al. (2004). In Figure 5.8 we show performance with various choices of roll-in and cost definitions. The best LOLS global models consistently improve over local models.

Absolute Performance

Throughout this work we have listed relative performance. Our absolute performance is 73.0 for Propbank (dev) and 55.3 for FrameNet (dev). This falls significantly short of the work of Zhou and Xu (2015) at 81.1 (PB dev), FitzGerald et al. (2015) at 79.2 (PB

[11] If you label a span as the Cognizer role for the frame Opinion and that span was the Cognizer role for the Judgment frame, then the label is wrong.

[12] with the exception of `ARG0` and `ARG1` which typically correspond to proto-Agent and proto-Patient roles.

		Gold f		Auto f	
Roll-in	Cost	PB $\Delta\ell$	FN $\Delta\ell$	PB $\Delta\ell$	FN $\Delta\ell$
model	Hamming	-24.5	-15.5	-10.1	-4.9
model	Hinge	-1.7	-1.1	-0.4	+0.2
hybrid	Hamming	-22.1	-12.9	-8.9	-1.0
hybrid	Hinge	+0.8	+1.0	+0.9	+1.1

Figure 5.8: Global model advantage using `roleCooc` and **easyfirst-dynamic** across LOLS variations: roll-in and cost function.

dev), and 72.0 (FN). Those works used non-linear neural models with multi-task distributed representations, which are not comparable to our results. However, the models of Pradhan et al. (2013) at 77.5 (PB test) and Das et al. (2012) at 64.6 (FN test) are roughly comparable, and the performance gap is still significant. While our efforts do not advance the state of the art in SRL, we hope that they are enlightening with respect to the application of various imitation learning methods.

5.4 Related Work

Berant and Liang (2015) used imitation learning for learning a semantic parser. Their prediction problem was parsing rather than SRL tagging, which they exploit with a manually designed grammar which greatly limited the space of trajectories. Their model also was not greedy, they used a chart-shaped beam instead of a linear beam or greedy inference. Their algorithm is probabilistic, similar to policy gradient (Sutton et al., 1999), compared to the methods used in this work which were discriminative (VFP) and cost-based (LOLS). While their methods are similar to LOLS, they did not explicitly describe the incentive-flipping behavior related to mistakes during model roll-in which we described in §5.3.7. It is possible that some of their tricks used during updates (i.e. history compression) minimized this as a problem. They may also have not had issues because their method is largely bottom up and uses a large beam in each cell which makes pruning mistakes significantly less likely.

Choi and Palmer (2011) explored transition based SRL and proposed some global features (e.g. copy `ARG0` from controlling predicates) but did not consider action re-ordering or imitation learning.

Wiseman and Rush (2016) derive a learning to search framework which is related to LaSO (Daumé III and Marcu, 2005). Similar to our hybrid roll-in, they "reset" the beam as soon as the oracle prefix falls off. This has the effect of preventing the z trajectory from deviating from the oracle too much, which may address the consistency with VFP. Their objective function includes a term for every index in the trajectory, similar to LOLS and distinct from VFP (which updates with respect to a prefix). Their work does not address (re-)ordering but does address beam search.

5.5 Conclusion

We started this chapter with a motivation for frame-based situation detection as a way of recognizing and understanding events, which plays a crucial role in report linking. We described some frame-based semantic theories and their associated resources for training statistical parsers. We then turned to the design and implementation of fast and efficient SRL parsers which we train with imitation learning methods. The parsers in this work don't do any backtracking and have to learn how much to trust the information captured in their history through global features. Other higher-order structured prediction methods have to either use a dynamic program (limiting the expressivity of their global features) or a large beam (slowing down inference) to search their way to better labelings. In this work we score our way to better labelings. Our contributions are a new method for doing hybrid roll-ins (related to pipeline training) which helps models with tight coupling between pairs of states learn more effectively (§5.3.7). We also show the conditions under which previous structured perceptron based methods will succeed and fail (§5.3.5 and §5.3.7). In §5.3.4 we describe a variety of action ordering techniques including frequency-based, random, left-to-right, and easy-first. We describe static and dynamic implementations of the last three and find that dynamic methods work better when you have more data and static versions otherwise. This Chapter therefore contributes a solution to identifying those predicates and arguments – situations, as defined initially – that may form the basic elements in a model for linking, which we turn to in the next chapter.

Chapter 6

Predicate Argument Alignment

6.1 Introduction

In the previous chapter we were concerned with identifying frames and roles in text which serve as a disambiguated latent form for natural language understanding. In this chapter we are interested in building models which can link mentions of equivalent latent forms in related documents. We call this task predicate argument linking (PAL) following Roth and Frank (2012). PAL is not about linking predicates to arguments (akin to semantic role labeling, see Chapter 5), but rather linking equivalent predicates and equivalent arguments mentioned in different discourses. The argument-linking half of PAL is related to, but more general than, cross document coreference resolution since arguments are not limited to restricted classes like entities (Bagga and Baldwin, 1998) or noun phrases (Soon et al., 2001), and can include pronouns, nominal phrases, or even situations ("[Playing fetch for hours]$_{arg}$ [tired out]$_{pred}$ [Fido]$_{arg}$"). The predicate-linking half of PAL is close to event coreference (Bagga and Baldwin, 1999)

Predicate argument linking is a central part of report linking because it allows users to take a claim in a report and find other mentions of it. This can be useful for verification (e.g. finding an equivalent claim stated by a more trustworthy source that is currently available (Dong et al., 2015)), finding new information (e.g. finding a source which discusses a claim and related details more extensively), or deduplication (e.g. hiding all claims which have already been read by the user (Dang and Owczarzak, 2008)).

We study predicate argument linking over a small collection of documents. We have discussed entity-based (Chapter 3) and relation-based (Chapter 4) methods for building these triage sets of related documents. In this chapter we use small document collections which were collected by using simple document clustering techniques, but this departure is useful in that it allows us to measure performance of the linking stages against other research and irrespective of the triage quality.

We treat the problem of predicate argument linking as decomposing over pairs of documents, similar to Roth and Frank (2012). Analyzing more than a pair of documents requires both more computation and nuanced representation theories for concepts. The pairwise framing of the problem allows models which are fast and discriminatively trained, allowing immediate progress on a task with immediate applications.

In this chapter we discuss two models for PAL. The first §6.3 poses linking as a classification problem and discusses what sorts of features are useful in a discriminatively trained linking model. The second §6.4 is a generalization of the first model which includes global factors which enforce coherence in the linking decisions made. In Figure 6.1 we give an illustration of the predicate argument linking task which can be used as a reference.

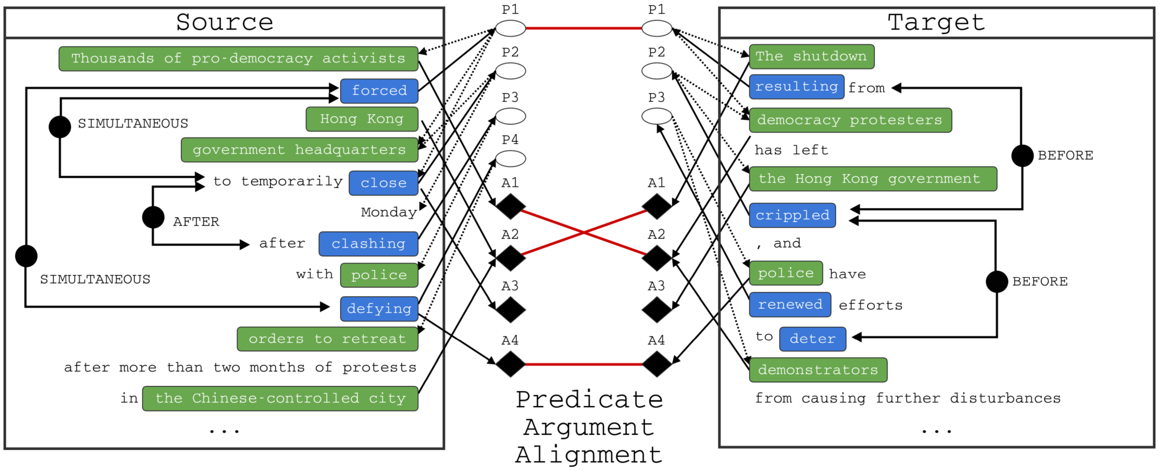


Figure 6.1: An example analysis and predicate argument alignment task between a source and target document. Predicates appear as hollow ovals, have blue mentions, and are aligned considering their arguments (dashed lines). Arguments, in black diamonds with green mentions, represent a document-level entity (coreference chain), and are aligned using their predicate structure and mention-level features. The alignment choices appear in the middle in red. Global information, such as temporal ordering, are listed as filled in circles and will be discussed in §6.4.

6.2 Resources

There are three datasets with predicate argument alignment annotations. In each case they do not include other annotations such as syntactic parses and named entity recognition, so we use the annotation tools described in Napoles et al. (2012) to add automatic labels. These datasets are all fairly small, less than 1000 documents each, and have predicate and argument mentions annotated with cluster labels indicating which links are valid. None of the datasets include any argument-binding labels such as semantic roles, grammatical function, or any other formal semantic representation. Though, we will see that considering such labels as latent variables can improve linking performance.

RF

- Australian [police]$_1$ have [arrested]$_2$ a man in the western city of Perth over an alleged [plot]$_3$ to bomb Israeli diplomatic buildings in the country , police and the suspect 's [lawyer]$_4$ [said]$_5$
- Federal [police]$_1$ have [arrested]$_2$ a man over an [alleged]$_6$ [plan]$_3$ to [bomb]$_7$ Israeli diplomatic [posts]$_8$ in Australia , the suspect 's [attorney]$_4$ [said]$_5$ Tuesday .

MTC

- As I [walked]$_1$ to the [veranda]$_2$ side , I [saw]$_2$ that a [tent]$_3$ is being decorated for [Mahfil-e-Naat]$_4$ -LRB- A [get-together]$_5$ in which the poetic lines in praise of Prophet Mohammad are recited -RRB-
- I [came]$_1$ towards the [balcony]$_2$, and while walking over there I [saw]$_2$ that a [camp]$_3$ was set up outside for the [Naatia]$_4$ [meeting]$_5$.

EECB

- [Gaetano Lo Presti]$_{15}$, one of [99 alleged Sicilian Mafia members]$_{17}$ [seized$_3$ on [Tuesday]$_{27}$, has apparently [hanged]$_1$ [himself]$_{15}$ with his [belt]$_{39}$ in [prison]$_{13}$.
- [A suspected Mafia leader]$_{15}$ committed [suicide]$_1$ overnight after being [arrested]$_3$ in [a major police sweep]$_5$, [police]$_{16}$ in [Palermo, Sicily]$_{21}$, [said]$_4$ on [Wednesday]$_{26}$.

Figure 6.2: Example pairs of sentences in aligned documents in the RF, MTC, and EECB corpora.

6.2.1 Extended Event Coreference Bank

The Extended Event Coreference Bank (EECB) which is based on the Event Coreference Bank of Bejan and Harabagiu (2010), with argument coreference annotations added by Lee et al. (2012). EECB structures the data into document clusters of articles from Google News, and only within these clusters can predicates and arguments corefer. Lee et al. (2012) considered clustering all mentions within a document cluster, but our methods work with pairs of documents, so we construct pairs by taking the first document in a cluster and creating a pair for every remaining document in the cluster. This yielded 340 document pairs. Since there is no train/test split, we evaluate on this data using 5-fold cross validation, and the scores reported are averages across these folds.

6.2.2 Roth and Frank

Roth and Frank (2012) (RF) annotated documents from the English Gigaword Fifth Edition corpus (Parker et al., 2011) for predicate argument alignments. Their goal was to align predicates in order to find null-instantiated arguments by projecting across alignments. Like EECB, they also used a document clustering technique based on headlines, which yields pairs of documents which are very similar. See Figure 6.2 for examples of how similar aligned sentences can be. The corpus is small with only 60 document pairs comprising the test set and 10 document pairs available for development or tuning.

6.2.3 Multiple Translation Corpora

We constructed a new predicate argument alignment dataset based on the LDC Multiple Translation Corpora (MTC),[1] which consist of multiple English translations for foreign news articles. Since the English translations should only differ in the words and phrases used while remaining true to the meaning in the source language, they provide a good resource for aligned predicate argument pairs in the target language. Other corpora like RF contain many aligned documents which are very lexically similar to each other, possibly stemming from a common news service article. We created the MTC corpus to test whether systems can recognize alignments in the face of lexical diversity, so we selected document pairs from the multiple translations that minimize the lexical overlap in English. The sentences in the MTC data are already aligned, and we use GIZA++ to determine token-level alignments within each sentence. We take all aligned nouns as arguments and all

[1] LDC2010T10 LDC2010T11 LDC2010T12 LDC2010T14 LDC2010T17 LDC2010T23 LDC2002T01 LDC2003T18 LDC2005T05

aligned verbs (excluding be-verbs, light verbs, and reporting verbs) as predicates. We then add negative examples by randomly substituting half of the sentences in one document with unrelated sentences from another corpus, introducing negative alignments. The amount of substitutions we perform can vary the "relatedness" of the two documents in terms of the predicates and arguments that they talk about. This reflects our expectation of real world data, where we do not expect perfect overlap in predicates and arguments between a source and target document, as you would in translation data.

Lastly, we prune any document pairs that have more than 80 predicates or arguments or have a Jaccard index of lemmas greater than 0.5, to give us a dataset of similar size to the EECB: 328 pairs.

6.3 Feature-rich Models of Alignment

Our first investigation is into whether discriminative classifier-based method are sufficient for predicate argument linking. And to the extent that they are, what features are predictive? We consider pairs of documents from the three corpora described in the last section and report on the accuracy based on a rich feature set drawing on a variety of semantic resources.

6.3.1 PARMA

PARMA (Predicate ARguMent Aligner) is the name of the classifier-based aligner in this section. It considers predicates as singletons within a document but arguments are represented as coreference chains inferred using the Stanford coreference resolver (Lee et al., 2011). For argument features which require a single mention, we select a canonical mention from coreference chains which has the highest score: the number of NNP words minus the number of mentions preceding it in the document, with ties going to the earlier mention.

Notation We refer to a predicate or an argument as an "item" with type *predicate* or *argument*. An alignment between two documents is a subset of all pairs of "items" in either documents with the same type[2]. We call the two documents being aligned the source document S and the target document T. Items are referred to by their index, and z_{ij} is a binary variable representing an alignment between item i in S and item j in T. A full alignment is an assignment $\mathbf{z} \in \{0,1\}^{|S|\times|T|}$. z_{ij} is a binary variable that is 1 for an alignment between item i in S and item j in T and 0 otherwise.

We train a logistic regression model for $p(z \mid x)$ and maximize the likelihood of a document alignment under the assumption that the item alignments are independent.

$$
\begin{aligned}
p(z \mid x) &= \prod_{ij} p(z_{ij} \mid x) && (6.1)\\
&= \prod_{ij} \frac{1}{1+\exp(-w \cdot f(z_{ij}, x))} && (6.2)
\end{aligned}
$$

The set of training data is a set of aligned documents $(z, x) \in D$. We optimize the log-likelihood of the alignments with an L_1 regularization term:

$$J(w) = \mathrm{E}_{(z,x)\in D} \log p(z \mid x) + \lambda ||w||_1 \quad (6.3)$$

After learning model parameters w we optimize F_1 by introducing a threshold τ on the

[2] Note that type is not the same thing as part of speech: we allow nominal predicates like "death".

probability for alignments. This two-step process lets us tune for either precision or recall and usually leads to a higher F_1 than by simply using the 50% probability threshold typically used in logistic regression.

6.3.2 Features

The focus of PARMA is the integration of a diverse range of features based on existing lexical semantic resources. We built PARMA on a supervised framework to take advantage of this wide variety of quality features since they can describe many different correlated aspects of generation. Our features cover the spectrum from high-precision (e.g. TED alignments) to high-recall (e.g. PPDB lexical rules). Each feature has access to the proposed argument or predicate spans to be linked and the containing sentences as context. While we use supervised learning, some of the existing datasets for this task are very small. For extra training data, we pool material from different datasets and use the multi-domain split feature space approach to learn dataset specific behaviors (Daumé, 2007).

Features in general are defined over mention spans, but we take the product of these features with the part of speech tag of the head word, which often corresponds to separate feature-spaces for predicates and arguments.

PPDB We use lexical features from the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013). PPDB is a large set of paraphrases extracted from bilingual corpora using pivoting techniques. We make use of the English lexical portion which contains over 7 million rules for rewriting terms like “planet” and “earth”. PPDB offers a variety of conditional probabilities for each (synchronous context free grammar) rule, which we treat as independent experts. For each of these rule probabilities (experts), we find all rules that match the head tokens of a given alignment and have a feature for the max and harmonic mean of the log probabilities of the resulting rule set.

FrameNet FrameNet is a lexical database based on Charles Fillmore's Frame Semantics (Fillmore, 1982; Baker et al., 1998). The resource is organized around semantic frames that can be thought of as descriptions of events (see §5.2.1 for more details). Frames crucially include specification of the participants, or frame elements, in the event. The Destroy frame for instance includes frame elements Destroyer or Cause (which differ in sentience) and Undergoer. Frames are related to other frames through inheritance and perspectivization. For instance the frames Commerce_buy and Commerce_sell (with respective lexical realizations "buy" and "sell") are both perspectives of Commerce_goods-transfer (no lexical realizations) which inherits from Transfer (with lexical realization "transfer").

We compute a shortest path between headwords given edges (hypernym, hyponym, perspectivized parent and child) in FrameNet and bucket by distance to get features. We also have a binary feature for whether two tokens evoke the same frame.

TED Alignments Given two predicates or arguments in two sentences, we attempt to align the two sentences using a Tree Edit Distance (TED) model that aligns two dependency trees. We represent a node in a dependency tree with three fields: lemma, POS tag and the type of dependency relation to the node's parent. The TED model aligns one tree with the other using the dynamic programming algorithm of Zhang and Shasha (1989) with three predefined edits: deletion, insertion and substitution, seeking a solution yielding the minimum edit cost. Once we have built a tree alignment, we extract features for 1) whether the heads of the two phrases are aligned and 2) the count of how many tokens are aligned in both trees. This type of feature has been shown to be very useful for tasks like question answering (Yao et al., 2013).

WordNet WordNet (Miller, 1995) is a database of information (synonyms, hypernyms, etc.) pertaining to words and short phrases. For each entry, WordNet provides a set of synonyms, hypernyms, etc. Given two spans, we use WordNet to determine semantic similarity by measuring how many synonym (or other) edges are needed to link two terms. Similar words will have a short distance. For features, we find the shortest path linking the head words of two mentions using synonym, hypernym, hyponym, meronym, and holonym edges and bucket the length.

String Transducer To represent similarity between arguments that are names, we use a stochastic edit distance model. This stochastic string-to-string transducer has latent "edit" and "no edit" regions where the latent regions allow the model to assign high probability to contiguous regions of edits (or no edits), which are typical between variations of person names. In an edit region, parameters govern the relative probability of insertion, deletion, substitution, and copy operations. We use the transducer model of Andrews et al. (2012). Since in-domain name pairs were not available, we instead applied the unsupervised model of Andrews et al. to a corpus of 5,000 Wikipedia redirects to estimate the transducer parameters. For a pair of mention spans, we compute the edit cost and bucket to get features. We duplicate these features with copies that only fire if both mentions are tagged

		F_1	Precision	Recall
EECB	lemma	63.5	84.8	50.8
	PARMA	**74.3**	80.5	69.0
RF	lemma	48.3	40.3	60.3
	Roth and Frank (2012)	54.8	59.7	50.7
	PARMA	57.6	52.4	64.0
	Roth (2013)	**58.2**	71.8	48.9
MTC	lemma	42.1	51.3	35.7
	PARMA	**59.2**	73.4	49.6

Table 6.1: Results on each of the datasets.

as `PER`, `ORG` or `LOC`.

6.3.3 Experiments

Metric We use precision, recall and F_1. For RF, we follow Roth and Frank (2012) and Cohn et al. (2008) and evaluate on a version of F_1 that considers SURE and POSSIBLE links, which are available in the RF data (but not EECB or MTC, where we treat all alignments as SURE). Given an alignment to be scored A and a reference alignment B which contains SURE and POSSIBLE links, $B_s \subseteq B_p$ respectively, precision and recall are:

$$P = \frac{|A \cap B_p|}{|A|} \qquad R = \frac{|A \cap B_s|}{|B_s|} \tag{6.4}$$

and F_1 as the harmonic mean of the two. Results for EECB and MTC reflect 5-fold cross validation, and RF uses the given dev/test split.

Lemma baseline We include a lemma baseline, in which two predicates or arguments align if they have the same lemma[3].

[3] We could not reproduce lemma from Roth and Frank (2012) (shown in Table 6.1) due to a difference in lemmatizers. We obtained 55.4; better than their system but worse than PARMA.

For the Roth and Frankdataset, we train PARMAwith 150 random examples from each of EECB and MTC, and the entire dev set from Roth and Frankusing multi-domain feature splitting. We also tune the threshold τ on the dev set, but choose the regularizer λ on the EECB experiments.

Results On every dataset PARMA significantly improves over the lemma baselines (Table 6.1). Our model performs better than Roth and Frank (2012) but worse than Roth (2013), who added a few more features to the model of Roth and Frank (2012). We also note that compared to Roth and Frank (2012) and Roth (2013) we obtain higher recall but lower precision, which reflects their goals of finding high precision/clean links for downstream tasks, versus our approach which optimizes for F_1.[4]

We observe that MTC was more challenging than the other datasets, at least as measured by the lemma baseline. Figure 6.3 shows the correlation between document similarity and alignment F_1 score for the lemma baseline on the RF and MTC data sets. For the RF dataset the more similar the document pairs were the more likely lemma matching was going to work well. This correlation means that by selecting document pairs with high lexical overlap is akin to choosing the easiest documents to link predicates and arguments on, where naive assumptions like lemma matching are likely to work. Our constructed MTC corpus does not exhibit this behavior. There are document pairs which are similar and some which are diverse (note, much more diverse than the RF documents, which all have a similarity of around 0.55 or higher), but the fact that they are similar does not mean

[4] In Chapter 4 we said that F_1 gave too much credit for recall and that precision was more important. This statement was made with respect to distant supervision, where perfect recall is a false goal (likely unattainable by any system given that some facts will not be attested to). In the case of this information extraction problem, perfect recall is an attainable goal, and should be rewarded.

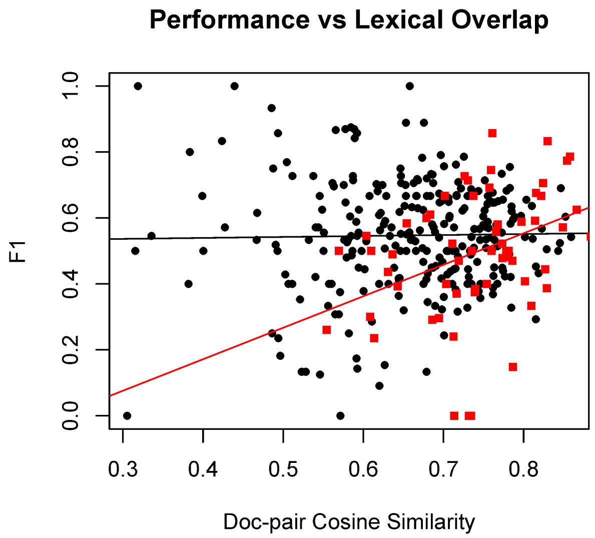


Figure 6.3: F_1 on RF (red squares) is correlated with document pair cosine similarity but with MTC (black circles) this is not the case.

that they are easy. The fact that we see the gap between PARMA and lemma matching open up on MTC vs RF means that PARMA does well not because of easy examples.

6.4 Structured Models of Alignment

In the last section we described an *unstructured* model for predicate argument alignment, meaning that the model treats every alignment as an independent decision. This is a naive assumption and in this chapter we address it by introducing structured factors which ensure that the alignment decisions are globally coherent.

6.4.1 Model

The classification framework used in the previous section has advantages: it's fast since individual decisions can be made independently, but it comes at the cost of potential incoherence in the linking decisions. The result may be links that conflict in their interpretation of the document. Figure 6.1 shows that there is a significant amount of latent structure (everything except the red links) involved in predicate argument linking, and independent models may predict alignments which are not consistent with this latent structure. In §6.4.2 we will discuss some of the inconsistencies which can occur and how to parameterize our scoring function to avoid them.

We can make a crude analogy between predicate argument alignment and word alignment for machine translation (MT). MT alignment models like IBM model 2 (Brown et al., 1993) decompose the probability of an alignment into two terms, a translation probability (e.g. $p(e_i = \text{“cat”} \mid f_j = \text{“chat”})$) and a distortion probability of where in a sentence a given word is generated from conditioned on the previous generation. The analogy carries over for the translation term where we model the probability that a predicate or argument is paraphrased in a particular way (this score is described in §6.3). But the distortion probability differs between MT and predicate argument alignment in that we don't expect any sort of monotonicity or other word ordering effects in our predicate or argument alignments.

Lacoste-Julien et al. (2006) introduced a discriminative alignment model which models the pairwise interactions between two alignments decisions z_{ab} and z_{xy}. When the alignment model is stated generally like this we can re-interpret the meaning of the quadratic factors and parameterize them however we like, which we will get into in the next section. Inference in this type of quadratic model is an instance of the Quadratic Assignment Problem (QAP), which is NP-hard in general, but very fast to solve for the size of problems we are concerned with.

We first introduce some notation for our model and then define the quadratic factors (or joint factors) which encourage predicate argument alignments which have global coherence. Following the description of factors we will describe inference and learning §6.4.3 followed by experiments §6.4.4.

Notation We say that a source document S and target document T together form an instance x, for which we would like to produce an alignment matrix (rows index items in the source and columns in the target) $\mathbf{z} \in \{0,1\}^{|S| \times |T|}$. $z_{ij} = 1$ indicates that item i in the source and item j in the target are aligned and in some cases we will explicitly indicate whether the items are predicates (z^p_{ij}) or argument (z^a_{ij}).

For each pair of items we use *local* feature functions $\mathbf{f}(z_{ij}, x)$ and corresponding parameters w, which capture the similarity between two items without the context of other alignments. These are the same local features as described in §6.3.

$$g_{local}(z) = \sum_{ij} s_{ij} = w \cdot \mathbf{f}(z_{ij}, x) \tag{6.5}$$

where s_{ij} is the score of linking items i and j.

Using only local features, our system would greedily select alignments. To capture global aspects we add joint factors that capture effects between alignment variables. Each joint factor ϕ is comprised of a constrained binary variable z_ϕ associated with features $\mathbf{f}(\phi)$

that indicates when the factor is active. Together with parameters w these form additional scores s_ϕ for the objective:

$$s_\phi = w \cdot \mathbf{f}(\phi) \tag{6.6}$$

The full linear scoring function on alignments sums over both local similarity and joint factors:

$$g_{global}(z) = \sum_{ij} s_{ij} z_{ij} + \sum_{\phi \in \Phi} s_\phi z_\phi. \tag{6.7}$$

6.4.2 Joint Factors

Our goal is to develop joint factors that improve over the feature rich local factors baseline by considering global information.

Fertility A common mistake when making independent classification decisions is to align many source items to a single target item. While each link looks promising on its own, they clearly cannot all be right. Empirically, the training set reveals that many to one alignments are uncommon; thus many to one predictions are likely errors. We add a fertility factor for predicates and arguments, where fertility is defined as the number of links to an item. Higher fertilities are undesired and are thus penalized. Formally, for matrix $\mathbf{z}$, the fertility of a row i or column j is the sum of that row or column.

We include two types of fertility factors. The first factor distinguishes between rows with at least one link from those with none. For every row and column we add an auxiliary variable z_i^{fert1}, which indicates if the fertility of a given row is at least 1, and a global factor ϕ_i^{fert1}. For the auxiliary variable, we add constraints relating it to the primary alignment variables:

$$z_i^{fert1} = \max_j z_{ij} \tag{6.8}$$

The factor is parameterized by weights and some simple features: an indicator for the type of alignment (predicate or argument), and log-sized bucketed features for the length of the row or column.

The second fertility factory considers items with a fertility greater than one, penalizing items for having too many links. These ϕ_i^{fert2} factors have the same type of parameterization (different weights, same features), but the auxiliary variables have a different set of constraints:

$$z_i^{fert2} \geq z_{ij} \cdot z_{ik} \; \forall j < k \tag{6.9}$$

$$z_i^{fert2} \leq z_{ij} \; \forall j \tag{6.10}$$

This factor penalizes rows and columns that have fertility of at least two, but does not distinguish beyond that. An alternative would be to introduce a factor for every pair of variables in a row, each with one constraint. This would heavily penalize fertilities greater than two. We found that the resulting quadratic program took longer to solve and gave worse results.

Since documents have been processed to identify in-document coreference chains, we do not expect multiple arguments from a source document to align to a single target item. For this reason, we expect ϕ^{fert2} for arguments to have a large negative weight. In contrast, since predicates do not form chains, we may have multiple source predicates for one target.

We note an important difference between our fertility factor compared with Lacoste-Julien et al. (2006). We parameterize fertility for only two cases (1 and 2) whereas they consider fertility factors from 2 to k. We do not parameterize fertilities higher than two because they are not common in our dataset and come at a high computational cost.

Predicate Argument Structure The predicate argument structure (PAS), i.e. which arguments bind to which predicates in either document, provides a structured way to view the linking decisions made by our model. We infer predicate argument structure by checking whether there is a short (no more than 3 edges) syntactic dependency path between a predicate and argument. This is similar to using a semantic role labeler like the one described in Chapter 5, but where the system is constrained by provided argument spans and where the roles are dropped.[5]

Predicate-centric We start with the predicate-centric factor ϕ_{ij}^{psa}, which is instantiated once for every predicate alignment variable z_{ij}^{p}. Ideally, two predicates can only align when they share the same arguments (e.g. "John $[\text{ate}]_i$ pizza" doesn't align to "John $[\text{ate}]_j$ guacamole"). However, in practice we may incorrectly resolve argument links, or there may be implicit arguments that do not appear as syntactic dependencies of the predicate trigger. Therefore, we settle for a weaker condition, that there should be *some* overlap in the arguments of two coreferent predicates.

For every predicate alignment z_{ij}^{p}, we add a factor ϕ_{ij}^{psa} whose score s_{ij}^{psa} the re-

[5] The factors we list here do not depend on roles so as to make as little assumptions as possible and be robust to role classification errors.

ward for having some argument overlap; predicates share arguments (PSA). We introduce auxiliary variables z_{ij}^{psa} and z_{ij}^{arg} which track whether the PAS for z_{ij}^{p} satisfies the at least one shared argument condition:

$$z_{arg(ij)} = \max_{\substack{k \in \text{args}(i) \\ l \in \text{args}(j)}} z_{kl}^{a} \tag{6.11}$$

$$z_{ij}^{psa} \geq z_{ij}^{p} \cdot z_{arg(ij)} \tag{6.12}$$

$$z_{ij}^{psa} \leq z_{ij}^{p} \tag{6.13}$$

$$z_{ij}^{psa} \leq z_{arg(ij)} \tag{6.14}$$

where args(i) finds the indices of all arguments governed by the predicate i. The features for this factor $\mathbf{f}(\phi_{ij}^{psa}, x)$ have an intercept, an indicators for $\min_{p \in \{i,j\}} |\text{args}(p)|$ and $\max_{p \in \{i,j\}} |\text{args}(p)|$.

Entity-centric We expect a similar type of behavior from arguments, or at least the subset which tend to be aligned which are usually named entities. If an entity appears in two documents, it is likely that this entity will be mentioned in the context of a common predicate, i.e. arguments share predicates (ASP). For a given argument alignment z_{ij}^{a} we add a factor ϕ_{ij}^{asp} and the auxiliary variables z_{ij}^{asp} and $z_{pred(ij)}$ needed to encode the ASP semantics:

$$z_{pred(ij)} = \max_{\substack{k \in \text{preds}(i) \\ l \in \text{preds}(j)}} z^a_{kl} \tag{6.15}$$

$$z^{asp}_{ij} \geq z^a_{ij} \cdot z_{pred(ij)} \tag{6.16}$$

$$z^{asp}_{ij} \leq z^a_{ij} \tag{6.17}$$

$$z^{asp}_{ij} \leq z_{pred(ij)} \tag{6.18}$$

where preds(i) returns all predicates which bind some mention of argument i in a coreference chain. The features $f(\phi^{asp}_{ij})$ are the same as $f(\phi^{psa}_{ij})$ but with $|\text{args}(p)|$ replaced with $|\text{preds}(p)|$.

Temporal Information In two documents which discuss a common set of events, as long as the authors have the same perception of what occurred, the order of events in time that they report should be the same. Any alignments which violate the common order of shared events should be penalized. Determining the temporal order of events described in text was the subject of a SemEval 2013 task (UzZaman et al., 2013). Many systems produce partial relations of events in a document based on lexical aspect and tense, as well as discourse connectives like “during” or “after”. We obtain temporal relations with CAEVO, a state-of-the-art sieve-based system (Chambers et al., 2014), and use these orderings to place soft constraints on some of our predicate alignments which correspond to events.

TimeML (Pustejovsky et al., 2003), the format for specifying temporal relations, defines relations between events (e.g. *immediately before* and *simultaneous*), each with an inverse (e.g. *immediately after* and *simultaneous* respectively). We will refer to a temporal relation as R and its inverse as R^{-1}. Suppose we had two event predicates p_a and p_b in the source, and two other event predicates p_x and p_y in the target. If we observed $p_a R_1 p_b$ and $p_x R_2 p_y$, the following alignments conflict with the in-doc relations:[6]

z_{ax}	z_{by}	z_{ay}	z_{bx}	In-Doc Relations
*	*	1	1	$R_1 = R_2$
1	1	*	*	$R_1 = R_2^{-1}$

where 1 means there is a link and * means there is a link or no link (wildcard). The simplest example that fits this pattern is: ‘a before b’, ‘x before y’, ‘a corefers with y’, and ‘b corefers with x’ implies a conflict.

We introduce a factor that penalizes these conflicting configurations. For every temporal ordering conflict we observe which matches that pattern in the table above, we

[6] We exclude *simultaneous* from this list of conflicts because it does not lead to the type of temporal ordering conflicts as the other relations.

instantiate an auxiliary variable z_{abxy}^{temp} with the following constraints:

$$
\begin{aligned}
z_{abxy}^{temp} &\geq z_{ay} \cdot z_{bx} \\
z_{abxy}^{temp} &\leq z_{ay} \\
z_{abxy}^{temp} &\leq z_{bx} \\
&\quad \text{if } p_a R_1 p_b, p_x R_2 p_y, R_1 = R_2 \\
z_{abxy}^{temp} &\geq z_{ax} \cdot z_{by} \\
z_{abxy}^{temp} &\leq z_{ax} \\
z_{abxy}^{temp} &\leq z_{by} \\
&\quad \text{if } p_a R_1 p_b, p_x R_2 p_y, R_1 = R_2^{-1}
\end{aligned}
\tag{6.19}
$$

Thus $s_{\phi_{abxy}^{temp}}$ is the cost of disagreeing with the in-doc temporal relations.

Since CAEVO gives each relation prediction a probability, we incorporate this into the feature by indicating the probability of a conflict *not* arising:

$$\mathbf{f}(\phi^{temp}) = \log\big(1 - p(R_1)p(R_2) + \epsilon\big) \tag{6.20}$$

ϵ avoids large negative values since CAEVO probabilities are not perfectly calibrated. We use $\epsilon = 0.1$, allowing feature values of at most -2.3.

Summary The objective is a linear function over binary variables. There is a local similarity score coefficient on every alignment variable, and a joint factor similarity score on every quadratic variable. These quadratic variables are constrained by products of the original

```
def train(alignments):
  w = init_weights()
  working_set = set()
  while True:
    xi = solve_ILP(w, working_set)
    c = most_violated_constraint(w, alignments)
    working_set.add(c)
    if hinge(c, w) < xi:
      break

def most_violated_constraint(w, alignments):
  delta_features = vector()
  loss = 0
  for z in alignments:
    z_mv = make_ILP(z)
    for phi in factors:
      costs = dot(w, phi.features)
      z_mv.add_terms(costs, phi.vars)
      z_mv.add_constraints(phi.constraints)
    solve_ILP(z_mv)
    mu = (z.size + k) / (avg_z_size + k)
    delta_features += mu * (f(z) - f(z_mv))
    loss += mu * Delta(z, z_mv)
  return Constraint(delta_features, loss)

def hinge(c, w):
  return max(0, c.loss - dot(w, c.delta_features))
```

Figure 6.4: Learning algorithm (caching and ILP solver not shown). The sum in each constraint is performed once when finding the constraint, and implicitly thereafter.

alignment variables. Decoding an alignment requires solving this quadratically constrained integer program; in practice is can be solved quickly without relations.

6.4.3 Inference

Learning We use the supervised structured SVM formulation of Joachims et al. (2009). As is common in structure prediction we use margin rescaling and 1 slack variable, with the structural SVM objective:

$$
\begin{gathered}
\min_{w} ||w||_2^2 + C\xi \\
\text{s.t. } \xi \geq 0 \\
\xi + \sum_{i=1}^{N} w \cdot f(z_i) \geq \sum_{i=1}^{N} w \cdot f(\hat{z_i}) + \Delta(z_i, \hat{z_i}) \\
\forall \hat{z_i} \in \mathcal{Z}_i
\end{gathered}
\tag{6.21}
$$

where $\mathcal{Z}_i$ is the set of all possible alignments that have the same shape as z_i.

The score function for an alignment uses three types of terms: weights, features, and alignment variables. When we decode, we take the product of the weights and the features to get the costs for the ILP (e.g. $s_\phi = w \cdot \mathbf{f}(\phi)$). When we optimize our SVM objective, we take the product of the alignment variables and the features to get modified features for the SVM:

$$f(z) = \sum_{ij} z_{ij} \mathbf{f}(z_{ij}) + \sum_{\phi \in \Phi} z_\phi \mathbf{f}(\phi) \tag{6.22}$$

Since we cannot iterate over the exponentially many margin constraints, we solve for this optimization using the cutting-plane learning algorithm. This algorithm repeatedly asks the "separation oracle" for the most violated SVM constraint, which finds this constraint by solving:

$$\arg \max_{\hat{z_1} \ldots \hat{z_N}} \sum_{i} w \cdot f(\hat{z_i}) + \Delta(z_i, \hat{z_i}) \tag{6.23}$$

subject to the constraints defined by the joint factors. When the separation oracle returns a constraint that is not violated or is already in the working set, then we have a guarantee that we solved the original SVM problem with exponentially many constraints. This is the most time-consuming aspect of learning, but since the problem decomposes over document alignments, we cache solutions on a per document alignment basis. With caching, we only call the separation oracle around 100-300 times.

We implement the separation oracle using an ILP solver, CPLEX,[7] due to complexity of the discrete optimization problem: there are 2^{m^n} possible alignments for and $m \times n$ alignment grid. In practice this is solved very efficiently, taking less than a third of a second per document alignment on average. We would like Δ to be F_1, but we need a decomposable loss to include it in a linear objective (Taskar et al., 2003). Instead, we use Hamming loss as a surrogate, as in Lacoste-Julien et al. (2006).

Our training data is heavily biased towards negative examples, performing poorly on F_1 since precision and recall are unbalanced. We use an asymmetric version of Hamming loss that incurs c_{FP} cost for predicting an alignment for two unaligned items and c_{FN} for predicting no alignment for two aligned items. We fixed $c_{FP} = 1$ and tuned $c_{FN} \in \{1, 2, 3, 4\}$ on dev data. Additionally we found it useful to tune the scale of the loss function across $\{\frac{1}{2}, 1, 2, 4\}$. Previous work, such as Joachims et al. (2009), use a hand-chosen constant for the scale of the Hamming loss, but we observe some sensitivity in this parameter and choose to optimize it.

Decoding We tune the threshold for classification τ on dev data to maximize F_1 (via linesearch). For SVMs τ is typically fixed at 0: this is not necessarily good practice when your training loss differs from test loss (Hamming vs F_1). In our case this extra parameter is worth allocating a portion of training data to enable tuning. Tuning τ addresses the same

[7] `http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/`

problem as using an asymmetric Hamming loss, but we found that doing both led to better results.[8] Since we are using a global scoring function rather than a set of classifications, τ is implemented as a test-time unary factor on every alignment.

6.4.4 Experiments

Due to the small data size, we use k-fold cross validation for both datasets. We choose $k = 10$ for RF due to its very small size (more folds give more training examples) and $k = 5$ on EECB to save computation time (amount of training data in EECB is less of a concern). Hyperparameters were chosen by hand using using cross validation on the EECB dataset using F_1 as the criteria (rather than Hamming). Figures report averages across these folds.

Due to the lack of annotated arguments on the Roth and Frankdataset, we can only report predicate linking performance and the PSA and ASP factors do not apply.

Systems Following Roth and Frank (2012) we include a *Lemma* baseline for identifying alignments which will align any two predicates or arguments that have the same lemmatized head word.[9] The *Local* baseline uses the same features as Wolfe et al., but none of our joint factors. In addition to running our joint model with all factors, we measure the efficacy of each individual factor by evaluating each with the local features.

For evaluation we use a generous version of F_1 that is defined for alignment labels

[8] Only tuning τ performed almost as well as tuning τ and the Hamming loss, but not tuning τ performed much worse than only tuning the Hamming loss at train time.

[9] The lemma baseline is obviously sensitive to the lemmatizer used. We used the Stanford CoreNLP lemmatizer (Manning et al., 2014) and found it yielded slightly better results than previously reported as the lemma baseline (Roth and Frank, 2012), so we used it for all systems to ensure fairness and that the baseline is as strong as it could be.

composed of sure, G_s, and possible links, G_p and the system's proposed links H (following Cohn et al. (2008) and Roth and Frank (2012)).

$$P = \frac{|H \cap G_p|}{|H|} \quad R = \frac{|H \cap G_s|}{|G_s|} F = \frac{2PR}{P+R}$$

Note that the EECB data does not have a sure and possible distinction, so $G_s = G_p$, resulting in standard F_1. In addition to F_1, we separately measure predicate and argument F_1 to demonstrate where our model makes the largest improvements.

We performed a one-sided paired-bootstrap test where the null hypothesis was that the joint model was no better than the *Local* baseline (described in Koehn (2004)). Cases where $p < 0.05$ are bolded.

	F_1	P	R	Arg F_1	Arg P	Arg R	Pred F_1	Pred P	Pred R
Lemma	68.1	79.3 *	59.6	61.7	79.1 *	50.6	75.0	87.3 *	65.7
Local	73.0	75.8	**70.5**	67.7	76.3	**60.8**	78.7	81.4	76.2
+Fertility	77.1 *	83.9 *	71.3	66.6	80.9 *	56.6	82.8 *	87.4 *	**78.7** *
+Predicate-centric	74.1 *	80.7 *	68.6	67.4	81.6 *	57.3	79.7 *	85.0 *	75.1
+Argument-centric	73.7	81.2 *	67.5	66.8	**83.0** *	55.9	79.3	85.1 *	74.3
+Temporal	73.7	78.2 *	69.7	**67.9**	80.6 *	58.7	79.0	82.1	76.1
+All Factors	**77.5** *	**86.3** *	70.3	65.8	83.1 *	54.5	**83.7** *	**89.7** *	78.4 *

Table 6.2: 5-fold cross validation performance on EECB (Lee et al., 2012). Statistically significant ($p < 0.05$ using a one-sided paired-bootstrap test) improvements from Local are bolded.

Results Results for EECB and RF are reported in Table 6.3. As previously reported, using just local factors (features on pairs) improves over lemma baselines. The joint factors make statistically significant gains over local factors in almost all experiments. Fertility factors provide the largest improvements from any single constraint. A fertility penalty actually allows the pairwise weights to be more optimistic in that they can predict more

	Pred F_1	Pred P	Pred R
Lemma	52.4	47.6	58.2 *
Local	58.1	63.5	53.6
Roth and Frank (2012)	54.8	59.7	50.7
Roth (2013)	58.2	71.8	48.9
+Fertility	**60.0**	57.4	**62.4** *
+Predicate-centric	NA	NA	NA
+Argument-centric	NA	NA	NA
+Temporal	59.0	57.4	60.6 *
+All factors	59.4	56.9	62.2 *

Table 6.3: Cross validation results for RF (Roth and Frank, 2012). Statistically significant improvements from Local marked * ($p < 0.05$ using a one-sided paired-bootstrap test) and best results are bolded.

alignments for reasonable pairs, allowing the fertility penalty to ensure only the best is chosen. This penalty also prevents the "garbage collecting" effect that arises for instances that have rare features (Brown et al., 1993).

Temporal constraints are relatively sparse, appearing just 2.8 times on average. Nevertheless, it was very helpful across all experiments, though only statistically significantly on the RF dataset. This is one of the first results to demonstrate benefits of temporal relations affecting an downstream task. Perhaps surprisingly, these improvements result from a a temporal relation system that has relatively poor absolute performance. Despite this, improvements are possibly due to the orthogonal nature of temporal information; no other feature captures this signal. This suggests that future work on temporal relation prediction may yield further improvements and deserves more attention as a useful feature for semantic tasks in NLP.

The predicate-centric factors improved performance significantly on both datasets. For the predicate-centric factor, when a predicate was aligned there is a 72.3% chance that there was at least one argument aligned as well, compared to only 14.1% of case of non-aligned predicates. As mentioned before, the reason the former number isn't 100% is primarily due to implicit arguments and errors in argument identification. The argument-centric features helped almost as much as the predicate-centric version, but the improvements were not significant on the EECB dataset. Running the same diagnostic as the predicate-centric feature reveals similar support: in 57.1% of the cases where an argument was aligned, at least one predicate it partook in was aligned too, compared to 7.6% of cases for non-aligned arguments. Both the predicate- and argument-centric improve similarly across both predicates and arguments on EECB.

While each of the joint factors all improve over the baselines on RF, the full model with all the joint factors does not perform as well as with some factors excluded. Specifically, the fertility model performs the best. We attribute this small gap to lack of training data (RF only contains 64 training document pairs in our experiments), as this is not a problem on the larger EECB dataset.

Additionally, the joint models seem to trade precision for recall on the RF dataset compared to the *Local* baseline. Note that both models are tuned to maximize F_1, so this tells you more about the shape of the ROC curve as opposed to either models' ability to achieve either high precision or recall. Since we don't see this behavior on the EECB corpus, it is more likely that this is a property of the data than the model.

6.5 Related Work

Bejan and Harabagiu (2010) studied unsupervised cross-document event coreference through the use of non-parametric Bayesian models. They include lexical features of event mentions as well as features which expose the headwords of arguments chosen by a semantic role labeler which are similar to our ϕ^{psa} factors. They found that lexical features were too fine grain which lead their model to over-split events. They also comment that a large part of their errors are caused by a lack of part-whole coreference labels such as "Israeli forces" are a part of "Israel". In the terminology used here, these inferences are needed to avoid spurious penalties incurred by the ϕ^{psa} factors, e.g. in inferring z^p_{ij} for "Israeli forces [retreated]$_i$" and "Israel [retreated]$_j$", if these part-whole coreference decisions cannot be made ϕ^{psa} will penalize an otherwise easy linking decision.

Lee et al. (2012) considered a similar problem but sought to produce *clusters* of entities and events rather than an alignment between two documents with the goal of improving coreference resolution. They used features which consider previous event and entity coreference decisions to make future coreference decisions in a greedy manner. This differs from our model which is built on non-greedy joint inference, but much of the signal indicating when two mentions corefer or are aligned is similar.

The task of predicate argument linking was introduced by Roth and Frank (2012), who used a graph parameterized by a small number of semantic features to express similarities between predicates and used min-cuts to produce an alignment. Their model was updated in Roth (2013) to include SRL-based features which check the semantic similarity of arguments with a common role, discourse similarity features which measure how far through a document a predicate is mentioned, and a predicate type context similarity which looks at a window of neighboring predicates. These improvements beat our local method but perform worse than our global model, even when we don't reason about arguments.

In the context of in-document coreference resolution, Recasens et al. (2013) sought to overcome the problem of opaque mentions, nominal references to entities which require word knowledge to justify (such as "chipmaker" referring to AMD), by finding high-precision paraphrases of entities by pivoting off verbs mentioned in similar documents. We address the issue of opaque mentions not by building a paraphrase table, but by jointly reasoning about entities that participate in coreferent events (c.f. §6.4.2); the approaches are complementary.

6.6 Conclusion

In this chapter we study methods for predicate argument alignment. We describe two new linking models, one of which is based on a rich set of features based on semantic resources, and another augmented with a joint quadratic inference model which enforces a set of coherence conditions for a predicate argument alignment.

Predicate argument linking is useful for report linking by enabling users to do a variety of tasks which require being able to tell when two mentions are referring to the same entity or event. This includes the ability to find new mentions of a claim make in a report, finding documents which contain descriptions of events in a report, or finding sentences which describe things which are already covered in a report so that a user can skip over redundant text.

Predicate argument linking is appropriate when a user would like to link a report against a small set of documents. This small set could be generated by the related entity mention finding work described in Chapter 3. The predicates and arguments which are the input to a linker can be inferred using the frame and role identification methods in Chapter 5.

Chapter 7

Entity Summarization

7.1 Introduction

The work described so far in this thesis is towards the goal of building a graph (links are edges) of entities and situations related to the contents of a report. This graph in part serves as a representation for machine understanding, but it is also intended to be a tool for helping users explore large text collections quickly and efficiently. Towards this second goal, we have assumed up to this point that users are happy to explore the link structure in the same way a machine does: with cues from features on nodes and edges and with pointers to mentions from which those features are computed. This may not be ideal for some users, who will not understand how such a report linking system works, or what the features mean. It may be preferable to have a more human-friendly representation for nodes in our knowledge graph, and creating these representations is the subject of this chapter.

Entity summarization is the task of producing short textual summaries describing information about a given entity. For report linking, entity summarization offers a human-friendly way to view entity nodes in a knowledge graph which supports exploration.

What is the relationship between entity summarization and other forms of text summarization like multi-document summarization? Entity summarization can be thought of as a faceted version of multi-document summarization where the facets are entities. Multi-document summarization is well suited for the case where the source documents talk about a relatively narrow topic for which a few tens or hundreds of words might suffice for covering the majority of users' information needs. This setup works well when there is a mechanism for linking a user's information needs to a small collection of documents covering this topic, most often realized through information retrieval techniques based on an informed query. Entity summarization is meant to aids in non-query-driven exploration of non-trivial sized text collections, often when a query is difficult to formulate given a large or under-specified information need. In these cases a few hundred word summary is not likely to cover the information need, so exploration is preferable. By organizing the summary into facets (entities), exploration can be efficient (skipping over irrelevant facets) and complete (once a summary has some structure, it does not get unwieldy when its length exceeds a few hundred words).

Topic models a commonly suggested method for exploring large text corpora, but they offer little in terms of structure which can be used to intelligently explore text.[1] Knowledge base population techniques offer a lot of structure and can cover information distributed over a large text collection, but require a user to be familiar with its relational schema and

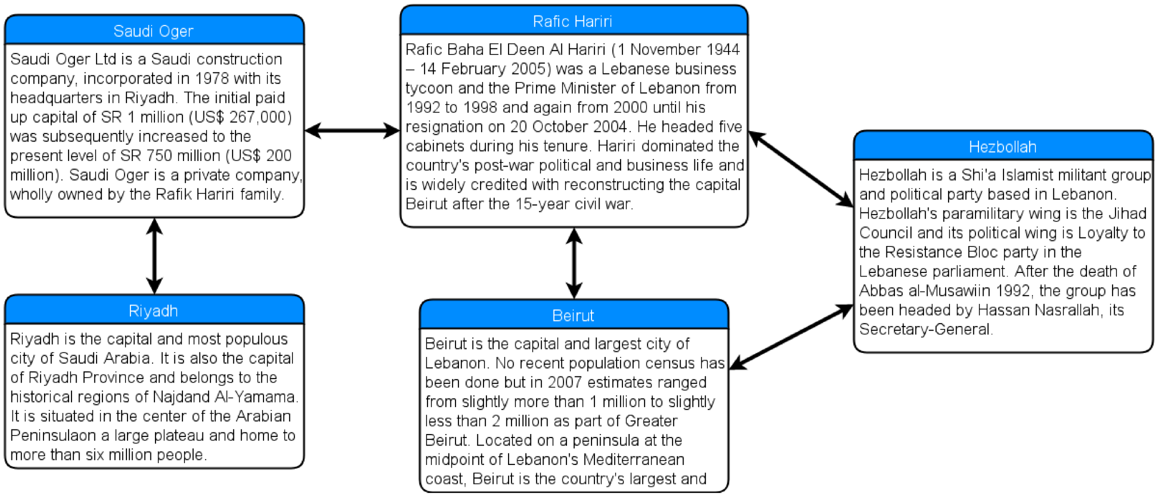
[1] By "intelligently explore" we mean in a manner which takes less time than brute force, as in reading everything. Intelligent exploration can be done when a user knows what they want and can use the information structure provided in order to find it faster than through enumeration.

pass along mistakes made during information extraction to unsuspecting users. Entity summaries offer significant structure to aid exploration and present information as the author of the source material wrote it, mitigating issues around noisy extraction.

To create entity summaries, we require a model of what information is worth describing about entities. This is another area where entity summarization can improve upon multi-document summarization. Using an entity as the subject of a summary may lend itself to more plausible models of information utility than can be used in the general case, unstructured heterogeneous collections of text. This helps us better address the question of *what information to include in a summary*, beyond the question of organizing information (by entity) discussed earlier.

Previous work in text summarization has mostly modeled information at the lexical level. We propose two new units of information about entities: the presence of related entities and of facts about an entity. In the former case we find that existing lexical methods are highly correlated with the related entities model, but are subject to ambiguity issues which entity disambiguation methods explicitly address. In the latter, we find that modeling facts about an entity produces better summaries than lexical models of information content.

In order to create fact-filled summaries, we use Wikipedia infoboxes as signal for what facts are most useful to users at a glance as well as for training high precision extractors. We incorporate the distant supervision methods described in Chapter 4 in order to train high-precision fact extractors which can be used as input for our entity summarization. We evaluate the generated summaries using human judgments and find that they are significantly more informative than previous work.



Saudi Oger

Saudi Oger Ltd is a Saudi construction company, incorporated in 1978 with its headquarters in Riyadh. The initial paid up capital of SR 1 million (US$ 267,000) was subsequently increased to the present level of SR 750 million (US$ 200 million). Saudi Oger is a private company, wholly owned by the Rafik Hariri family.

Rafic Hariri

Rafic Baha El Deen Al Hariri (1 November 1944 – 14 February 2005) was a Lebanese business tycoon and the Prime Minister of Lebanon from 1992 to 1998 and again from 2000 until his resignation on 20 October 2004. He headed five cabinets during his tenure. Hariri dominated the country's post-war political and business life and is widely credited with reconstructing the capital Beirut after the 15-year civil war.

Hezbollah

Hezbollah is a Shi'a Islamist militant group and political party based in Lebanon. Hezbollah's paramilitary wing is the Jihad Council and its political wing is Loyalty to the Resistance Bloc party in the Lebanese parliament. After the death of Abbas al-Musawiin 1992, the group has been headed by Hassan Nasrallah, its Secretary-General.

Riyadh

Riyadh is the capital and most populous city of Saudi Arabia. It is also the capital of Riyadh Province and belongs to the historical regions of Najdand Al-Yamama. It is situated in the center of the Arabian Peninsulaon a large plateau and home to more than six million people.

Beirut

Beirut is the capital and largest city of Lebanon. No recent population census has been done but in 2007 estimates ranged from slightly more than 1 million to slightly less than 2 million as part of Greater Beirut. Located on a peninsula at the midpoint of Lebanon's Mediterranean coast, Beirut is the country's largest and

Figure 7.1: An example of entity summaries for a few related entities taken from Wikipedia.

The rest of this chapter is structured as follows. In §7.2 we discuss modeling frameworks used in previous work on extractive summarization and our generalizations for incorporating information extraction. In §7.3 we introduce models of information content and describe how sources like Wikipedia infoboxes can provide signal. In §7.4 we describe a regularization method for extractive summarization systems which must be robust against low-quality input text (e.g. web text) and in §7.5 we discuss linguistic sentence-level features for characterizing which sentences make for good summaries. In §7.6 we discuss our evaluation framework and results. We discuss related work in §7.7 and conclude in §7.8.

7.2 Extractive Summarization Model

We begin by presenting the extractive summarization model used to produce entity summaries. This extractive summarization model assumes a set of *source sentences*, which in the case of entity summarization will be sentences that mention the *query entity* being summarized. The summarization model will choose what subset of the source sentences should be included in the entity summary.

Our baseline method will be to treat each sentence as a document and use a state of the art multi-document extractive summarization system to populate the entity summary. We are focused on extractive summarization in this work to more simply investigate the effect of models of information content, and leave compressive and other summarization models to future work. We will discuss the model abstractly in terms of *concepts*, or pieces of information mentioned in the source sentences. Later in §7.3 we will describe *implementations* for these concepts.

The model described in Gillick and Favre (2009) provides an extensible framework for extractive summarization based on the appearance of concepts, each of which has a *utility* to the reader and a list of locations it was mentioned. Their model is declarative: it describes an objective to be maximized rather than an heuristic algorithm for finding good solutions such as (Carbonell and Goldstein, 1998). Modern black box solvers can perform inference quickly, making this model very scalable. The model finds a summary of maximal utility which respects a length constraint. It is an integer linear program (ILP) of the form:

$$\max_{s,c} \sum_i w_i c_i \tag{7.1}$$

$$s.t. \sum_j l_j s_j \leq L \tag{7.2}$$

$$s_j Occ_{ij} \leq c_i \tag{7.3}$$

$$\sum_j s_j Occ_{ij} \geq c_i \tag{7.4}$$

$$c_i \in \{0,1\} \; \forall i \tag{7.5}$$

$$s_j \in \{0,1\} \; \forall j \tag{7.6}$$

where:

- w_i is a weight, or utility, for including the i^{th} concept in the summary
- c_i is a binary variable indicating whether the i^{th} concept is included in the summary
- Occ_{ij} is a binary value indicating that sentence j contains concept i
- s_j is a binary variable indicating that the summary contains sentence j

Later in §7.3 we will discuss the identification of concepts through relation extraction, which is a noisy prediction task. This entails uncertainty over the Occ_{ij} values, which is a problem for the model above where these values were observed (Gillick and Favre, 2009). In that work, concepts were implemented as word bigrams, which can be observed in the source sentence without any prediction.[2]

[2] This is true in English. In character-based languages such as Chinese extracting bigrams is a non-trivial segmentation problem.

To handle this uncertainty, our summarization model incorporates MAP inference over classifier predictions. This is done by introducing e_{ij} *variables* in place of Occ_{ij} *values*, and a cost p_{ij}. Our model only assumes p_{ij} is non-negative but for probabilistic classifiers it is natural to set it to $\log \frac{1}{p(Occ_{ij}=1)}$.

$$\max_{s,e,c} \sum_i w_i c_i - \sum_{ij} p_{ij} e_{ij} \tag{7.7}$$

$$s.t. \sum_j s_j l_j \leq L \tag{7.8}$$

$$\sum_j e_{ij} \geq c_i \;\forall i, j \tag{7.9}$$

$$e_{ij} \leq c_i \;\forall i, j \tag{7.10}$$

$$s_j \geq e_{ij} \;\forall i, j \tag{7.11}$$

$$\sum_j s_j e_{ij} \geq c_i \;\forall i \tag{7.12}$$

$$s_j \in \{0,1\}, c_i \in \{0,1\}, e_{ij} \in \{0,1\} \tag{7.13}$$

This model has many more variables than the model above (Gillick and Favre, 2009), but it is able to handle uncertainty about what concepts are described in text. Lexical models of information content have the advantage of lacking this uncertainty, but they cannot quantify information content in terms of facts which which are inferred through natural language understanding.

Pruning and Regularization ILPs for entity summarization can become fairly large[3] and may require pruning depending on the quality of the solver used and the time budget available. In development we tried a few methods of pruning and found that taking the top k sentences according to an upper bound on their utility (under no length penalty) consistently produced quickly-solved ILPs without sacrificing solution quality (according to the original problem). We keep a priority queue of sentences ordered by $U(j) = \sum_{i:p_{ij}<\tau} w_i$, an upper bound on the utility which could be extracted from a sentence. This method is sparse (many sentences can be eliminated outright) and requires time linear in the number of sentences, making it a scalable triage step. In practice we find that k does not need to grow very quickly in the summary length n, and set it to $k = 128 \log n$ and $\tau = 2$.

In §7.3 we will describe three concept definitions which may be used together. We found that that using more than one type of concept in the same model made it difficult to balance the influence of each concept type using linear coefficients per concept type. There is significant variation across queries of what the utility scale is for one type of concepts versus another. We found that we could make our model attendant to each type of concept for most queries by normalizing each concept to have unit weight. More precisely, if a concept type is a set of indices T, we ensure that $\sum_{i \in T} w_i = 1 \ \forall T$.

Additionally we found that in some cases the costs outweighed the utility of the concepts. In other cases this might be worth knowing (i.e. "we can't infer much about this entity given their mentions"), but in our case we would like the best summary possible. If our system produces an empty summary, we halve the p_{ij} values and re-run optimization.

[3] The size of the problem is linear in the number of source sentences, which can be very large if the entity being summarized is popular and a large corpus is used.

7.3 Concept Definitions

7.3.1 Lexical Baseline

Our baseline closely resembles the work of Gillick and Favre (2009) and uses word bigrams as concepts. The utility for a bigram w_i is computed in the same fashion as computing a tf-idf vector for bigrams. Gillick and Favre (2009) report that they didn't need to use idf weighting, but idf weighting helped in our experiments (c.f. §7.6). One reason could be that we are summarizing a great deal more material than in the TAC summarization task. There they summarized 10 articles, whereas we are summarizing all mentions of an entity in ClueWeb09, which can be over 1 million tokens for common entities. With so many choices, the ILP solver is good at finding silly sentences with abnormal amounts of frequent bigrams, to the exclusion of sentences with oft-repeated content words, which leads to bad summaries. When we do not use idf weighting, all of the top concepts are frequent bigrams like “of the” and “He said”, which have very little to no actual utility. When the set of source sentences is significantly smaller and comprised of high quality newswire text, the model has a harder time gaming the objective.

7.3.2 Wikipedia Infobox Distant Supervision

A natural way to model the informativeness of an entity summary is to count the number of facts contained in the summary. This intuition is even the impetus for some post-ROUGE based summarization evaluation metrics (Nenkova et al., 2007). Some facts matter more than others, e.g. where a person was born is more important than whether a university has an even or odd number of students. To determine which facts are informative we turn to Wikipedia *infoboxes*, which contain structured facts about an entity. These facts cover a wide range of types of entities and express things like which countries were affected by an earthquake, what company manufactures a drug, or where a person was born. Because they consume two finite resources, the time required by a person to write them down and the space at the top of a Wikipedia article, it is safe to conclude that these facts have utility.

The difficulty in using facts as concepts in our summarization model (§7.2) is that our systems can't observe facts directly, they must be extracted from text, requiring relation extractors for the relations in infobox facts. Therefore, we perform textual relation extraction to discover concepts that are worth including in our summaries. We build relation extractors using the distant supervision methods described in §4.3. As relation extraction relates to the utility and costs described in §7.2, we assume each mention of a relation (infobox facts) has unit utility[4] and the extraction cost is the extractor's confidence (MAP estimate of precision).

Using factual rather than lexical concepts explicitly addresses the sparsity and ambiguity problems with words. For example we could say "Lincoln was born in Hodgenville", "Lincoln is from Hodgenville", or "Lincoln hails from Hodgenville". Bigrams like "born in" and "is from" will correlate with mentions of Lincoln's birth place, but their utility will be computed independently, and thus underestimated. Lexical concepts are also subject to word sense errors, like "Palantir is from Tolkien's legendarium" vs "Lincoln is from Kentucky".

[4] Facts which have a relation type taken from a infobox has unit utility, and all other relations receive no utility. This method is coarse, but still injects a lot of signal into what should and shouldn't go into a summary.

7.3.3 Related Entities

Finally, we have our Goldilocks implementation: *related entities* as concepts. We started with lexical concepts, which are trivial to extract and have many ambiguity and sparsity issues, and then moved to factual concepts, which are difficult to extract but closely track information content. The purpose of related entities is to be a compromise. Entity disambiguation methods are more reliable than relation extraction methods and offer some of the information that factual concepts capture.

Take the example of summarizing a company. Their most related entities will likely be companies they do business with, high-level employees, products they make, people who review their products, and relevant regulatory bodies. Even if the summarization model does not know what the relation between the query and these related entities is, it can still assign them high utility on account of their association. At least in some circumstances we would expect that a sentence containing both the entity being summarized and one of its related entities: that sentence may sufficiently reflect the underlying relationship between those two entities, when presented to a human reader.

Related entities are also useful for disambiguation. For example, if there were two "Michael Jordan" entities, a very quick way to tell them apart is to produce two summaries; one with "Michael Jordan" and "Scottie Pippen" mentioned together in a sentence and another with "Michael Jordan" and "David Blei".

In general, there are many functions from an entity's associates to properties of entities which are helpful during exploration. For example you can predict someone's nationality fairly well if you know the nationality of most of their associates. The same goes for the language someone speaks, the company they work at, and sometimes more subtle things like their political affiliations (Volkova et al., 2014) or whether someone smokes (Ennett and Bauman, 1993).

This leads to the question of how to infer what entities are related to a query from text. In cases where the text is a part of communications like emails which have a senders and receivers, a graph of associated entities is fairly easy to create. We are primarily interested in sources which do not have senders and receivers, but we have found that the presence of two entities in the same sentences is a good source of evidence of association. Frequency alone can be misleading though. For example news agencies often report on many entities, and are thus mentioned frequently but are not essentially associated with the entities they report about. We found that the same $idf(i)$ correction applied to lexical concepts works well for related entity concepts too.

An additional factor in computing the utility of a related entity is whether the entity is known to the user who issued the query. Exploration may be driven by the need to connect an unknown entity to what a knowledge worker is already aware of, and displaying a related entity which is unknown to the user may not make for an effective summary. We do not model this factor in this work due to the difficulty of eliciting this sort of user knowledge and of evaluation when utility is conditional on what the user knows.

Given the utility of related entity concepts, we set the extraction cost for each mention of a related entity equal to the number of words between the query and related entity mentions divided by 10. We are not concerned with direct supervision, but future work might discriminatively model the extraction cost based on features of the co-occurrence

Cost	Sentence
32.31	Bundoran : Kevin McManus , Barry McGowan , Shane O'Donnell , Peter McGonigle -LRB- 0-1 -RRB- , Diarmuid McCaughey , James Keaney -LRB- 0-1 ... *and 53 more words*
29.42	Red House Painters -LRB- 1 -RRB- ; Red My Lips -LRB- 11 -RRB- ; Regina Spektor -LRB- 1 -RRB- ; Sam Sparro -LRB- 2 -RRB- ; Santogold -LRB- 1 ... *and 35 more words*
29.16	Bundoran : Kevin McManus , Ryan Walsh , Niall Gillespie , Gavin Croghan , Adam Dunmore , Rossa McKiernan , Ciaran Gunne , Shane McGowan ... *and 44 more words*
28.51	Bundoran : Johnny Keenan , Barry McGowan , Shane O'Donnell , Diarmuid McCaughey , Rossa McKiernan -LRB- 0-1 -RRB- , James Keaney , Peter ... *and 45 more words*
27.07	-RRB- Ambiance verzekerd , als ze materiaal van ondermeer Joe Jackson , Tammy Wynette , Van Morrison -LRB- het tot aegejasomgedoopte eal ... *and 54 more words*
26.17	Shane MacGowan guests with Sharon Shannon in Castlebar , Donegal , Cork , Galway , Killarney , Wexford , Ennis and Kilkenny .
25.36	Leader Dan McGee is the new great American songwriter-on-the-dole , whose multi-purposed voice sings of two things : girls and women , and ... *and 58 more words*
25.33	John Lydon , The Beatles , Morrissey , Billy Connelly , Tom Waits , Dropkick Murphys , Shane MacGowan , Robert DeNiro , Woody Guthrie , ... *and 21 more words*
25.12	This concert film finds Shane MacGowan cnn grace headline nancy news video and The Popes performing live at the Montreux Jazz Festival that ... *and 56 more words*
25.11	Merle Haggard in Nashville , TN ; Steve Earle in Chicago , IL ; Johnny Cash / Mark Lanegan in Portland , OR ; Shane MacGowan in Minneapolis ... *and 4 more words*
2.19	And speaking of the Pogues , this interview with Shane MacGowan has an interesting discussion of Lapsed Catholicism .
2.76	The legendary SHANE MacGOWAN 's genius is evident from his many hits with The Pogues .
2.84	The Pogues have always lived this contradiction : on one hand playing with amazing speed and precision , yet at dead-center is MacGowan 's ... *and 16 more words*
2.09	Wee Willie Harris was not manufactured because you could n't invent him and Shane McGowan was n't because , frankly my dear , you would n't ... *and 4 more words*
3.32	A good article on Shane MacGowan of Pogues fame in Fluctuat.net .
2.81	' And suddenly the spirits of Ewan McColl and Shane MacGowan are with us .
2.48	Shane McGowan was born on Christmas Day in 1957 when his parents were visiting relatives in Kent , and was brought up in a farmhouse in ... *and 4 more words*
2.68	Of course , if Strummer – who had guested alongside MacGowan on a 1987 Pogues tour – had asserted himself more , he might have risked ... *and 9 more words*
2.77	Shane played bass , Jem guitar and Ollie Watts from The Millwall Chainsaws was the drummer .
2.98	Touring has it 's pressures but Shane finds it a welcome relief from living in London all the time .

Figure 7.2: Examples of common extraction costs (§7.4) for input sentences for Shane McGowan. Above: most costly and irregular sentences, which are often lists, titles, or ungrammatical utterances which do not make for a fluent summary. Below: random sample of sentences with below median cost which our model can select sentences from.

like the words between the two related entities. Intervening phrases like "according to" are a good way to handle the problem of news agencies and may also provide a more general method of eliciting evidence of interesting entity relationships.

7.4 Common Extraction Costs

In the last section we introduced concept definitions for identifying information content but another important and orthogonal issue for our extractive model is grammaticality of the source sentences (and thus the summary in the case of our extractive model). Web text is full of "sentences" which are not like prototypical high quality newswire sentences and make poor additions to a summary. Some "sentences" are titles (can be noun phrases, ungrammatical, or less often, proper sentences), some are lists of entities, and some contain sentence-splitting mistakes. See the top half of Figure 7.2 for some examples of undesirable sentences.

One particularly pernicious example are over-split sentences: as long as the concept mentions (e.g. named entities) are not affected, an extractive summarization system will prefer a sentence with a few words lopped off (ungrammatical) to a full sentence (grammatical) due to the length budget.

However most of the sentences are fairly high quality (in grammatically and information content). We leverage this fact to come up with an outlier score, which is used as a cost for using a sentence in a summary. We define a few coarse features reflecting linguistic structure, and use the rank order of a given sentence to quantify whether it is an outlier. For example, if a given sentence is ranked 3 out of 1M sentences for the number of commas, chances are it is an abnormal and ungrammatical sentence.

This rank transformation gives an intuitive scale and obviates the need for parametric assumptions about the outlier features. Given a sentence j and features $f \in \mathcal{F}$, we rank its "outlier-ness" according to:

$$cost(j) = \sum_{f \in \mathcal{F}} A(f)B(j,f) \tag{7.14}$$

$$A(f) = \frac{N_f}{N_f + \alpha} \tag{7.15}$$

$$B(j,f) = \frac{N_f}{\min(L(f(j),f), G(f(j),f)) + \beta N_f} \tag{7.16}$$

$$L(x,f) = \sum_i I(f(x_i) < x) \tag{7.17}$$

$$G(x,f) = \sum_i I(f(x_i) > x) \tag{7.18}$$

Where N_f is the number of times feature f was observed.

A feature f which is near the max or min (having $L(x,f)$ or $G(x,f)$ near 0) may indicate a sentence is an outlier, as reflected in the denominator of Equation 7.16. The

$A(f)$ term is used to quantify how significant an extreme value is. We sum this cost across features, so sentence which are outliers with respect to many features are more costly to use in a summary. The features are counts of POS tags, the number of "nuisance" tokens,[5] the number of content words, the number of named entities, and the length of the sentence. All features except the last are conjoined with the sentence length to ensure that extracted values are comparable (e.g. a sentence with 4 named entities is a common for a sentence with 35 words, but rare for one with 6 words). We present some examples of good and bad sentences in Figure 7.2.

We add a $-\sum_j cost(j)s_j$ term to our objective in Equation 7.7 to penalize the inclusion of outlier sentences in our summaries. We found that leaving this penalty off lead to a huge detriment in all models. The difference was so obvious that we decided not to pay to annotate examples without this cost.

7.5 Alternative Sentence Costs

7.5.1 Topicality

We hypothesize that sentences where the query entity is in subject or topic position (Prince, 1998) make for summaries which are easier to read, more directly pertain to the query, and are judged as more informative. The topic of a sentence is a constituent which has been moved to the front for emphasis. We model this by computing a de-topicalization cost for each source sentence equal to the number of words preceding the first mention of the query plus the depth of the entity in a dependency tree. For example, if the query

[5] Punctuation, symbols, list items, foreign words, interjections, and parentheticals

was George Bush, this cost would prefer the sentence “George Bush was president and commander in chief at the time of 9/11.” over “During 9/11, the president and commander in chief was George Bush.” Or if the query was ACME, the cost would prefer “ACME is planning to acquire SoftCorp in 2017” over “Jane Smith, the CTO of ACME, lives in Washington DC.” This score will serve as a tie-breaker in cases where the model can choose between two differently phrased but comparably informative sentences.

7.5.2 1^{st} and 2^{nd} Person Pronouns

We hypothesize that first and second person pronouns make for bad summaries. Third person pronouns can also be problematic for extractive summarization systems because their referents may or may not be contained within the sentence or summary, leading to disfluencies (Durrett, 2016). Leaving aside cases which involve narration and quotation (which are not germane to fact-based summarization), first person pronouns are effectively guaranteed to have missing referents because we do not preserve the author of the sentences we include in the summary. Second person pronouns presumably refer to the reader, but it is not clear whether the same author-reader relationship holds now that the reader is the person viewing the summary rather than the reader of the source text.

Additionally first and second person pronouns are common in web text and exclusively used for interpersonal and subjective language, we believe is not informative in the way that our concepts measure. For example, sentimental: “The Kinks are great, but I don’t like Village Green Preservation Society” vs objective: “Village Green Preservation Society is the sixth studio album by the Kinks”. Other work on summarization has focused on sentiment for summarization, but it is incongruous with our definitions of concepts and utility.

7.6 Experiments

To evaluate our entity summarization methods, we construct a dataset of entities taken from the ClueWeb09 dataset (Callan et al., 2009) with the entity linking annotations provided by FACC1 (Gabrilovich et al., 2013). This data has over 340M documents and 5.1B entity mentions which are resolved to Freebase MIDs (which can be mapped to DBpedia entities and aligned with infobox facts). There are 18M infobox facts which have an entity as the subject and object that we considered in this work. This covers 4165 relation types which occur at least 30 times and 2580 which occur at least 100 (see Figure 4.2 for more details).

We select train, dev, and test sets of entities for our summaries. The set of all entities in FACC1 is partitioned into buckets by log frequency. We opt for a stratified test set so that we can measure the effectiveness of summarization methods at various levels of entity "popularity", see Table 7.1. For each bucket we selected 3800 train, 100 dev, and 100 test entities. The three most popular buckets (rare0, rare1, and rare2 in Table 7.1) did not provide enough entities to constitute full sized train and test set, and we exclude these from our experiments.

Depending on the use case and the amount of information available about an entity, one may want to produce summaries of various lengths. We generate summaries of length $L \in \{20, 40, 80, 160\}$ words for each entity and average our results over all performance at every length.

Bucket	$\min f(e)$	$\max f(e)$	# Entities
rare0	27,938,100	∞	1
rare1	2,940,784	27,938,099	39
rare2	309,549	2,940,783	614
rare3	32,583	309,548	6,533
rare4	3,430	32,582	49,792
rare5	361	3,429	241,994
rare6	39	360	777,152
rare7	4	38	1,465,706

Table 7.1: Frequency of entities mentioned in FACC1.

7.6.1 Metrics

One category of summarization evaluation methods is based on expert annotation, either through the production of reference summaries or through annotation of a pool of summaries identifying their information content. The canonical example of the former is ROUGE (Lin, 2004) which is widely used, and examples of the latter include the pyramid method (Nenkova et al., 2007), and Basic Elements (Tratz and Hovy, 2008). These methods are problematic for this work because we are not aware of any existing references for entity-centric summaries and because creating them, or annotating a pool of summaries, is costly and requires expert annotators.

Another line of evaluation coming from information retrieval (Blanco and Zaragoza, 2010; Chhabra, 2014) grades summaries by giving them an ordinal quality label (e.g. a 4 level scheme like non-relevant, fairly relevant, relevant, and very relevant). While this gets around the need for costly expert annotations, it has some drawbacks. First, the number of categories, and thus the precision by which annotators can describe summary quality, must be kept small for practical reasons (ease of task and consistency of labels). Second, in order to get a single score to compare systems, this requires converting ordinal to scalar values, which is not optimal.

We adopt methods from machine translation evaluation (Sakaguchi et al., 2014) which are based on pairwise comparisons of summary quality. For each entity in our evaluation set, each summary length, and each pair of systems which produce a summary, we present Amazon Mechanical Turk annotators with a choice of which summary is more informative. We do not present examples where differing systems produced the same summary. In cases where the summaries contain a common sentence, we put these sentences elsewhere on the page so that annotators can see the entirety of both summaries, but focus primarily on what differs between them.

We force annotators to choose a summary even if they believe they are equally informative. If the summaries are actually equally informative then the annotators' choices will be random with respect to the system being evaluated; which corresponds to the null hypothesis in our statistical tests.

There are two types of evaluation we consider. In the first we have a group of system types and we would like to know the relative ranking of each of them. For this we use the TrueSkill algorithm (Herbrich et al., 2006) to infer the skill, a parameter which determines the likelihood that any one system will produce a more informative summary than another system, which we use to rank systems. In the second case, we only need to know if a baseline system or an alternative system produces better summaries. TrueSkill could infer this, but it is more statistically efficient to perform an exact binomial test when n-way rankings are not needed. In measuring statistical significance, in the first case we report the standard error inferred by TrueSkill and in the second case we report p-values determined by a two-sided test on the null hypothesis that both systems have an equal chance of being labeled more informative.

One final advantage of binary comparison-based evaluation is that it has lower variance than methods comparing means of un-paired examples. This is accomplished by removing the variance associated with the average informativeness of a system over a set of queries – pairwise tests cancel out this variance by using the same sets of queries. This makes for a more powerful test which requires less annotation, and is therefore cheaper.

7.6.2 Systems

We define a system as some combination of the concept definitions. We denote the baseline method of word bigrams as W, the infobox relation method as S (§7.3.2), and the related entities method as E (§7.3.3). Systems with two types of concepts are denoted with concatenation, e.g. S + E = SE. In total there are 7 systems: [E, S, W, ES, SW, EW, ESW]

For the alternative sentence costs defined in §7.5, we sample pairs of summaries with and without the additional cost and elicit a pairwise preference. We denote topicality costs with +T and first and second person pronoun costs with +P.

7.6.3 Results

In Figure 7.3 we plotted the skill for every system. Defining concepts as infobox facts produces the most informative summaries. This is the only statistically significant result using W as the baseline.

Using related entities (E) produces summaries with comparable quality to word ngrams (W). This is consistent with properties we saw during development: when using

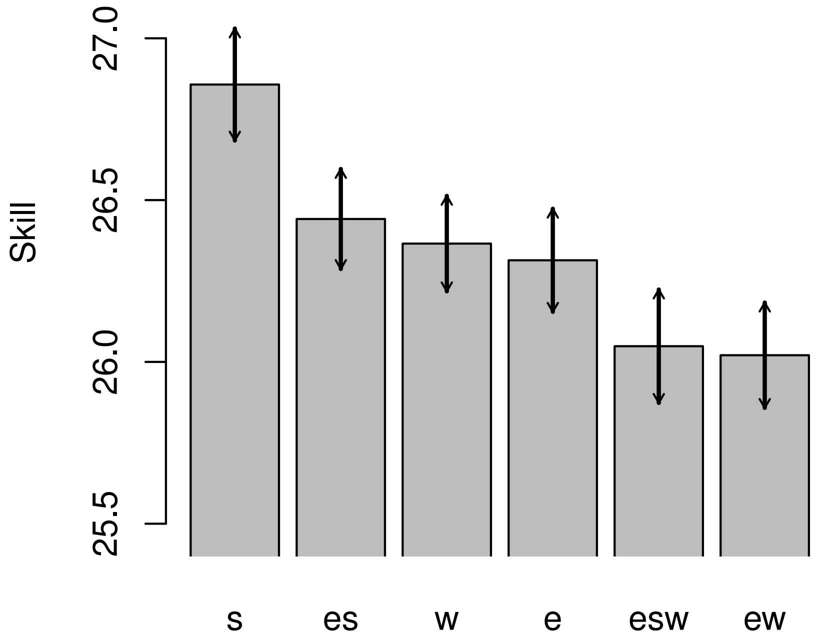


Figure 7.3: System (concept definition) skill (summary quality). Means and standard errors are inferred by TrueSkill.

W concepts, the highest utility concepts, i.e. ngrams with high tf-idf weight, tend to be named entities. This indicates that most of the information that the E and W models are trying to preserve is the same. This may explain the fact that models which have both E and W components perform the worst: in these cases the model is redundantly rewarded for including the same information, thus not spending its word budget in the most informative way.

We can decompose the set of concepts which W is trying to preserve into entity concepts and non-entity concepts. While E is a fairly natural replacement for the entity concepts in W (with associated benefits described in §7.3.3), its unclear how useful the non-entity concepts are. To quantify this, we defined a new set of concepts V which are the ngram concepts which do not overlap with any entity mention. We treat them the same as W concepts and weight them with tf-idf weighting. In principle these concepts could pick up on the information which S discovers by highly weighting lexical predicates and open information extraction patterns like *"born in", "founded a", "headquartered in", "is a"* which do not overlap with entities. These concepts can also refer to noun phrase concepts like *"gay rights", "hazardous waste",* and *"pituitary gland"* which are not entities and thus not linked to the knowledge base in these experiments. Further, this information should be orthogonal to entities, indicating that perhaps EV might have fewer duplication problems compared to EW.

In Table 7.2 we performed a head-to-head evaluation between systems which use W concepts and the corresponding system with V instead. We produced a summary for both concept definitions and asked annotators which they preferred when the summaries

h_0	h_1	$c(h_0\text{wins})$	$c(h_1\text{wins})$	$p(x \mid \mu_0 = \mu_1)$
W	V	135	116	0.256
EW	EV	112	83	0.00780
ESW	ESV	134	93	0.00779
*W	*V	381	292	0.000682
E	E+T	70	67	0.864
S	S+T	51	61	0.395
W	W+T	92	65	0.0376
ES	ES+T	73	65	0.551
EW	EW+T	78	68	0.456
ESW	ESW+T	87	86	1.00
*	*+T	451	412	0.196
E	E+P	28	24	0.678
S	S+P	26	25	1.00
W	W+P	35	29	0.532
ES	ES+P	33	35	0.904
EW	EW+P	41	42	1.00
ESW	ESW+P	50	30	0.0330
*	*+P	213	185	0.176

Table 7.2: Head-to-head matchups for differing summaries. The third and fourth columns are counts of the number of summary pairs where either the baseline system (h_0, first column) or the alternative (h_1, second column) is judged as a better summary. The final column is a significance test, likelihood of observing these counts if the two systems had the same quality. The first block addresses the effect of removing entity names from ngram concepts. The second block addresses whether topicality costs (+T) improve summary quality. The third block addresses whether penalizing 1st and 2nd person pronouns (+P) improves summary quality.

were different. We see that in every case the V model performs worse, most frequently by statistically significant amount. This is further evidence that ngrams are not a sufficient method for modeling information beyond what named entities already capture. This highlights the effectiveness of the S concept definitions: S uses only non-entity information and performs significantly better than the strong entity baseline.

Topicality Costs In §7.5 we hypothesized that topicality affected the quality of a summary by rewarding the model for choosing sentences where the entity being summarized is

Pronoun	$c(h_0 \text{ wins})$	$c(h_1 \text{ wins})$	$p(\mu_a = \mu_b)$
i	66	55	0.363
we	45	50	0.682
you	54	37	0.093
us	29	19	0.193
our	25	16	0.211
your	21	18	0.749
my	11	10	1.000
me	3	10	0.092
u	7	0	0.016
mine	2	3	1.000
myself	1	1	1.000
yourself	1	0	1.000
yours	0	1	1.000

Table 7.3: Baseline vs +P annotations for various penalized pronouns.

the subject or topic of the sentence. We tested this in the same fashion as W vs V and the results are listed in the second block of Table 7.2. Overall adding the topicality cost did not produce statistically significantly higher or lower quality results (two-sided test),[6] but in our sample annotators more often preferred the summary with no topicality cost added. We are not aware of other work which addresses the role of topicality at the level of syntax in summary quality. Our work indicates that annotators do not care about topicality of the summary subject as we have defined it in §7.5.

Pronoun Costs Similarly in §7.5 we described a cost which was a softened version of eliminating first and second person pronouns. We perform the same type of head-to-head evaluation and give the results in the third section of Table 7.2. The results were not statistically significant, but it seemed as if annotators might have preferred the summaries

[6] Note, this is not a statement about how similar the summaries themselves are, which is a function of hyperparameters concerning how heavily topicality should be rewarded, but rather the assessed quality of the summaries when they differed.

without the cost on pronouns. To investigate further we looked at each matchup where the baseline model produced a summary which contained a first or second person pronoun. In Table 7.3 we pool the annotations across systems, showing weak evidence that annotators prefer the baseline system.

Example Infobox Fact Extractions To offer some qualitative explanation for how the infobox method works, we have selected some sentences which were included in a summary because they were predicted as including an infobox fact for the subject of the summary, listed in Table 7.4. There are a wide range of infobox relations which show up in summaries.

7.7 Related Work

There is a related line of work on entity summarization where the output is a set of facts or triples in a knowledge base (Cheng et al., 2011; Gunaratna et al., 2015; Thalhammer et al., 2012). These methods compute the importance of facts connected to a knowledge base entity using graph-based methods. This work focuses on producing textual summaries rather than triples and uses repetition in text or presence in an infobox as an orthogonal signal for a concept or fact's utility. Additionally this work is concerned with extracting facts rather than ranking them.

The TREC Entity track (2009-2011) (Balog et al., 2010b) addressed the task of finding related entities to a query entity, which was treated as its own information retrieval task. Our work shows that related entities are an effective means of producing summaries.

Gong et al. (2010) performed TAC topic-based summarization (not entity-based) using links into Wikipedia to determine how popular or salient an entity is, and thus how

Relation	**Summary Sentence**
japanActor	(Hugo Weaving) will voice [Megatron] , leader of the Decepticons , in the live-action Transformers movie .
yeshiva livingPlace	[Shlomo Amar] , (Israel) 's chief rabbi , has welcomed the Falash Mura back to the fold .
associatedActs	(Elvis) ' original drummer , [DJ Fontana] , is also scheduled to appear .
nflDraftedTeam nhlTeams	[Pablo Sandoval] had two of the (Giants) four hits .
birthPlace	[Paul Morrison] was born in (Liverpool) in he studied at Hugh Baird College Liverpool .
licensingAuthority	[ChexSystems] is governed by the (Fair Credit Reporting Act) -LRB- FCRA -RRB- and other laws .
watercourse landmark	The (Igua Falls) are waterfalls located on the border of Brazilian state of Parana -LRB- in the Southern Region -RRB- and the Argentinian Province of [Misiones] .
brightestStarName nearestStarName	(Alphard) is the brightest star in the constellation [Hydra] .
nearestStarName	(NGC 3314) is about 140 million light-years away in the southern constellation of [Hydra] .
subtribus	(Amaryllis) -LRB- amaryllis -RRB- still belongs to the Amaryllidaceae , but also includes many genera that were once included in the [Liliaceae] .

Table 7.4: Sentences which were included in a summary produced by the infobox relation method. The [subject] of the summary is shown in square brackets and the (object) of the fact which justified this sentence's inclusion in the summary are rendered in parens. The fact's relation is listed on the left. In some cases the model predicts that multiple relations hold. Not all of the predicted facts are correct, but the sentences tend to be informative nonetheless. Since source material is drawn from the web, it can contain typos and ungrammaticalities in some cases.

much utility should be assigned to including it in a summary. This method of estimating utility is directly applicable to our summarization model and is similar our definition of related entities, but we do not assume that one has access to Wikipedia logs to determine popularity.

Meng et al. (2012) created entity-centric summaries for companies on Twitter, where the goal was highlight positive and negative sentiment. They used hashtags as concepts. Similarly, Mason et al. (2016) jointly performs sentiment analysis and summarization jointly to create “micro reviews” on services like Yelp.

Liu et al. (2015) perform abstractive summarization using AMR as a semantic representation. Our work is not abstractive, but is perhaps a hybrid in that it uses extracted relations as concepts, which are a very weak meaning representation. They transform the text summarization problem into one of graph summarization and lift the textual training data to graph training data using an AMR parser. Our summarization model, being related to Gillick and Favre (2009), maintains the textual provenance of concepts which is used to produce a summary. They do not address the text generation from AMR problem and instead produce unigram summaries (which can be evaluated with ROUGE-1).

7.8 Conclusion

In this work we investigated the task of entity summarization. We described how existing extractive summarization methods can be adapted to produce entity summaries and compared the effectiveness of these methods to novel entity-centric concept definitions. To enable summarization in the presence of noisy extractors, we describe a new extractive summarization ILP model which jointly performs MAP inference over extractions and summarization choices. Our new model and concept definitions significantly outperform existing summarization methods. We find that using distant supervision of infobox facts from Wikipedia lets us train extractors which find useful information to incorporate in our summaries, outperforming all other methods. Our experiments indicate that entity co-occurrences explain most of the signal that previous methods had exploited. By modeling entities explicitly through linking, we can better weigh the importance of entities and avoid concept duplication issues introduced by only using words. For non-entity signal, we show that our distant supervision model greatly outperforms previous methods. This work also includes new applications of evaluation techniques from machine translation which make evaluating summarization techniques cheaper and more statistically powerful.

Chapter 8

Conclusion

8.1 Report Linking

First, we reflect on the goals of this thesis and how we went about accomplishing them. Report linking is a broad task of building semantic link structures atop reports used by knowledge workers to communicate information about a subject. The goal of report linking is to make the organization and exploration of information in large volumes of reports easier and more efficient for knowledge workers. This is clearly a broad topic to address in a single work, and we do not claim to have uttered the last word on methods for accomplishing the goals of report linking. We have however clearly defined a framework and related tools which deliver on the goals of report linking. A guiding principle in our definition of report linking is the need to ground out our analysis in the reports themselves which knowledge workers of today use. This is important because research on report linking can only be improved through empirical verification of the utility of the tools it provides. The more abstractions that report linking tools make which knowledge workers do not, the more difficult it is to proceed with empirical research which must at some point consume the time and attention its subjects: knowledge workers. The contribution of this thesis, in total, are a set of working tools which accomplish the goals of report linking and can help knowledge workers today.

The contribution of this thesis in composite however is advances in many natural language processing techniques which make report linking possible. This thesis is organized around the contributions made to these sub-goals and we conclude by summarizing them and their relationship to report linking here.

The first step in our conception of report linking is the construction of the topic knowledge base (TKB) from a set of reports. In this work we assumed that these reports were given and that there were between tens and tens of millions of them. In Chapter 3 we begin by discussing how to construct entity nodes in the TKB from entity mentions found in reports. We propose a simple unsupervised method for enumerating recognizing distinct entities and enumerating all the mentions within a specified entity. Our experiments show that our method can recognize entity mentions with precision above 90%, as good or better than comparable previous work without requiring training data. In that chapter we also address the question of determining whether two entities are related or not, and whether there should be an edge connecting them in the TKB. Our experiments showed that simple methods which only require co-occurrence statistics like pointwise mutual information are capable of finding related entities with greater than 80% accuracy. This chapter lays out how one can construct an unlabeled TKB from a large collection of reports.

Next in Chapter 4 we turn to the problem of labeling the edges connecting two entities in a TKB. The goal of such a label is to explain to a knowledge worker the nature of the relationship between two related entities, using the text provided their reports. We demonstrate two approaches for choosing these labels, one which is high recall and unsupervised and the other is high precision and leverages supervision from knowledge resources like Wikipedia to identify facts and relations which are informative. The first approach is a novel way of picking trigger words which explain entity co-occurrences. These trigger words are selected based on syntactic and information-theoretic criteria for what makes a good explanation. Our experiments showed that these trigger words were informative twice as often as those chosen by a syntactically informed baseline. The next approach uses relation extractors trained using distant supervision to search through reports to find common and informative types of facts like where a person was born and whether a person is an executive of a company. This method leverages the Wikipedia infoboxes to determine what facts are most interesting and to train models to recognize these facts. Our contributions in this area are a new syntactic approach for finding high precision extractors. We also define and validate a new objective which rewards type-based diversity which is better at separating extractors which correlate with the meaning of a relation with those which express the meaning of the relation. A final contribution in this chapter was the release of an annotated dataset for training relation extractors which is an order of magnitude larger than used in other work.

In the next two chapters we returned to the problem of building the TKB structure which connects reports to other reports. In chapters 3 and 4 we built up methods for linking reports to the entity nodes and for linking entity nodes to other entity nodes respectively.

To connect reports to other reports, we develop deeper methods of linking which go beyond lexical or entity-centric linking methods.

In Chapter 5 we develop the basic tools for identifying events and situations described in reports. We develop tools which use semantic frames as the abstraction over events in reports which we will later link. A frame-level analysis provides a disambiguated view of the events in a report and explain what roles the entities in the TKB play in these events. Towards this goal, we develop new models for frame semantic parsing which leverage global features and greedy inference. The contributions in this chapter are a careful analysis of how existing methods for training these types of models can go wrong as well as new methods for training pipeline-based models which use global features. The methods in this chapter outperform comparable models with no global features without the need for asymptotically more complex inference. This improves report linking by making more accurate frame analyses of reports in less time.

In Chapter 6 we build upon the work in the previous chapter and develop methods to create structured links between reports in the TKB called predicate argument alignment. Each link is an alignment between the entities and events in two reports. This type of link can be used to tell knowledge workers what is novel vs what is known if they have only read one report in the alignment. The contributions in this chapter are two models for predicate argument alignment, one of which is state of the art on two datasets. The first model is a classification-based model which uses a rich set of semantic features based on lexical resources. The second is a 2^{nd} order structured model which scores pairs of alignments which appear in predicate argument structures in each report. This model is much more expressive and can ensure that the predicted links are consistent with relations like temporal ordering of events described in both reports.

Having explained how to build a topic knowledge base and annotate the edges with entity relationships and predicate argument alignments, in Chapter 7 we add one more view to our TKBs, a textual summary of every entity in the graph, generated from the source reports. These summaries are intended to distill all the reporting on a given entity down to the most important facts. In addition to the entity relation explanations, these summaries offer an efficient means for a knowledge worker to bone up on the salient entities discussed in a set of reports. The primary contribution of this chapter is an extractive summarization model for generating entity summaries which models facts and other related entities as first class concepts. Compared to a strong lexical summarization baseline, the fact-based entity summarization model performs significantly better as judged by annotators. The summarization model presented in this chapter is also extensible, and can naturally incorporate other concepts which are important to be summarized, and even noisy predictors of these concepts.

8.2 Future Work

Evaluation in this thesis has been primarily on metrics which have to do with the quality or veracity of the predictions made. For example, we might predict that a mention refers to an entity, and this can be correct or incorrect, so we measure the accuracy. This was done as a necessary measure to ensure that the building blocks of a report linking system worked as intended, but it does not speak to the overall utility of such as system, except in limited cases such as our evaluation of the utility of entity summaries. Future work on human computer interaction, as it relates to report linking, might answer questions like which types of links are most useful to which users during which tasks.

Another related avenue of study in improving report linking is in incorporating human-in-the-loop methods for learning and tuning the models involved in report linking. The work we have done is based on either unsupervised methods (when possible) or supervised methods with labeling schemes which purport to be domain robust (independent is too strong). We have not tested the generalization of these models to new domains and it seems likely that some domains will force systematic errors in a full report linking system. Methods for collecting annotations and amending predictions by knowledge workers would make the tools more useful as well as better performing. The three areas which might benefit from online learning are entity, event, and relation detection. Methods which make it easy to elicit either examples, lexicons, or prototypes to aid in extraction of types which were not seen frequently during initial training could make for more precise and useful tools.

Work on event representations can also yield improvements in predicate argument alignment. The structured alignment model described in Chapter 6 does leverage the predicate argument structure to a degree, but does not provide a full account of the role assignments on either end of an event link. Partially this is done because the alignment models which we train and evaluate in this work need to work on a variety of different datasets, each of which have their own annotation scheme which makes assumptions which may go beyond or fall short of more detailed theories of event and role linking. But partially this limitation in model complexity is done because richer joint inference is both computationally costly and may lead to overfitting (the corpora available for training these models is relatively small). Future work on modeling event coreference and linking may be able to avoid some of the problems of variations in annotation schemata by proposing generative models which do not need to use idiosyncratic annotations. Future work on joint inference might benefit from exploring de-lexicalized, or embeddings-based models, which have less ability to overfit.

Bibliography

Omri Abend and Ari Rappoport. 2017. The state of the art in semantic representation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, pages 77–89. http://aclweb.org/anthology/P17-1008.

Karl Aberer, Philippe Cudré-Mauroux, and Manfred Hauswirth. 2003. The chatty web: Emergent semantics through gossiping. In *Proceedings of the Twelfth International World Wide Web Conference*. ACM Press, Budapest, Hungary, pages 197–206. https://doi.org/http://doi.acm.org/10.1145/775152.775180.

Jacqueline Aguilar, Charley Beller, Paul McNamee, Benjamin Van Durme, Stephanie Strassel, Zhiyi Song, and Joe Ellis. 2014. A comparison of the events and relations across ace, ere, tac-kbp, and framenet annotation standards. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*. Association for Computational Linguistics, Baltimore, Maryland, USA, pages 45–53. http://www.aclweb.org/anthology/W/W14/W14-2907.

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Josef Sivic, Ivan Laptev, and

Simon Lacoste-Julien. 2016. Unsupervised learning from narrated instruction videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. 1998. Topic detection and tracking pilot study: Final report. In *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*. Lansdowne, VA, USA, pages 194–218. 007.

Nicholas Andrews, Mark Dredze, Benjamin Van Durme, and Jason Eisner. 2017. Bayesian modeling of lexical resources for low-resource settings. In *Proceedings of the Association for Computational Linguistics*. Vancouver.

Nicholas Andrews, Jason Eisner, and Mark Dredze. 2012. Name phylogeny: A generative model of string variation. In *EMNLP-CoNLL*. ACL, pages 344–355.

Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Support vector machines for multiple-instance learning. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, MIT Press, pages 577–584. http://papers.nips.cc/paper/2232-support-vector-machines-for-multiple-instance-learning.pdf.

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, pages 344–354. http://www.aclweb.org/anthology/P15-1034.

BIBLIOGRAPHY

Austin Appleby. 2017. aappleby/smhasher. https://github.com/aappleby/smhasher/blob/master/src/Mur

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: a nucleus for a web of open data. *The semantic web* pages 722–735.

Amit Bagga and Breck Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '98, pages 79–85. https://doi.org/10.3115/980845.980859.

Amit Bagga and Breck Baldwin. 1999. Cross-document event coreference: Annotations, experiments, and observations. In *Proceedings of the Workshop on Coreference and Its Applications*. Association for Computational Linguistics, Stroudsburg, PA, USA, CorefApp '99, pages 1–8. http://dl.acm.org/citation.cfm?id=1608810.1608812.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*. ACL.

Niranjan Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. 2013. Generating coherent event schemas at scale. In *EMNLP*. ACL, pages 1721–1731. http://dblp.uni-trier.de/db/conf/emnlp/emnlp2013.html#BalasubramanianSME13.

BIBLIOGRAPHY

Breck Baldwin. 1997. Cogniac: High precision coreference with limited knowledge and linguistic resources. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*. Association for Computational Linguistics, pages 38–45.

Krisztian Balog, Edgar Meij, and Maarten de Rijke. 2010a. Entity search: Building bridges between two worlds. In *Proceedings of the 3rd International Semantic Search Workshop*. ACM, New York, NY, USA, SEMSEARCH '10, pages 9:1–9:5. https://doi.org/10.1145/1863879.1863888.

Krisztian Balog, Pavel Serdyukov, and Arjen P de Vries. 2010b. Overview of the trec 2010 entity track. Technical report, DTIC Document.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2012. Abstract meaning representation (amr) 1.0 specification. In *Parsing on Freebase from Question-Answer Pairs. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Seattle: ACL*. pages 1533–1544.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, IJCAI'07, pages 2670–2676. http://dl.acm.org/citation.cfm?id=1625275.1625705.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. 2002. Correlation clustering. In *Founda-*

tions of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on. IEEE, pages 238–247.

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the ACL/EACL 1997 Workshop on Intelligent Scalable Text Summarization*. pages 10–17. http://research.microsoft.com/en-us/um/people/cyl/download/papers/lexical-chains.pdf.

Cosmin Adrian Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '10, pages 1412–1422. http://dl.acm.org/citation.cfm?id=1858681.1858824.

Jonathan Berant and Percy Liang. 2015. Imitation learning of agenda-based semantic parsers. *Transactions of the Association for Computational Linguistics* 3:545–558.

Anders Björkelund, Love Hafdell, and Pierre Nugues. 2009. Multilingual semantic role labeling. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Stroudsburg, PA, USA, CoNLL '09, pages 43–48. http://dl.acm.org/citation.cfm?id=1596409.1596416.

Roi Blanco, Peter Mika, and Sebastiano Vigna. 2011. *Effective and Efficient Entity Search in RDF Data*, Springer Berlin Heidelberg, Berlin, Heidelberg, pages 83–97. https://doi.org/10.1007/978-3-642-25073-6_6.

Roi Blanco and Hugo Zaragoza. 2010. Finding support sentences for entities. In *Pro-*

ceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, SIGIR '10, pages 339–346. https://doi.org/10.1145/1835449.1835507.

David M. Blei. 2012. Probabilistic topic models. *Commun. ACM* 55(4):77–84. https://doi.org/10.1145/2133806.2133826.

David M. Blei and John D. Lafferty. 2006. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*. ACM, New York, NY, USA, ICML '06, pages 113–120. https://doi.org/10.1145/1143844.1143859.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3:993–1022. http://dl.acm.org/citation.cfm?id=944919.944937.

Daniel G. Bobrow and Terry Winograd. 1977. An overview of krl, a knowledge representation language. *Cognitive Science* 1(1):3–46. https://doi.org/10.1207/s15516709cog0101_2.

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

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013a. Translating embeddings for modeling multi-relational data. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, Curran Associates, Inc., pages 2787–2795. http://papers.nips.cc/paper/5071-translating-embeddings-for-modeling-multi-relational-data.pdf.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013b. Irreflexive and hierarchical relations as translations. *CoRR* abs/1304.7158. http://dblp.uni-trier.de/db/journals/corr/corr1304.html#abs-1304-7158.

Ronald J. Brachman and James G. Schmolze. 1985. An overview of the kl-one knowledge representation system. *Cognitive Science* 9(2):171–216. Http://www.cogsci.rpi.edu/CSJarchive/1985v09/i02/p0171p0216/MAIN.PDF. http://www.sciencedirect.com/science/article/B6W48-4DXC46N-S/1/60b9d9d36d07bb395e7229a36183dcea.

Eric Brill. 1992. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, ANLC '92, pages 152–155. https://doi.org/10.3115/974499.974526.

Katy Brner. 2010. *Atlas of Science: Visualizing What We Know*. The MIT Press.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Comput. Linguist.* 19(2):263–311. http://dl.acm.org/citation.cfm?id=972470.972474.

Razvan Bunescu and Raymond Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association* *of Computational Linguistics*. Association for Computational Linguistics, Prague, Czech Republic, pages 576–583. http://acl.ldc.upenn.edu/P/P07/P07-1073.pdf.

Razvan Bunescu and Marius Paşca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06), Trento, Italy*. pages 9–16. http://www.cs.utexas.edu/ ml/publication/paper.cgi?paper=encyc-eacl-06.ps.gz.

James P. Callan. 1994. Passage-level evidence in document retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Springer-Verlag New York, Inc., New York, NY, USA, SIGIR '94, pages 302–310. http://dl.acm.org/citation.cfm?id=188490.188589.

Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. 2009. Clueweb09 data set.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, SIGIR '98, pages 335–336. https://doi.org/10.1145/290941.291025.

Andrew Carnie. 2006. *Syntax: a Generative Introduction*. John Wiley & Sons.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Stroudsburg, PA, USA, CONLL '05, pages 152–164. http://dl.acm.org/citation.cfm?id=1706543.1706571.

Nathanael Chambers. 2013. Event schema induction with a probabilistic entity-driven model. In *EMNLP*. ACL, pages 1797–1807. http://dblp.uni-trier.de/db/conf/emnlp/emnlp2013.html#Chambers13.

Nathanael Chambers, Taylor Cassidy, Bill McDowell, and Steven Bethard. 2014. Dense event ordering with a multi-pass architecture. *Transactions of the Association for Computational Linguistics* 2.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '09, pages 602–610. http://dl.acm.org/citation.cfm?id=1690219.1690231.

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pages 976–986. http://dl.acm.org/citation.cfm?id=2002472.2002595.

Nathanael Chambers and Daniel Jurafsky. 2008. Unsupervised learning of narrative event chains. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*. pages 789–797. http://www.aclweb.org/anthology/P08-1090.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daume, and John Langford. 2015. Learning to search better than your teacher. In David Blei and Francis Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*. JMLR Workshop and Conference Proceedings, pages 2058–2066. http://jmlr.org/proceedings/papers/v37/changb15.pdf.

Kevin C. Chang. 2007. Entity search engine: Towards agile best-effort information integration over the web. In *Entity Search Engine: Towards Agile Best-Effort Information Integration over the Web*. http://www-db.cs.wisc.edu/cidr/cidr2007/papers/cidr07p12.pdf.

Eugene Charniak. 1977. A framed painting: The representation of a common sense knowledge fragment. *Cognitive Science* pages 355–394.

Tongfei Chen and Benjamin Van Durme. 2017. Discriminative information retrieval for question answering sentence selection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, Valencia, Spain, pages 719–725. http://www.aclweb.org/anthology/E17-2114.

Gong Cheng, Thanh Tran, and Yuzhong Qu. 2011. *RELIN: Relatedness and Informativeness-Based Centrality for Entity Summarization*, Springer Berlin Heidelberg, Berlin, Heidelberg, pages 114–129.

Tao Cheng, Xifeng Yan, and Kevin Chen-Chuan Chang. 2007. Supporting entity search: A large-scale prototype search engine. In *Proceedings of the 2007 ACM SIGMOD Interna-*

tional Conference on Management of Data. ACM, New York, NY, USA, SIGMOD '07, pages 1144–1146. https://doi.org/10.1145/1247480.1247636.

Xiao Cheng and Dan Roth. 2013. Relational inference for wikification. In *In EMNLP*.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*. pages 837–846. http://aclweb.org/anthology/N/N13/N13-1104.pdf.

Shruti Chhabra. 2014. Entity-centric summarization: Generating text summaries for graph snippets. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, New York, NY, USA, WWW '14 Companion, pages 33–38. https://doi.org/10.1145/2567948.2567959.

Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss, and Shivakumar Vaithyanathan. 2010. Domain adaptation of rule-based annotators for named-entity recognition tasks. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '10, pages 1002–1012. http://dl.acm.org/citation.cfm?id=1870658.1870756.

Jinho D Choi and Martha Palmer. 2011. Transition-based semantic role labeling using predicate argument clustering. In *Proceedings of the ACL 2011 Workshop on Relational Models of Semantics*. Association for Computational Linguistics, pages 37–45.

Noam Chomsky. 1981. Lectures on government and binding.

Jason Chuang, Yuening Hu, Ashley Jin andJohn D. Wilkerson, Daniel A. McFarland, Christopher D. Manning, and Jeffrey Heer. 2013. Document exploration with topic modeling: Designing interactive visualizations to support effective analysis workflows. In *NIPS Workshop on Topic Models: Computation, Application and Evaluation*.

Wesley T. Chuang and Jihoon Yang. 2000. Extracting sentence segments for text summarization: A machine learning approach. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, SIGIR '00, pages 152–159. https://doi.org/10.1145/345508.345566.

Kenneth W. Church. 2000. Empirical estimates of adaptation: The chance of two noriegas is closer to p/2 than p2. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '00, pages 180–186. https://doi.org/10.3115/990820.990847.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.* 16(1):22–29. http://dl.acm.org/citation.cfm?id=89086.89095.

Trevor Cohn, Chris Callison-Burch, and Mirella Lapata. 2008. Constructing corpora for the development and evaluation of paraphrase systems. *Comput. Linguist.* 34(4):597–614. https://doi.org/10.1162/coli.08-003-R1-07-044.

Trevor Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *J. Artif. Int. Res.* 34(1):637–674. http://dl.acm.org/citation.cfm?id=1622716.1622733.

Michael Collins and Brian Roark. 2004. Incremental parsing with the perceptron algorithm. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '04. https://doi.org/10.3115/1218955.1218970.

Graham Cormode and S. Muthukrishnan. 2005. An improved data stream summary: The count-min sketch and its applications. *J. Algorithms* 55(1):58–75. https://doi.org/10.1016/j.jalgor.2003.12.001.

Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. 2013. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22Nd International Conference on World Wide Web*. ACM, New York, NY, USA, WWW '13, pages 249–260. https://doi.org/10.1145/2488388.2488411.

Silviu Cucerzan. 2007. Large-scale named entity disambiguation based on Wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, Prague, Czech Republic, pages 708–716. http://www.aclweb.org/anthology/D/D07/D07-1074.

Jeffrey Dalton and Laura Dietz. 2013. Constructing query-specific knowledge bases. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*. ACM, New York, NY, USA, AKBC '13, pages 55–60. https://doi.org/10.1145/2509558.2509568.

Hoa Trang Dang and Karolina Owczarzak. 2008. Overview of the tac 2008 update summarization task. In *TAC*.

Dipanjan Das, André F. T. Martins, and Noah A. Smith. 2012. An exact dual decomposition algorithm for shallow semantic parsing with constraints. In *SemEval*. Association for Computational Linguistics, SemEval '12. http://dl.acm.org/citation.cfm?id=2387636.2387671.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith. 2010. Probabilistic frame-semantic parsing. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*. Association for Computational Linguistics, pages 948–956.

Hal Daumé. 2007. Frustratingly easy domain adaptation. In *Annual meeting-association for computational linguistics*. volume 45, page 256.

Hal Daumé III, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine Learning Journal (MLJ)* http://pub.hal3.name/#daume06searn.

Hal Daumé III and Daniel Marcu. 2005. Learning as search optimization: Approximate large margin methods for structured prediction. In *International Conference on Machine Learning (ICML)*. Bonn, Germany. http://hal3.name/docs/#daume05laso.

Thomas H Davenport and John C Beck. 2001. *The Attention Economy: Understanding the New Currency of Business*. Harvard Business Press.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE* 41(6):391–407.

Laura Dietz and Michael Schuhmacher. 2015. An interface sketch for queripidia: Query-driven knowledge portfolios from the web. In Krisztian Balog, Jeffrey Dalton, Antoine

Doucet, and Yusra Ibrahim, editors, *Proceedings of the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval, ESAIR 2015, Melbourne, Australia, October 23, 2015*. ACM, pages 43–46. https://doi.org/10.1145/2810133.2810145.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *LREC*. volume 2, page 1.

Pedro Domingos. 2015. *The Master Algorithm: How the Quest for the Ultimate Learning Machine will Remake our World*. Basic Books.

Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang, Wilko Horn, Camillo Lugaresi, Shaohua Sun, and Wei Zhang. 2015. Knowledge-based trust: Estimating the trustworthiness of web sources. *Proceedings of the VLDB Endowment* 8(9):938–949.

Doug Downey, Oren Etzioni, and Stephen Soderland. 2010. Analysis of a probabilistic model of redundancy in unsupervised information extraction. *Artificial Intelligence* 174(11):726 – 748. https://doi.org/http://dx.doi.org/10.1016/j.artint.2010.04.024.

David Dowty. 1991. Thematic proto-roles and argument selection. *language* pages 547–619.

Gregory Durrett. 2016. *Identifying and Resolving Entities in Text*. Ph.D. thesis, University of California, Berkeley.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie Strassel. 2015. Overview of linguistic resources for the tac kbp 2015 evaluations: Methodologies and results. In *Proceedings of TAC KBP 2015 Workshop, National Institute of Standards and Technology*. pages 16–17.

Susan T Ennett and Karl E Bauman. 1993. Peer group structure and adolescent cigarette smoking: a social network analysis. *Journal of Health and Social Behavior* pages 226–236.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.* 22(1):457–479. http://dl.acm.org/citation.cfm?id=1622487.1622501.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artif. Intell.* 165(1):91–134. https://doi.org/10.1016/j.artint.2005.03.001.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '11, pages 1535–1545. http://dl.acm.org/citation.cfm?id=2145432.2145596.

Francis Ferraro, Max Thomas, Matthew R. Gormley, Travis Wolfe, Craig Harman, and Benjamin Van Durme. 2014. Concretely annotated corpora. In *The NIPS 2014 AKBC Workshop*.

Francis Ferraro and Benjamin Van Durme. 2016. A Unified Bayesian Model of Scripts, Frames and Language. In *AAAI*.

David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine* 31(3):59–79.

Charles Fillmore. 1982. Frame semantics. *Linguistics in the morning calm* .

Charles J Fillmore. 1967. The case for case. In Emmon Bach and Robert T. Harms, editors, *Universals in Linguistic Theory*.

Charles J. Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech* 280(1):20–32.

Tim Finin, Dawn Lawrie, Paul McNamee, James Mayfield, Douglas Oard, Nanyun Peng, Ning Gao, Yiu-Chang Lin, Josh MacLin, and Tim Dowd. 2015. Hltcoe participation in tac kbp 2015: Cold start and tedl. In *Text Analytics Conference (TAC)*.

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic role labelling with neural network factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisboa, Portugal.

John R Frank, Steven J Bauer, Max Kleiman-Weiner, Daniel A Roberts, Nilesh Tripuraneni, Ce Zhang, Christopher Re, Ellen Voorhees, and Ian Soboroff. 2013. Evaluating stream filtering for entity profile updates for trec 2013 (kba track overview). Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE.

Lea Frermann, Ivan Titov, and Manfred Pinkal. 2014. A hierarchical bayesian model for unsupervised induction of script knowledge. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-*

30, 2014, Gothenburg, Sweden. pages 49–57. http://aclweb.org/anthology/E/E14/E14-1006.pdf.

Yoav Freund and Robert E Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine learning* 37(3):277–296.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. Facc1: Freebase annotation of clueweb corpora, version 1 (release date 2013-06-26, format version 1, correction level 0). *Note: http://lemurproject. org/clueweb09/FACC1/Cited by* 5.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*. Association for Computational Linguistics, Atlanta, Georgia, pages 758–764. http://cs.jhu.edu/ ccb/publications/ppdb.pdf.

Ning Gao and Silviu Cucerzan. 2017. Entity linking to one thousand knowledge bases. In *Advances in Information Retrieval - 39th European Conference on IR Research, ECIR 2017, Aberdeen, UK, April 8-13, 2017, Proceedings*. pages 1–14. https://doi.org/10.1007/978-3-319-56608-5_1.

Ning Gao, Mark Dredze, and Douglas Oard. 2017. Person entity linking in email with nil detection. *Journal of the Association for Information Science and Technology (JAIST)* .

Guillermo Garrido, Bernardo Cabaleiro, Anselmo Penas, Alvaro Rodrigo, and Damiano Spina. 2011. a distant supervised learning system for the tac-kbp slot filling and temporal slot filling tasks. In *TAC*.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational linguistics* 28(3):245–288.

Dan Gillick and Benoit Favre. 2009. A scalable global model for summarization. In *Proceedings of the Workshop on Integer Linear Programming for Natural Langauge Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, ILP '09, pages 10–18. http://dl.acm.org/citation.cfm?id=1611638.1611640.

Namrata Godbole, Manja Srinivasaiah, and Steven Skiena. 2007. Large-scale sentiment analysis for news and blogs. *ICWSM* 7(21):219–222.

A. E. Goldberg. 1995. *Constructions: a Construction Grammar Approach to Argument Structure*. University of Chicago Press, Chicago. http://groups.lis.illinois.edu/amag/langev/paper/goldberg95constructionsA.html.

Yoav Goldberg and Michael Elhadad. 2010. An efficient algorithm for easy-first non-directional dependency parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '10, pages 742–750. http://dl.acm.org/citation.cfm?id=1857999.1858114.

Shu Gong, Youli Qu, and Shengfeng Tian. 2010. Summarization using wikipedia. In *TAC*.

Jonathan Gordon and Benjamin Van Durme. 2013. Reporting bias and knowledge extraction. In *Automated Knowledge Base Construction (AKBC) 2013: The 3rd Workshop on Knowledge Extraction, at CIKM 2013*. AKBC '13.

Ralph Grishman and Beth Sundheim. 1996. Message understanding conference-6: A brief history. In *Coling*. volume 96, pages 466–471.

Kalpa Gunaratna, Krishnaprasad Thirunarayan, and Amit P. Sheth. 2015. FACES: diversity-aware entity summarization using incremental hierarchical conceptual clustering. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*. pages 116–122. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9562.

David Gunning. 2017. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency (DARPA), nd Web* .

Zhou GuoDong and Su Jian. 2004. A high-performance coreference resolution system using a constraint-based multi-agent strategy. In *Proceedings of the 20th International Conference on Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '04. https://doi.org/10.3115/1220355.1220430.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. 2013. Evaluating entity linking with wikipedia. *Artif. Intell.* 194:130–150. https://doi.org/10.1016/j.artint.2012.04.005.

Aria Haghighi and Dan Klein. 2009. Simple coreference resolution with rich syntactic and semantic features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '09, pages 1152–1161. http://dl.acm.org/citation.cfm?id=1699648.1699661.

Aria Haghighi, Kristina Toutanova, and Christopher D. Manning. 2005. A joint model for semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural*

Language Learning. Association for Computational Linguistics, Stroudsburg, PA, USA, CONLL '05, pages 173–176. http://dl.acm.org/citation.cfm?id=1706543.1706574.

Jan Hajič. 2017. Results: Treebanks ranked by best las f1. http://universaldependencies.org/conll17/results-treebanks.html.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL-2009), June 4-5*. Boulder, Colorado, USA.

A. Y. Halevy, Z. G. Ives, D. Suciu, and I. Tatarinov. 2003. Schema mediation in peer data management systems. In *Proceedings 19th International Conference on Data Engineering (Cat. No.03CH37405)*. pages 505–516. https://doi.org/10.1109/ICDE.2003.1260817.

Xianpei Han and Le Sun. 2012. An entity-topic model for entity linking. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, pages 105–115.

Xianpei Han, Le Sun, and Jun Zhao. 2011. Collective entity linking in web text: A graph-based method. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, SIGIR '11, pages 765–774. https://doi.org/10.1145/2009916.2010019.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '04. https://doi.org/10.3115/1218955.1219008.

Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. Trueskill$^{\text{TM}}$: A bayesian skill rating system. In *Proceedings of the 19th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge, MA, USA, NIPS'06, pages 569–576. http://dl.acm.org/citation.cfm?id=2976456.2976528.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In *Proceedings of the 52th Annual Meeting of the Association for Computational Linguistics*.

Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* .

Jerry R. Hobbs. 1987. World knowledge and word meaning. In *Proceedings of the 1987 Workshop on Theoretical Issues in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, TINLAP '87, pages 20–27. https://doi.org/10.3115/980304.980308.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artif. Intell.* 194:28–61. https://doi.org/10.1016/j.artint.2012.06.001.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '11, pages 782–792. http://dl.acm.org/citation.cfm?id=2145432.2145521.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pages 541–550. http://dl.acm.org/citation.cfm?id=2002472.2002541.

Thomas Hofmann. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, New York, NY, USA, SIGIR '99, pages 50–57. https://doi.org/10.1145/312624.312649.

Matthew Honnibal. 2016. Syntaxnet in context: Understanding google's new tensorflow nlp model. https://explosion.ai/blog/syntaxnet-in-context.

Liang Huang, Suphan Fayong, and Yang Guo. 2012. Structured perceptron with inexact search. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL HLT '12, pages 142–151. http://dl.acm.org/citation.cfm?id=2382029.2382049.

Rodolphe Jenatton, Nicolas L. Roux, Antoine Bordes, and Guillaume R Obozinski. 2012. A latent factor model for highly multi-relational data. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, Curran Associates, Inc., pages 3167–3175. http://papers.nips.cc/paper/4744-a-latent-factor-model-for-highly-multi-relational-data.pdf.

Heng Ji, Ralph Grishman, and Hoa Dang. 2011. Overview of the TAC2011 knowledge base population track. In *TAC 2011 Proceedings Papers*.

Yuzhe Jin, Emre Kiciman, Kuansan Wang, and Ricky Loynd. 2014. Entity linking at the tail: Sparse signals, unknown entities, and phrase models. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. ACM, New York, NY, USA, WSDM '14, pages 453–462. https://doi.org/10.1145/2556195.2556230.

Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. 2009. Cutting-plane training of structural svms. *Mach. Learn.* 77(1):27–59. https://doi.org/10.1007/s10994-009-5108-8.

Richard Johansson and Pierre Nugues. 2008a. Dependency-based semantic role labeling of propbank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '08, pages 69–78. http://dl.acm.org/citation.cfm?id=1613715.1613726.

Richard Johansson and Pierre Nugues. 2008b. The effect of syntactic representation on semantic role labeling. In *Proceedings of the 22Nd International Conference on Computational Linguistics - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, COLING '08, pages 393–400. http://dl.acm.org/citation.cfm?id=1599081.1599131.

Anastasios Kementsietsidis, Marcelo Arenas, and Renée J. Miller. 2003. Mapping data in peer-to-peer systems: Semantics and algorithmic issues. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*. ACM, New York, NY, USA, SIGMOD '03, pages 325–336. https://doi.org/10.1145/872757.872798.

Paul Kingsbury and Martha Palmer. 2002. From treebank to propbank. In *LREC*. Citeseer.

Jon Kleinberg. 2002. Bursty and hierarchical structure in streams. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '02, pages 91–101. https://doi.org/10.1145/775047.775061.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: a new dataset for email classification research. In *European Conference on Machine Learning*. Springer, pages 217–226.

Kevin Knight and Daniel Marcu. 2002. Summarization beyond sentence extraction: a probabilistic approach to sentence compression. *Artificial Intelligence* 139(1):91 – 107. https://doi.org/http://dx.doi.org/10.1016/S0004-3702(02)00222-9.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*. Association for Computational Linguistics, Barcelona, Spain, pages 388–395.

Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. 2009. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '09, pages 457–466. https://doi.org/10.1145/1557019.1557073.

Nicholas Kushmerick. 1997. *Wrapper Induction for Information Extraction*. Ph.D. thesis, University of Washington. AAI9819266.

Simon Lacoste-Julien, Benjamin Taskar, Dan Klein, and Michael I. Jordan. 2006. Word alignment via quadratic assignment. In *HLT-NAACL*.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Comput. Linguist.* 39(4):885–916. https://doi.org/10.1162/COLI_a_00152.

Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*. Association for Computational Linguistics, Stroudsburg, PA, USA, CONLL Shared Task '11, pages 28–34. http://dl.acm.org/citation.cfm?id=2132936.2132938.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Pro-*

ceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP-CoNLL '12, pages 489–500. http://dl.acm.org/citation.cfm?id=2390948.2391006.

Yoonkyung Lee, Yi Lin, and Grace Wahba. 2004. Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association* 99:67–81.

Douglas B. Lenat, Mayank Prakash, and Mary Shepherd. 1986. CYC: using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Magazine* 6(4):65–85. http://www.aaai.org/ojs/index.php/aimagazine/article/view/510.

Beth Levin. 1993. *English Verb Classes and Alternations: a Preliminary Investigation*. University of Chicago Press.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In Stan Szpakowicz Marie-Francine Moens, editor, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. Association for Computational Linguistics, Barcelona, Spain, pages 74–81.

Fei Liu, Jeffrey Flanigan, Sam Thomson, Norman M. Sadeh, and Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. In Rada Mihalcea, Joyce Yue Chai, and Anoop Sarkar, editors, *HLT-NAACL*. The Association for Computational Linguistics, pages 1077–1086. http://dblp.uni-trier.de/db/conf/naacl/naacl2015.html#0004FTSS15.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM J. Res. Dev.* 2(2):159–165. https://doi.org/10.1147/rd.22.0159.

Pattie Maes. 1994. Agents that reduce work and information overload. *Commun. ACM* 37(7):30–40. https://doi.org/10.1145/176789.176792.

Inderjeet Mani and Eric Bloedorn. 1998. Machine learning of generic and user-focused summarization. In *AAAI/IAAI*. pages 821–826.

Gideon S. Mann and David Yarowsky. 2003. Unsupervised personal name disambiguation. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*. Association for Computational Linguistics, Stroudsburg, PA, USA, CONLL '03, pages 33–40. https://doi.org/10.3115/1119176.1119181.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. pages 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.* 19(2):313–330. http://dl.acm.org/citation.cfm?id=972470.972475.

Oded Maron and Tomas Lozano-Perez. 1998. A framework for multiple instance learning. In *Advances in Neural Information Processing Systems 10*. MIT Press, Cambridge, MA. http://lis.csail.mit.edu/pubs/tlp/maron98framework.pdf.

Rebecca Mason, Benjamin Gaska, Benjamin Van Durme, Pallavi Choudhury, Ted Hart, Bill Dolan, Kristina Toutanova, and Margaret Mitchell. 2016. Microsummarization of online reviews: An experimental study. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*. pages 3015–3021. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12548.

Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 2012. 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*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP-CoNLL '12, pages 523–534. http://dl.acm.org/citation.cfm?id=2390948.2391009.

David D. McDonald. 1978. Story understanding: the beginning of a consensus. Technical Report WORKING PAPER 168, ARTIFICIAL INTELLIGENCE LABORATORY MASSACHUSETTS INSTITUTE OF TECHNOLOGY.

Paul McNamee and Hoa Trang Dang. 2009. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC)*. volume 17, pages 111–113.

Paul McNamee, James Mayfield, Dawn Lawrie, Douglas W Oard, and David S Doermann. 2011. Cross-language entity linking. In *IJCNLP*. pages 255–263.

Paul McNamee, Veselin Stoyanov, James Mayfield, Tim Finin, Tim Oates, Tan Xu, Douglas W Oard, and Dawn Lawrie. 2012. Hltcoe participation at tac 2012: Entity linking and cold start knowledge base construction. In *TAC*.

Xinfan Meng, Furu Wei, Xiaohua Liu, Ming Zhou, Sujian Li, and Houfeng Wang. 2012. Entity-centric topic-oriented opinion summarization in twitter. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '12, pages 379–387. https://doi.org/10.1145/2339530.2339592.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. 2004. The nombank project: an interim report. In *HLT-NAACL 2004 workshop: Frontiers in corpus annotation*. volume 24, page 31.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*. ACM, New York, NY, USA, CIKM '07, pages 233–242. https://doi.org/10.1145/1321440.1321475.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

David Milne and Ian H. Witten. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. ACM, New York, NY, USA, CIKM '08, pages 509–518. https://doi.org/10.1145/1458082.1458150.

Thomas P. Minka. 2003. Bayesian inference, entropy, and the multinomial distribution. https://tminka.github.io/papers/minka-multinomial.pdf.

Marvin Minsky. 1974. A framework for representing knowledge. Technical report, Cambridge, MA, USA.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '09, pages 1003–1011. http://dl.acm.org/citation.cfm?id=1690219.1690287.

RV Mises and Hilda Pollaczek-Geiringer. 1929. Praktische verfahren der gleichungsauflösung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* 9(1):58–77.

Ashutosh Modi, Ivan Titov, and Alexandre Klementiev. 2012. Unsupervised induction of frame-semantic representations. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*. Association for Computational Linguistics, Stroudsburg, PA, USA, WILS '12, pages 1–7. http://dl.acm.org/citation.cfm?id=2390426.2390428.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme. 2012. Annotated gigaword. In *AKBC-WEKEX Workshop at NAACL 2012*.

Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch. 2011. Evaluating sentence compression: Pitfalls and suggested remedies. In *Proceed-*

ings of the Workshop on Monolingual Text-To-Text Generation. Association for Computational Linguistics, Stroudsburg, PA, USA, MTTG '11, pages 91–97. http://dl.acm.org/citation.cfm?id=2107679.2107690.

Ani Nenkova and Kathleen McKeown. 2012. *A Survey of Text Summarization Techniques*, Springer US, Boston, MA, pages 43–76. https://doi.org/10.1007/978-1-4614-3223-4_3.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.* 4(2). https://doi.org/10.1145/1233912.1233913.

Robert Neumayer, Krisztian Balog, and Kjetil Nørvåg. 2012. On the modeling of entities for ad-hoc entity search in the web of data. In *ECIR*. Springer, volume 12, pages 133–145.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan T McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *LREC*.

Peter Norvig. 1983. Frame activated inferences in a story understanding program. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*. Karlsrhue, Germany, pages 624–626.

Judith Reitman Olson and Henry H Rueter. 1987. Extracting expertise from experts: Methods for knowledge acquisition. *Expert systems* 4(3):152–168.

Martha Palmer. 2009. Semlink: Linking propbank, verbnet and framenet. In *Proceedings of the Generative Lexicon Conference*. GenLex-09, 2009 Pisa, Italy, pages 9–15.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Comput. Linguist.* 31(1):71–106. https://doi.org/10.1162/0891201053630264.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition, linguistic data consortium. Technical report, Technical report, Technical Report. Linguistic Data Consortium, Philadelphia.

Saša Petrović, Miles Osborne, and Victor Lavrenko. 2010. Streaming first story detection with application to twitter. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '10, pages 181–189. http://dl.acm.org/citation.cfm?id=1857999.1858020.

Peter Pirolli and Stuart Card. 1999. Information foraging. *Psychological review* 106(4):643.

Peter Pirolli and Stuart Card. 2005. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis pages 2–4.

Jeffrey Pound, Peter Mika, and Hugo Zaragoza. 2010. Ad-hoc object retrieval in the web of data. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, New York, NY, USA, WWW '10, pages 771–780. https://doi.org/10.1145/1772690.1772769.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural*

Language Learning. Sofia, Bulgaria: Association for Computational Linguistics. pages 143–52.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012)*. Jeju, Korea.

Ellen F. Prince. 1998. On the limits of syntax, with reference to left-dislocation and topicalization.

Vasin Punyakanok, Dan Roth, Wen tau Yih, Dav Zimak, and Yuancheng Tu. 2004. Semantic role labeling via generalized inference over classifiers. In *In: Proc. of the 8th Conference on Natural Language Learning (CoNLL-2004)*. pages 130–133.

James Pustejovsky, José Castaño, Robert Ingria, Roser Saurí, Robert Gaizauskas, Andrea Setzer, and Graham Katz. 2003. Timeml: Robust specification of event and temporal expressions in text. In *in Fifth International Workshop on Computational Semantics (IWCS-5)*.

Hema Raghavan, James Allan, and Andrew McCallum. 2004. An exploration of entity models, collective classification and relation description. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’04.

Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural*

Language Processing. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '10, pages 492–501. http://dl.acm.org/citation.cfm?id=1870658.1870706.

Delip Rao, Paul McNamee, and Mark Dredze. 2010. Streaming cross document entity coreference resolution. In *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*. pages 1050–1058. http://aclweb.org/anthology-new/C/C10/C10-2121.pdf.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pages 1375–1384. http://dl.acm.org/citation.cfm?id=2002472.2002642.

Marta Recasens, Matthew Can, and Daniel Jurafsky. 2013. Same referent, different words: Unsupervised mining of opaque coreferent mentions. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Atlanta, Georgia, pages 897–906. http://www.aclweb.org/anthology/N13-1110.

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. 2015. Semantic proto-roles. *Transactions of the Association for Computational Linguistics* 3:475–488.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22Nd ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining. ACM, New York, NY, USA, KDD '16, pages 1135–1144. https://doi.org/10.1145/2939672.2939778.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *HLT-NAACL*. The Association for Computational Linguistics, pages 74–84.

Stéphane Ross, Geoffrey J. Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey J. Gordon and David B. Dunson, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*. Journal of Machine Learning Research - Workshop and Conference Proceedings, volume 15, pages 627–635. http://www.jmlr.org/proceedings/papers/v15/ross11a/ross11a.pdf.

Michael Roth. 2013. *Inducing Implicit Arguments via Cross-document Alignment: a Framework and its Applications*. Ph.D. thesis, Heidelberg University. http://d-nb.info/1054050406.

Michael Roth and Anette Frank. 2012. Aligning predicate argument structures in monolingual comparable texts: a new corpus for a new task. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for

Computational Linguistics, Stroudsburg, PA, USA, SemEval '12, pages 218–227. http://dl.acm.org/citation.cfm?id=2387636.2387672.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1192–1202. http://www.aclweb.org/anthology/P16-1113.

Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. 2015. Script induction as language modeling. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *EMNLP*. The Association for Computational Linguistics, pages 1681–1686. http://dblp.uni-trier.de/db/conf/emnlp/emnlp2015.html#RudingerRFD15.

Josef Ruppenhofer, Michael Ellsworth, Miriam R.L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2006. *FrameNet II: Extended Theory and Practice*. International Computer Science Institute, Berkeley, California. Distributed with the FrameNet data.

Daniel M. Russell, Mark J. Stefik, Peter Pirolli, and Stuart K. Card. 1993. The cost structure of sensemaking. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, CHI '93, pages 269–276. https://doi.org/10.1145/169059.169209.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme. 2014. Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Work-*

shop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA. pages 1–11. http://aclweb.org/anthology/W/W14/W14-3301.pdf.

Gerard Salton, Chris Buckley, and James Allan. 1991. Automatic structuring of text files. Technical report, Cornell University.

Gerard Salton and Michael J. McGill. 1986. Introduction to modern information retrieval .

Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley. 1997. Automatic text structuring and summarization. *Inf. Process. Manage.* 33(2):193–207. https://doi.org/10.1016/S0306-4573(96)00062-3.

Christina Sauper and Regina Barzilay. 2009. Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '09, pages 208–216. http://dl.acm.org/citation.cfm?id=1687878.1687909.

Stefan Schaal. 1999. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences* 3(6):233–242.

Roger C. Schank. 1975. Using knowledge to understand. In *Proceedings of the 1975 Workshop on Theoretical Issues in Natural Language Processing*. Association for Computational Linguistics, Stroudsburg, PA, USA, TINLAP '75, pages 117–121. https://doi.org/10.3115/980190.980223.

Roger C. Schank and Robert P. Abelson. 1977. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. L. Erlbaum, Hillsdale, NJ.

Dafna Shahaf and Carlos Guestrin. 2010. Connecting the dots between news articles. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '10, pages 623–632. https://doi.org/10.1145/1835804.1835884.

Dafna Shahaf, Carlos Guestrin, and Eric Horvitz. 2012. Metro maps of science. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, New York, NY, USA, KDD '12, pages 1122–1130. https://doi.org/10.1145/2339530.2339706.

Dafna Shahaf, Jaewon Yang, Caroline Suen, Jeff Jacobs, Heidi Wang, and Jure Leskovec. 2013. Information cartography: Creating zoomable, large-scale maps of information. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, USA, KDD '13, pages 1097–1105. https://doi.org/10.1145/2487575.2487690.

Libin Shen, Giorgio Satta, and Aravind Joshi. 2007. Guided learning for bidirectional sequence classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, Prague, Czech Republic, pages 760–767. http://www.aclweb.org/anthology/P07-1096.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2011. Large-scale cross-document coreference using distributed inference and hierarchical models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pages 793–803. http://dl.acm.org/citation.cfm?id=2002472.2002573.

Sameer Singh, Limin Yao, David Belanger, Ari Kobren, Sam Anzaroot, Mike Wick, Alexandre Passos, Harshal Pandya, Jinho D Choi, Brian Martin, et al. 2013. Universal schema for slot filling and cold start: Umass iesl at tackbp 2013. In *TAC*.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, MIT Press, pages 1297–1304. http://papers.nips.cc/paper/2659-learning-syntactic-patterns-for-automatic-hypernym-discovery.pdf.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems (NIPS)*.

Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S Weld. 2013. Open information extraction to kbp relations in 3 hours. In *TAC*.

Zhiyi Song, Ann Bies, Stephanie Strassel, Tom Riese, Justin Mott, Joe Ellis, Jonathan Wright, Seth Kulick, Neville Ryant, and Xiaoyi Ma. 2015. From light to rich ere: Annotation of entities, relations, and events. In *Proceedings of the 3rd Workshop on EVENTS at the NAACL-HLT*. pages 89–98.

Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. 2001. A machine learning approach to coreference resolution of noun phrases. *Comput. Linguist.* 27(4):521–544. http://dl.acm.org/citation.cfm?id=972597.972602.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*. ACM, New York, NY, USA, WWW '07, pages 697–706. https://doi.org/10.1145/1242572.1242667.

Ang Sun, Ralph Grishman, Wei Xu, and Bonan Min. 2011. New york university 2011 system for kbp slot filling. In *TAC*.

Beth M Sundheim. 1996. Overview of results of the muc-6 evaluation. In *Proceedings of a workshop on held at Vienna, Virginia: May 6-8, 1996*. Association for Computational Linguistics, pages 423–442.

Beth M. Sundheim and Nancy A. Chinchor. 1993. Survey of the message understanding conferences. In *Proceedings of the Workshop on Human Language Technology*. Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '93, pages 56–60. https://doi.org/10.3115/1075671.1075684.

Mihai Surdeanu. 2013. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *TAC*.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In

Proceedings of the Twelfth Conference on Computational Natural Language Learning. Association for Computational Linguistics, pages 159–177.

Mihai Surdeanu, David McClosky, Julie Tibshirani, John Bauer, Angel X Chang, Valentin I Spitkovsky, and Christopher D Manning. 2010. A simple distant supervision approach for the tac-kbp slot filling task. In *TAC*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 2012. 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*. Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP-CoNLL '12, pages 455–465. http://dl.acm.org/citation.cfm?id=2390948.2391003.

Ilya Sutskever, Joshua B. Tenenbaum, and Ruslan R Salakhutdinov. 2009. Modelling relational data using bayesian clustered tensor factorization. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, Curran Associates, Inc., pages 1821–1828. http://papers.nips.cc/paper/3863-modelling-relational-data-using-bayesian-clustered-tensor-factorization.pdf.

Richard S Sutton and Andrew G Barto. 1998. *Reinforcement Learning: an Introduction*. MIT press Cambridge.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation.

In *Proceedings of the 12th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge, MA, USA, NIPS'99, pages 1057–1063. http://dl.acm.org/citation.cfm?id=3009657.3009806.

R. Swan and D. Jensen. 2000. Timemines: Constructing timelines with statistical models of word usage.

Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Efficient inference and structured learning for semantic role labeling. *Transactions of the Association for Computational Linguistics* 3:29–41.

Ben Taskar, Carlos Guestrin, and Daphne Koller. 2003. Max-margin markov networks. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge, MA, USA, NIPS'03, pages 25–32. http://dl.acm.org/citation.cfm?id=2981345.2981349.

Andreas Thalhammer, Magnus Knuth, and Harald Sack. 2012. *Evaluating Entity Summarization Using a Game-Based Ground Truth*, Springer Berlin Heidelberg, Berlin, Heidelberg, pages 350–361.

Ivan Titov and Ehsan Khoddam. 2015. Unsupervised induction of semantic roles within a reconstruction-error minimization framework. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Denver, Colorado, pages 1–10. http://www.aclweb.org/anthology/N15-1001.

Ivan Titov and Alexandre Klementiev. 2012. A bayesian approach to unsupervised semantic role induction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, EACL '12, pages 12–22. http://dl.acm.org/citation.cfm?id=2380816.2380821.

Kristina Toutanova, Aria Haghighi, and Christopher D. Manning. 2008. A global joint model for semantic role labeling. *Comput. Linguist.* 34(2):161–191. https://doi.org/10.1162/coli.2008.34.2.161.

Stephen Tratz and Eduard Hovy. 2008. Summarisation evaluation using transformed basic elements. In *Proceedings TAC 2008*. NIST, page 10 pages. http://www.nist.gov/tac/publications/2008/additional.papers/ISI.proceedings.pdf.

Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. 2013. Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Association for Computational Linguistics, Atlanta, Georgia, USA, pages 1–9. http://www.aclweb.org/anthology/S13-2001.

Ashish Vaswani and Kenji Sagae. 2016. Efficient structured inference for transition-based parsing with neural networks and error states. *Transactions of the Association for Computational Linguistics* 4:183–196.

Patrick Verga, David Belanger, Emma Strubell, Benjamin Roth, and Andrew McCallum. 2016. Multilingual relation extraction using compositional universal schema. In *NAACL*

HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016. pages 886–896. http://aclweb.org/anthology/N/N16/N16-1103.pdf.

Svitlana Volkova, Glen Coppersmith, and Benjamin Van Durme. 2014. Inferring user political preferences from streaming communications. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*. pages 186–196. http://aclweb.org/anthology/P/P14/P14-1018.pdf.

Xuerui Wang and Andrew Mccallum. 2006. Topics over time: a non-markov continuous-time model of topical trends. In *in SIGKDD*.

Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. Ontonotes: a large training corpus for enhanced processing. *Handbook of Natural Language Processing and Machine Translation. Springer* .

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. 2016. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, pages 1713–1723. https://aclweb.org/anthology/D16-1177.

Michael Wick, Sameer Singh, and Andrew McCallum. 2012. A discriminative hierarchical model for fast coreference at large scale. In *Proceedings of the 50th Annual Meeting* *of the Association for Computational Linguistics: Long Papers - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '12, pages 379–388. http://dl.acm.org/citation.cfm?id=2390524.2390578.

Michael Wick, Sameer Singh, Harshal Pandya, and Andrew McCallum. 2013. A joint model for discovering and linking entities. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*. ACM, New York, NY, USA, AKBC '13, pages 67–72. https://doi.org/10.1145/2509558.2509570.

Robert Wilensky. 1978. *Understanding Goal-Based Stories*. Ph.D. thesis, Yale University. Computer Science Department Technical Report 140.

Sam Wiseman and Alexander M. Rush. 2016. Sequence-to-sequence learning as beam-search optimization. *CoRR* abs/1606.02960. http://arxiv.org/abs/1606.02960.

Travis Wolfe, Mark Dredze, and Benjamin Van Durme. 2015. Predicate argument alignment using a global coherence model. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings*.

Travis Wolfe, Mark Dredze, and Benjamin Van Durme. 2016a. Pocket knowledge base population. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '16.

Travis Wolfe, Mark Dredze, and Benjamin Van Durme. 2016b. A study of imitation learning methods for semantic role labeling. In *Proc. of the EMNLP Workshop on Structured Prediction for NLP*. page 44.

Travis Wolfe, Mark Dredze, and Benjamin Van Durme. 2017. Distantly supervised entity summarizatoin. In *in submission*.

Travis Wolfe, Benjamin Van Durme, Mark Dredze, Nicholas Andrews, Charley Bellar, Chris Callison-Burch, Jay DeYoung, Justin Snyder, Jonathann Weese, Tan Xu, and Xuchen Yao. 2013. Parma: A predicate argument aligner. In *Proceedings of the 51th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *EMNLP*.

Xuchen Yao. 2014. *Feature-driven Question Answering With Natural Language Alignment*. Ph.D. thesis, Johns Hopkins University.

Xuchen Yao. 2015. Lean question answering over freebase from scratch. In *Proceedings of NAACL Demo*.

Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of ACL*.

Xuchen Yao, Benjamin Van Durme, Peter Clark, and Chris Callison-Burch. 2013. Answer extraction as sequence tagging with tree edit distance. In *Proceedings of NAACL*.

David Yarowsky. 1993. One sense per collocation. In *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, pages 266–271.

Scott Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. ACL – Association for Computational Linguistics. https://www.microsoft.com/en-us/research/publication/semantic-parsing-via-staged-query-graph-generation-question-answering-with-knowledge-base/.

Omar F. Zaidan, Jason Eisner, and Christine Piatko. 2007. Using "annotator rationales" to improve machine learning for text categorization. In *NAACL HLT 2007; Proceedings of the Main Conference*. pages 260–267.

K. Zhang and D. Shasha. 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.* 18(6):1245–1262.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*. pages 1127–1137. http://aclweb.org/anthology/P/P15/P15-1109.pdf.

Chunyao Zou and Daqing Hou. 2014. Lda analyzer: A tool for exploring topic models. In *ICSME*. IEEE Computer Society, pages 593–596.

Vita

Travis Wolfe was born in Trenton, New Jersey on January 18^{th}, 1989. He attended Carnegie Mellon University from 2007 to 2011, receiving a Bachelors of Science in Statistics and Information Systems. He began graduate studies at Johns Hopkins University in 2011 and received a Masters of Science in 2014 working in the area of computer science and natural language processing. During the summer of 2014 he was an intern performing research at Google with Marius Paşca.