Extracting Knowledge from Twitter and The Web

Alan Ritter

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

Doctor of Philosophy

University of Washington

2012

Reading Committee:

Oren Etzioni, Chair

Mausam, Chair

Luke Zettlemoyer

Program Authorized to Offer Degree:
Computer Science & Engineering

University of Washington

Abstract

Extracting Knowledge from Twitter and The Web

Alan Ritter

Co-Chairs of the Supervisory Committee:
Professor Oren Etzioni
Computer Science & Engineering

Professor Mausam
Computer Science & Engineering

The internet has revolutionized the way we communicate, leading to a constant flood of text in electronic format, including the Web, email, SMS and the short informal texts found in microblogs such as Twitter. This presents a big opportunity for Natural Language Processing (NLP) and Information Extraction (IE) technology to enable new large scale data-analysis applications by extracting machine-processable information from unstructured text at scale. This thesis discusses the challenges and opportunities which arise when applying NLP and IE to large open-domain and *heterogeneous* text corpora such as Twitter and the Web, and presents solutions to a number of issues which arise in this setting.

Good performance is achieved using a mostly supervised approach in cases where the number of output labels is small and well-balanced. We build a set of low-level syntactic annotation tools for noisy informal Twitter text including a POS tagger, shallow parser, named entity segmenter and event recognizer using supervised learning techniques trained on an annotated corpus of tweets.

Supervised learning becomes impractical however, for semantic processing tasks such as: named entity categorization, event categorization, inferring selectional preferences and relation extraction, where the number of output labels is large and/or unknown a-priori. A key hypothesis which is evaluated throughout this thesis is that semantic processing of massive, diverse text corpora such as Twitter and the Web requires unsupervised and

weakly supervised methods which can leverage large *unlabeled* datasets for learning, rather than relying on the relatively small corpora which are feasible to annotate. We present a set of techniques for unsupervised and weakly-supervised information extraction based on *probabilistic latent variable models*, which are applied to infer the semantics of large numbers of words and phrases and extract knowledge from large open-domain text corpora.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ACKNOWLEDGMENTS

I am indebted to all the excellent colleagues and mentors that I've interacted with over the years, without whose support my success would not have been possible.

During my time at UW I have had the fortune to have many excellent mentors from whom I learned how to choose worthwhile problems to work on and also how to make progress on them, starting with my advisers, Mausam and Oren Etzioni, and also Stephen Soderland. I was lucky to have the opportunity to do two internships at Microsoft Research, both of which were amazing experiences for me. Again I was lucky to have excellent mentors: Sumit Basu, Colin Cherry and Bill Dolan, whose continued advice and encouragement has been invaluable to me. My work has also benefited from many interactions with other researchers and interns at MSR. In addition I would like to thank Luke Zettlemoyer. Luke is incredibly knowledgeable, and I feel like he has really gone out of his way to discuss research and mentor me, despite his busy schedule. I would also like to thank Dan Weld from whom I benefited greatly from from interactions with. In addition I was lucky to have large number of incredibly bright current and former students and postdocs at UW as colleagues including: Hoifung Poon, Wei Xu, Sam Clark, Fei Wu, Raphael Hoffmann, Yoav Artzi, Janara Christensen, Jesse Davis, Doug Downey, Tony Fader, Brian Hutchinson, Vibhav Gogate, Jeff Huang, Thomas Lin, Xiao Ling, Nicholas FitzGerald, Tom Kwiatkowski, Stefan Schoenmackers, Adrienne Wang, Alex Yates and Mark Yatskar. Interactions and discussions with all of these people have helped to shape my ideas about how to do research.

I was also very fortunate to have two excellent mentors as an undergraduate at WWU: James Hearne and Philip Nelson, both of whom encouraged my interest in pursuing research and also opened up many opportunities for me.

Finally I would like to thank Patrick Allen and Lindsay Michimoto who have gone out of their way to support me throughout graduate school.

Chapter 1

INTRODUCTION

The internet has changed how we communicate. This has led to a flood of *user-generated* text available in electronic format including text contained in large heterogeneous text corpora such as the Web, Email, SMS and Twitter. The amount of informal, user-generated text written each day is much larger than the amount of formal, professionally written text such as that found in books and newspaper articles; for instance over 200 million Tweets are written each day.[1] Clearly no person can read all these messages motivating the need for automatic text processing techniques to extract, aggregate and organize this information.

A lot of high-value information turns up first in informal user-generated messages. One famous example is the Twitter user, @ReallyVirtual, who inadvertently live-tweeted the raid that killed Osama Bin Laden.[2] Although this Twitter user didn't know what was actually happening at the time, the first news that Bin Laden had been killed showed up on Twitter when Kieth Urbahn, a former chief of staff to Donald Rumsfeld tweeted at 10:25pm EST on May 1 2011:

> *"So Im told by a reputable person they have killed Osama Bin Laden. Hot damn."*

Accurate knowledge extraction from big, noisy text corpora such as Twitter and the Web has the potential to enable many new data analysis applications, however these corpora have unique characteristics which suggest it may be worth re-considering our overall approach to processing language. *This thesis proposes to address the challenges raised by these corpora through a series of weakly-supervised and unsupervised approaches to information extraction based on probabilistic latent variable models.*

---

[1] `http://blog.twitter.com/2011/06/200-million-tweets-per-day.html`

[2] `http://www.forbes.com/sites/parmyolson/2011/05/02/man-inadvertently-live-tweets-osama-bin-laden-raid/`

1.1 Twitter and Web Text: Characteristics

Massive, user-generated text corpora such as Twitter and the Web are *heterogeneous*. They are highly diverse in style as compared to more homogeneous corpora such as newswire which follow strict stylistic conventions. In addition to their heterogeneity of style, they also contain highly heterogeneous content. Because this text is user-generated, it can cover virtually any topic. Furthermore, due to the large, diverse and dynamically changing nature of these text collections, we often do not know what kinds of information will be of interest in advance. Supervised machine learning is less effective in this setting because even large annotated corpora are insufficient to capture the full range of style and content which is found in these massive, open-domain text corpora.

1.1.1 Challenges

Noisy Text

Noisy informal user-generated text such as that found on Twitter and the Web presents unique challenges for NLP and IE technology [44; 53]. As an instance of this problem, we demonstrate in Chapter 3 that existing NLP tools which are tuned to work on grammatical newswire text fail when applied to noisy Twitter text. To address this challenge, we annotate a corpus of tweets with parts of speech, chunk tags, named entities and events which are then used to train in-domain sequence labeling models. In addition we found that word clusters learned from a large unlabeled corpus of tweets using Jcluster [71] help to deal with Twitter's high degree of lexical variation. Full details of our approach and data are presented in Chapter 3.

Open-Domain

The text contained in large user-generated corpora such as Twitter and the Web can discuss almost any topic, so it is important not to limit our focus to a narrow set of possible relations in advance. The field of information extraction has traditionally followed an annotate-train-test paradigm, which begins with the design of annotation guidelines, followed by the collection and labeling of corpora[168]. Only then can one train machine learning models

to automatically extract new relation instances from text. This paradigm has been quite successful, but the labeling process is both slow and expensive, limiting the amount of data available for training. In addition this approach requires designing annotation guidelines and committing to a specific set of relations before investing resources into an annotation. To address these challenges, Open Information Extraction [60; 114] proposes to extract information from text without a pre-defined set of relations. This thesis follows the Open IE paradigm, avoiding pre-defined set relations and hand-labeled training data, by leveraging unsupervised and distantly supervised learning.

The annotate-train-test approach is appropriate when the set of categories to be predicted is small, well balanced in the data and fixed in advance, which is the case for most syntactic annotation tasks, such as part of speech tagging or named entity segmentation. Supervised learning becomes impractical however, for semantic processing tasks such as: named entity categorization (Chapter 4), event categorization (Chapter 7), inferring selectional preferences (Chapter 6) and relation extraction (Chapter 5). In contrast to syntax, semantic processing typically involves a large number of output labels which are highly unbalanced and are often unknown a-priori. In these cases, unsupervised or weakly supervised approaches which can leverage large volumes of unlabeled data often perform better. We demonstrate the benefits of unsupervised learning over supervised learning for semantic processing tasks with experimental results throughout this thesis.

1.1.2 Opportunities

Simple Discourse Structure

While shallow syntactic annotation tasks are more challenging on informal texts, such as those found on Twitter, the short and self-contained nature of these messages means they contain very simple discourse structure as compared with longer texts containing narratives such as news articles. To understand why this, is imagine a user writing a message on Twitter; because it will be mixed up in the feeds of all their followers, they typically don't assume any context the message is meant to be understood in. In contrast a sentence written as part of a news article is meant to be understood within the discourse context

of the article, so there is a tendency for important information about an event to get spread across sentences in the article. Discourse-related issues account for some of the more difficult and unsolved problems in NLP, which include active areas of research such as coreference resolution[131] and classifying relations between events and time expressions[27]. We leverage Twitter's simple discourse structure in order to extract a calendar of popular events occurring in the near future in Chapter 7.

Redundancy

Another unique characteristic of corpora such as Twitter and the Web is their scale. These corpora contain a large amount of redundant information, contributing to information overload, and motivating the need for automatic text processing techniques to aggregate and organize information. On the other hand, redundant information can be exploited to improve the performance of extraction systems [48].

Redundancy of information is leveraged throughout this thesis: Chapter 4 exploits redundancy for categorizing named entities, large amounts of data and redundancy are used to infer the argument types of a relation in Chapter 6, and Chapter 7, uses redundancy to identify popular events mentioned on Twitter.

1.2 Proposed Approach

To address the challenges of large, open-domain, noisy text corpora, and exploit redundancy of information we present a number of unsupervised and weakly supervised methods throughout this thesis which are based on probabilistic latent variable models.

Supervised Learning for Syntax

Syntactic annotations, for example parts of speech and named entities, are a typical first step in most NLP pipelines. While the main focus of this thesis is semantic processing of large heterogeneous text corpora, we need syntactic annotations to get started. Supervised learning is appropriate for syntactic annotation tasks where the output labels are fixed and known in advance. While there is much interesting work on unsupervised learning for

syntax, supervised systems tend to perform better. Available off-the shelf tools for syntax are designed for use on grammatical texts, making them ineffective on the noisy and informal style of text which is prevalent on Twitter, and also common on the Web. In order to enable progress on semantic processing on top of Twitter, we take a practical approach to building NLP tools which is based on in-domain annotated corpora, in addition to semi-supervised techniques which are described in detail in Chapter 3.

Unsupervised and Distantly Supervised Learning for Open-Domain Semantics

Unsupervised learning is appropriate in the situation where the categories are unknown in advance. For example, in Chapter 6 we apply unsupervised models to automatically discover the argument types for relations. In Chapter 7, we make use of unsupervised models to automatically infer an appropriate set of types to describe events extracted from Twitter, and also to categorize events in context. We demonstrate that by leveraging large quantities of unlabeled data we are able to outperform a supervised baseline at this task.

Unsupervised learning is attractive because it allows us to take advantage of large quantities of unlabeled text; however an important question is what to do in the situation where we also have access to structured data related to the task at hand. For some tasks we have readily available access to large structured databases, which can be leveraged as a source of weak (distant) supervision. Chapter 4 presents a new approach to distant supervision which is based on constrained latent variable models and is appropriate for the situation of highly ambiguous training data. We evaluate this approach in the context of weakly supervised named entity categorization in Twitter. A natural question which arises in the context of distant supervision is the issue of missing information either in the text or the database; this issue is addressed in Chapter 5. By explicitly modeling the possibility of missing data we show large improvements in the performance of both distantly supervised named entity categorization and also relation extraction.

Probabilistic Latent Variable Models

Previous work on weakly supervised learning has taken a mostly heuristic approach based on pattern learning [2; 15]. In contrast we propose weakly supervised learning methods based on probabilistic latent variable models. This approach has a number of advantages; for example it provides us with guidelines for what algorithms are likely to work based on theoretical principles rather than relying on intuition for algorithm design; the models are flexible and extensible making it possible to incorporate various sources of background knowledge. For example in Chapter 5 we extend a probabilistic model of distantly supervised relation extraction to model missing data [107], which we show substantially improves performance. In addition, generative probabilistic models have the advantage that they provide a principled way to answer many different kinds of probabilistic queries about the data, and therefore can be useful in a number of different applications; for instance our model of selectional preferences is shown to be useful for filtering improper applications of inference rules in Chapter 6.

1.3 Overview

The rest of this thesis is structured as follows. We begin in Chapter 2 with background information on Bayesian inference techniques for mixed membership models which are used throughout Chapters 4, 6, and 7.

To demonstrate the feasibility of knowledge extraction from noisy user-generated text, we build a pipeline for syntactic annotation of tweets in Chapter 3, taking a practical approach based on supervised learning. These tools are then used as a basis for semantic processing tasks: in Chapter 4 we present a new approach to distant supervision based on constrained latent variable models which is appropriate for the situation of highly ambiguous training data and is evaluated in the context of weakly supervised named entity categorization; in Chapter 7 we use our NLP tools to extract a calendar of popular events occurring in the near future from Twitter.

One issue that arises in distant (weak) supervision is the problem of missing data. This is a general issue affecting both weakly supervised NER (Chapter 4) and weakly supervised

relation extraction; we investigate this issue in Chapter 5, and develop an approach which shows large performance gains by explicitly modeling the possibility of missing information.

Distantly supervised learning is an attractive framework; however in some circumstances we don't have access to relevant structured datasets, or don't even know what the correct set of categories should be in advance. To address these challenges we propose an unsupervised approach to problems in lexical semantics based on Latent Dirichlet Allocation which is applied to the problem of modeling selectional preferences in Chapter 6, and also to the task of categorizing millions of events which are automatically extracted from Twitter in Chapter 7.

Chapter 2

BACKGROUND: BAYESIAN MODELING FOR LEXICAL SEMANTICS

This chapter presents background material on the bayesian models which are used in Chapters 4, 6 and 7. Many excellent and related tutorials are also available online [78; 147; 84]. This chapter focuses on presenting the intuition behind how these models can be applied to problems in lexical semantics in practice rather than going through proofs and derivations in detail.

2.1 Motivation: Latent Variable Models of Word Meaning

A word or phrase can have many different meanings; previous efforts to manually build lexical resources, for example WordNet, have had some success, however, they don't provide a good way to disambiguate word meaning in context. Supervised learning on small manually annotated corpora won't scale up to the challenge of modeling lexical semantics in large open-domain text collections such as Twitter or the web. To build semantic models at scale, we need a way to learn the meanings of individual words and phrases from large quantities of unlabeled text. Generative probabilistic models present an attractive solution. These models have the advantage that they provide a principled way to perform many different kinds of probabilistic queries about the data and are therefore applicable to a wide variety of tasks. Examples of the applications of these bayesian models to problems in lexical semantics are outlined in the following paragraphs.

As an example of how latent variables can be used to model problems in lexical semantics, Chapter 6 demonstrates that a variant of latent Dirichlet allocation [13] can effectively be used to automatically infer the argument types or *selectional preferences* [145] of textual relations [154]. To demonstrate its flexibility and utility, our model of selectional preferences is applied to the task of filtering improper applications of inference rules in context, showing a substantial improvement over a state-of-the-art rule-filtering system which makes use of

a predefined set of classes. The argument/relation clusters automatically discovered by our model are available online.[1]

Chapter 4 proposes a new approach to weakly supervised named entity categorization based on constrained topic models [152]. As a distant source of supervision we make use of lexical entries from Freebase, a large, open-domain database, to generate constraints in the model. This approach leverages the ambiguous supervision provided by Freebase in a principled way, significantly outperforming both a supervised baseline and a state of the art semi-supervised approach to named entity categorization [34] on a Twitter named entity recognition task.

2.2 Generative Models

Generative probabilistic models are an attractive framework for unsupervised and weakly supervised learning. The approach is to define the model as an intuitive story about the generation of our observed data which may involve hidden variables and probabilities. The generative story presents an idealized model for the process by which observed data was generated, and often involves hidden variables that represent latent information we wish to infer. For example, hidden (latent) variables could correspond to named entity categories or argument types (selectional preferences). Given a generative model and observed dataset, Bayesian inference techniques such as Gibbs sampling, or variational methods can then be applied to invert the generative story and infer values for the hidden variables which tell us the information of interest. In the subsequent sections we will walk through a series of increasingly complex generative models.

2.2.1 Mixture of Multinomials

A mixture model is a common way to model grouped data. In a supervised learning scenario, this model is commonly referred to as *Naïve Bayes*. As an example, consider the situation where each group corresponds to a bag of words (for instance the words contained in a web document). Documents can be considered as belonging to different categories (topics), for

[1] `http://www.cs.washington.edu/research/ldasp/ldasp.tgz`

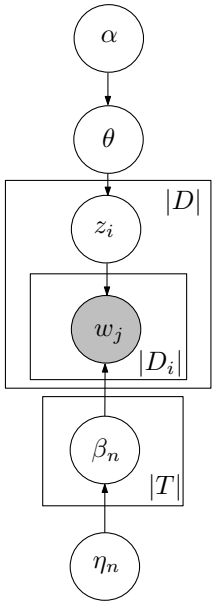

Figure 2.1: Mixture of Multinomials

example Sports, Finance, Politics, etc... We can imagine a generative process by which documents were created in which a topic is first chosen at random, and then the words in the document are randomly picked conditioned on the topic. We don't know which topic, $z_d = t$, belongs to each document, d, in advance, nor do we know the probability of each word given a topic, $\beta_t^w = P(w|t)$, or the probability of each topic $\theta_t = P(t)$. These correspond to hidden (latent) variables which we would like to infer from the observed data.

As a first step towards estimating these hidden variables from our observed documents, a more formal generative story is presented as algorithm 1. A graphical model representation is presented in Figure 2.1. The plates or boxes in Figure 2.1 represent replication, for instance there are $|D|$ documents, and document i contains $|D_i|$ words.

Here the Dirichlet distribution, $\text{Dir}(\alpha)$, is a conjugate prior to the multinomial, which simplifies inference. For a detailed explanation of the Dirichlet see Heinrich [78].

While this generative story is an extreme simplification of the real process by which documents are created, it is useful because we know how to invert this generative story and infer values for the hidden variables which will then give us an estimate of the category

Algorithm 1 Generative Story for Mixture of Multinomials

Generate θ according to symmetric Dirichlet distribution Dir(α).
for each topic $t = 1 \dots |T|$ **do**
    Generate β_t according to symmetric Dirichlet distribution Dir(η).
end for
for each document $d = 1 \dots |D|$ **do**
    Generate document d's topic, z_d, from Mult(θ)
    for each word position $i = 1 \dots n$ **do**
        Generate word $w_{d,i}$ from Mult(β_{z_d})
    end for
end for

of each document based on all the information we have available. Gibbs sampling is one approach to inference in which a markov chain is constructed which transitions between states corresponding to assignments to the hidden variables, and which converges to the posterior distribution conditioned on observed evidence. We go through the details for a Gibbs sampler for another model, Latent Dirichlet Allocation, in Section 2.2.3, for details of Gibbs sampling in the multinomial mixture model, see [147].

2.2.2 Latent Dirichlet Allocation

The mixture model presented in Section 2.2.1 explicitly assumes that each bag of words is generated from a unique latent topic. This assumption is reasonable in some scenarios, however problems in lexical semantics often involve significant amounts of ambiguity. For instance a given verb can correspond to different meanings or word senses, and the same named entity can refer to multiple real-world entities which belong in different categories. Requiring a single category for each lexical entry (which can appear across many contexts) can be a problematic restriction. This restriction is relaxed in Latent Dirichlet Allocation (LDA) [13] which is a bayesian version of PLSA [81]. In LDA, each bag of elements (e.g. words in a document or contexts of a lexical item) is modeled as a mixture of topics as opposed to having a single underlying topic. These *mixed membership*, or *admixture* models

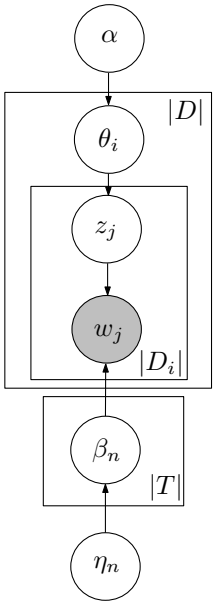

Figure 2.2: Latent Dirichlet Allocation

are more appropriate for many problems in lexical semantics where individual lexical items are highly ambiguous, associating with different underlying categories in different contexts. One example is the problem of named entity categorization which is investigated in detail in Chapter 4, for instance the entity *JFK* could refer to either a *PERSON* or a *FACILITY* (airport). Another instance of a problem in lexical semantics where mixed membership models are advantageous is the problem of modeling argument types or selectional preferences which we present in Chapter 6.

We present the generative story for LDA in the context of unsupervised named entity categorization below as Algorithm 2. A graphical model representation is presented in Figure 2.2.

2.2.3 Collapsed Gibbs Sampling for Latent Dirichlet Allocation

A popular and effective approach to inferring the latent variables, $z_{e,i}$ in LDA is collapsed Gibbs sampling [73]. Gibbs sampling is a Markov chain Monte Carlo method. This approach samples from the posterior distribution over the hidden variables by constructing a Markov

Algorithm 2 Generative Story for Latent Dirichlet Allocation

for each type: $t = 1 \dots T$ **do**

Generate β_t according to symmetric Dirichlet distribution $\text{Dir}(\eta)$.

end for

for each entity string $e = 1 \dots |E|$ **do**

Generate a distribution over types, θ_e, according to Dirichlet distribution $\text{Dir}(\alpha)$.

for each word position $i = 1 \dots N_e$ **do**

Generate a type $z_{e,i}$ from $\text{Mult}(\theta_e)$.

Generate the word $w_{e,i}$ from $\text{Mult}(\beta_{z_{e,i}})$.

end for

end for

chain whose stationary distribution is proven to correspond to the posterior. In Gibbs sampling, we transition through the state space by iterating through each hidden variable $z_{e,i}$ individually and resampling a new value, $z_{e,i} = t$, conditioned on the current assignment of all other variables $z_{-(e,i)}$. In *collapsed* Gibbs sampling, parameters β and θ are integrated out during this process. We start with an initial, random assignment to the hidden variables, then run the markov chain for a large number of *burn-in* iterations until the chain has likely converged to the posterior distribution, at which point we can begin taking samples.

Pseudocode for collapsed Gibbs sampling in LDA is presented in Algorithm 3. The distribution over topics which is sampled from in line 15 can be computed as follows:

$$P(z|\mathbf{z}_{-(b,i)}, \mathbf{w}) \propto \frac{C(d,z) + \alpha_z}{\sum_{t=1}^{T} (C(d,t) + \alpha_t)} \cdot \frac{C(z, w_{b,i}) + \eta^z_{w_{b,i}}}{\sum_{v=1}^{V} (C(z,v) + \eta_v)}$$

Where $C(a,b)$ are counts, for example $c(d,z)$ is the number of times type z appears in document d (see Algorithms 3 and 1). This formula makes sense intuitively: the probability of a word, w being assigned to type t is related to the proportion of other words assigned to t in the same document, and also the proportion of mentions of word w which are assigned to t across the entire corpus. For a detailed derivation of the collapsed Gibbs sampler for latent Dirichlet allocation see Heinrich [78].

Algorithm 3 Collapsed Gibbs sampling for Latent Dirichlet Allocation

Initialize topic-word counts $C(z, w) = 0 \; \forall z, w$
Initialize document-topic counts $C(d, z) = 0 \; \forall d, z$
for each bag of words: $b = 1 \ldots |B|$ **do** ▷ Random Initialization
    for each word position: $i = 1 \ldots N_b$ **do**
        Randomly initialize $z_{b,i}$
        Increment document-topic count $C(d, z_{b,i})$
        Increment topic-word count $C(z_{b,i}, w_{b,i})$
    end for
end for
for number of burn in iterations **do**
    for each bag of words: $b = 1 \ldots |B|$ **do**
        for each word position: $i = 1 \ldots N_b$ **do**
            Decrement document-topic count $C(d, z_{b,i})$ ▷ Subtract counts for $w_{d,i}$
            Decrement topic-word count $C(z_{b,i}, w_{b,i})$
            Sample $z_{b,i} \sim P(z|\mathbf{z}_{-(b,i)}, \mathbf{w})$ ▷ Sample a new topic
            Increment document-topic count $C(d, z_{b,i})$ ▷ Add back new counts
            Increment topic-word count $C(z_{b,i}, w_{b,i})$
        end for
    end for
end for

Chapter 3

SUPERVISED LEARNING FOR SYNTACTIC ANNOTATION IN MICROBLOG TEXT

Syntactic annotation is a first step in most NLP and IE pipelines. Off-the-shelf NLP tools which are mostly designed for processing news articles perform very poorly when applied to Twitter due to its noisy and unique style. To address this challenge and enable applications of information extraction on top of Twitter, we present an NLP pipeline which has been tuned to work well on noisy Twitter data. The tools developed here are used as input to the weakly supervised named entity categoirzation system for tweets presented in Chapter 4, and also to extract a calendar of popular events as described in Chapter 7.

Status Messages posted on Social Media websites such as Facebook and Twitter present a new and challenging style of text for language technology due to their noisy and informal nature. Like SMS [85], tweets are particularly terse and difficult (See Table 3.1). Yet tweets provide a unique compilation of information that is more up-to-date and inclusive than news articles, due to the low-barrier to tweeting, and the proliferation of mobile devices.[1] The corpus of tweets already exceeds the size of the Library of Congress [76] and is growing far more rapidly. Due to the volume of tweets, it is natural to consider named-entity recognition, information extraction, and text mining over tweets. Not surprisingly, the performance of "off the shelf" NLP tools, which were trained on news corpora, is weak on tweet corpora.

To adress this challenge, we present a re-trained "NLP pipeline" that leverages previously-annotated out-of-domain text, [2] annotated tweets, and unlabeled tweets to enable higher quality part-of-speech tagging, chunking, and named-entity recognition.

We experimentally evaluate the performance of off-the-shelf news trained NLP tools when applied to Twitter. For example POS tagging accuracy drops from about 0.97 on

[1] See the "trending topics" displayed on `twitter.com`

[2] Although tweets can be written on any subject, following convention we use the term "domain" to include text styles or genres such as Twitter, News or IRC Chat.

1	The Hobbit has FINALLY started filming! I cannot wait!
2	Yess! Yess! Its official Nintendo announced today that they Will release the Nintendo 3DS in north America march 27 for $250
3	Government confirms blast n nuclear plants n japan...don't knw wht s gona happen nw...

Table 3.1: Examples of noisy text in tweets.

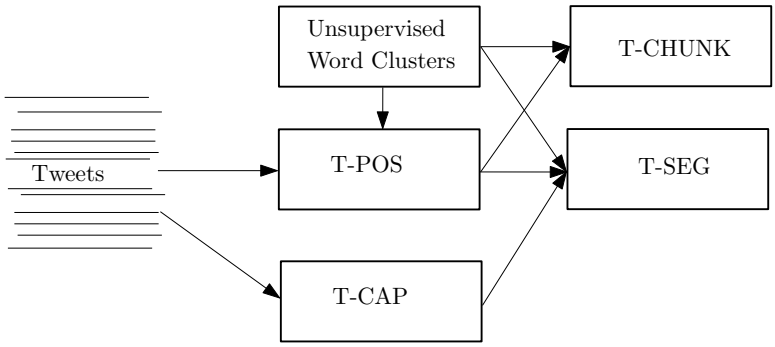

Figure 3.1: Processing pipeline for syntax in Twitter.

news to 0.80 on tweets. By utilizing in-domain, out-of-domain, and unlabeled data we are able to substantially boost performance, for example obtaining a 52% increase in F_1 score on segmenting named entities.

The rest of the chapter is organized as follows. We successively build the NLP pipeline for Twitter feeds in Sections 3.1 and 3.2. We first present our approaches to shallow syntax – part of speech tagging (Section 3.1.1), and shallow parsing (Section 3.1.2). Section 3.1.3 describes a novel classifier that predicts the informativeness of capitalization in a tweet. All tools in Section 3.1 are used as features for named entity segmentation in Section 3.2.1. We describe related work in Section 3.4 and conclude in Section 3.5.

3.1 Shallow Syntactic Annotation in Tweets

We first study two fundamental shallow syntactic annotation tasks – POS tagging and shallow parsing (also referred to as chunking). To address Twitter's lack of reliable capitalization, we also discuss a novel capitalization classifier in Section 3.1.3, which predicts whether capitalization within a specific tweet is informative. The outputs of these tools are used in feature generation for named entity recognition in the next section.

For all experiments described in this section we use a dataset of 800 randomly sampled tweets. All results (Tables 3.2, 3.7 and 3.8) are evaluated using 4-fold cross-validation experiments on the respective tasks.[3]

3.1.1 Part of Speech Tagging

Part of speech tagging provides an initial level of syntactic analysis, which is useful for feature generation across a wide range of NLP tasks, for example: named entity segmentation, semantic role labeling and information extraction.

Prior experiments have suggested that POS tagging has a very strong baseline: assign each word to its most frequent tag and assign each out of vocabulary (unseen) word the most common POS tag. This baseline obtained 0.9 accuracy on the Brown corpus [29]. However, the application of a similar baseline on tweets (see Table 3.2) obtains a much weaker 0.76, which suggests the challenging nature of Twitter data.

One explanation for this drop in accuracy is that Twitter contains far more out of vocabulary words than grammatical text. Many of these out of vocabulary words stem from spelling variation, *e.g.*, the use of the word "n" for "in" in Table 3.1 example 3. Although NNP is the most frequent tag for out of vocabulary words, only about 1/3 are NNPs.

The performance of off-the-shelf news-trained POS taggers is also quite poor on Twitter data. The state-of-the-art Stanford POS tagger [177] improves on the simple baseline, obtaining an accuracy of 0.8. This performance is impressive given that its training data, the Penn Treebank WSJ (PTB), is so different in style from Twitter, however it is a huge

[3]We used Brendan O'Connor's Twitter tokenizer `https://github.com/brendano/tweetmotif`.

	Accuracy	Error Reduction
Majority Baseline (NN)	0.189	-
Word's Most Frequent Tag	0.760	-
Stanford POS Tagger	0.801	-
T-POS(PTB)	0.813	6%
T-POS(Twitter)	0.853	26%
T-POS(IRC + PTB)	0.869	34%
T-POS(IRC + Twitter)	0.870	35%
T-POS(PTB + Twitter)	0.873	36%
T-POS(PTB + IRC + Twitter)	0.883	41%

Table 3.2: POS tagging performance on tweets. By training on in-domain labeled data, in addition to annotated IRC chat data, we obtain a 41% reduction in error over the Stanford POS tagger.

Gold	Predicted	Stanford Error	T-POS Error	Error Reduction
NN	NNP	0.102	0.072	29%
UH	NN	0.387	0.047	88%
VB	NN	0.071	0.032	55%
NNP	NN	0.130	0.125	4%
UH	NNP	0.200	0.036	82%

Table 3.3: Most common errors made by the Stanford POS Tagger on tweets. For each case we list the fraction of times the gold tag is misclassified as the predicted for both our system and the Stanford POS tagger. All verbs are collapsed into VB for compactness.

drop from the 97% accuracy reported on the PTB. There are several reasons for this drop in performance. Table 3.3 lists common errors made by the Stanford tagger. First, due to unreliable capitalization, common nouns are often misclassified as proper nouns, and vice versa. Also, interjections and verbs are frequently misclassified as nouns. In addition to differences in vocabulary, the grammar of tweets is quite different from edited news text. For instance, tweets often start with a verb (where the subject 'I' is implied), as in: "watchng american dad."

To overcome these differences in style and vocabulary, we manually annotated a set of 800 tweets (16K tokens) with tags from the Penn TreeBank tag set for use as in-domain training data for our POS tagging system, T-POS.[4] We add new tags for the Twitter specific phenomena: retweets, @usernames, #hashtags, and urls. Note that words in these categories can be tagged with 100% accuracy using simple regular expressions. To ensure fair comparison in Table 3.2, we include a postprocessing step which tags these words appropriately for all systems.

To help address the issue of out of vocabulary words and lexical variations, we perform clustering to group together words which are distributionally similar [18; 179]. In particular, we perform hierarchical clustering using Jcluster [71] on 52 million tweets; each word is uniquely represented by a bit string based on the path from the root of the resulting hierarchy to the word's leaf. We use the Brown clusters resulting from prefixes of 4, 8, and 12 bits. These clusters are often effective in capturing lexical variations, for example, following are lexical variations on the word "tomorrow" from one cluster after filtering out other words (most of which refer to days):

'2m', '2ma', '2mar', '2mara', '2maro', '2marrow', '2mor', '2mora', '2moro', '2morow', '2morr', '2morro', '2morrow', '2moz', '2mr', '2mro', '2mrrw', '2mrw', '2mw', 'tmmrw', 'tmo', 'tmoro', 'tmorrow', 'tmoz', 'tmr', 'tmro', 'tmrow', 'tmrrow', 'tmrrw', 'tmrw', 'tmrww', 'tmw', 'tomaro', 'tomarow', 'tomarro', 'tomarrow', 'tomm', 'tommarow', 'tommarrow', 'tommoro', 'tommorow', 'tommorrow', 'tommorw', 'tommrow', 'tomo', 'tomolo', 'tomoro', 'tomorow', 'tomorro', 'tomorrw', 'tomoz', 'tomrw', 'tomz'

[4] Using MMAX2 [123] for annotation.

Feature	Description
POS-Dict	Part of speech tag dictionary constructed from the Penn Treebank corpus [176].
Brown-Clusters	4, 8 and 12 bit brown clusters constructed from a corpus of 52 million tweets [18; 179].
Orthographic-Features	Features based on capitalization, word prefixes and suffixes, presence of punctuation, digits
Context	Features from previous and following words

Table 3.4: List of features used for POS tagging.

T-POS uses Conditional Random Fields[5] [94], both because of their ability to model strong dependencies between adjacent POS tags, and also to make use of highly correlated features (for example a word's identity in addition to prefixes and suffixes). Besides employing the Brown clusters computed above, we use a fairly standard set of features that include POS dictionaries, spelling and contextual features. A detailed list of features is presented in Table 3.4. [6]

We evaluated the performance of T-POS using 4-fold cross validation over 800 tweets, which leads to a 26% reduction in error over the Stanford tagger.

In addition we experimented with including additional training data, starting with 40K tokens of annotated IRC chat data [65], which is similar in style. Like Twitter, IRC data contains many misspelled/abbreviated words, and also more pronouns, and interjections, but fewer determiners than news. We also leverage 50K POS-labeled tokens from the Penn Treebank [113], which was found to be beneficial during devlopment. Overall T-POS trained on 102K tokens (12K from Twitter, 40K from IRC and 50K from PTB) results in a 41% error reduction over the Stanford tagger, obtaining an accuracy of 0.883. Table 3.3 lists

[5]We use MALLET [115].

[6]Complete feature extraction code is also available online: `https://github.com/aritter/twitter_nlp/blob/master/python/pos_tag/features.py`.

gains on some of the most common error types, for example, T-POS dramatically reduces error on interjections and verbs that are incorrectly classified as nouns by the Stanford tagger.

Although performance should increase with more labeled in-domain data, we believe that it will likely always be less than performance on Newswire text, due to inherent noise, and wide variety in style found on Twitter. For instance, Table 3.3 indicates there is still much confusion between common and proper nouns, indicating that Twitter's unreliable capitalization is problematic even with in-domain labeled data.

	Twitter	PTB	IRC
DT	6%	9%	5%
PRP	7%	2%	10%
UH	3%	0.0002%	12%

Table 3.5: Percent usage of selected tags in Twitter, IRC and Penn Treebank (PTB) text. Both Twitter and IRC Chat data contain fewer determiners (DT), and more pronouns(PRP) and interjections(UH) than PTB text.

	# Tweets	# Tokens
POS	800	15,185
Chunk	800	15,185
Informative Capitalization	400	6,692
NE Segmentation	2,400	41,765
NE Classification	400	7,624

Table 3.6: Summary of annotated Twitter datasets

3.1.2 Shallow Parsing (Chunking)

Shallow parsing, or chunking is the task of identifying non-recursive phrases, such as noun phrases, verb phrases, and prepositional phrases in text. Similar to the benefits of part

	Accuracy	Error Reduction
Majority Baseline (B-NP)	0.266	-
OpenNLP	0.839	-
T-CHUNK(CoNLL)	0.854	9%
T-CHUNK(Twitter)	0.867	17%
T-CHUNK(CoNLL + Twitter)	0.875	22%

Table 3.7: Token-Level accuracy at shallow parsing tweets. We compare against the OpenNLP chunker as a baseline.

of speech taging, features generated from accurate shallow parsing of tweets could benefit many applications such as Information Extraction and Named Entity Recognition.

Off the shelf shallow parsers perform noticeably worse on tweets, motivating us again to annotate in-domain training data. We annotate the same set of 800 tweets mentioned previously with tags from the CoNLL shared task [175]. We use the set of shallow parsing features described by Sha and Pereira [165], in addition to the Brown clusters mentioned above. Part-of-speech tag features are extracted based on cross-validation output predicted by T-POS. For inference and learning, again we use Conditional Random Fields. We utilize 16K tokens of in-domain training data (using cross validation), in addition to 210K tokens of newswire text from the CoNLL dataset.

Table 3.7 reports T-CHUNK's performance at shallow parsing of tweets. We compare against the off-the shelf OpenNLP chunker[7], obtaining a 22% reduction in error.

3.1.3 Identifying Informative Capitalization

A key orthographic feature for recognizing named entities is capitalization [64; 47]. Unfortunately in tweets, capitalization is much less reliable than in edited texts. In addition, there is a wide variety in the styles of capitalization. In some tweets capitalization is informative, whereas in other cases, non-entity words are capitalized simply for emphasis. Some tweets

[7] `http://incubator.apache.org/opennlp/`

	P	R	F_1
Majority Baseline	0.70	1.00	0.82
T-CAP	0.77	0.98	0.86

Table 3.8: Performance at predicting reliable capitalization.

contain all lowercase words (8%), whereas others are in ALL CAPS (0.6%).

To address this issue, it is helpful to incorporate information based on the entire content of the message to determine whether or not its capitalization is informative. To this end, we build a capitalization classifier, T-CAP, which predicts whether or not a tweet is informatively capitalized. Its output is used as a feature for Named Entity Recognition. We manually labeled our 800 tweet corpus as having either "informative" or "uninformative" capitalization. The criteria we use for labeling is as follows: if a tweet contains any non-entity words which are capitalized, but do not begin a sentence, or it contains any entities which are not capitalized, then its capitalization is "uninformative", otherwise it is "informative".

For learning , we use Support Vector Machines.[8] The features used include: the fraction of words in the tweet which are capitalized, the fraction which appear in a dictionary of frequently lowercase/capitalized words but are not lowercase/capitalized in the tweet, the number of times the word 'I' appears lowercase and whether or not the first word in the tweet is capitalized. Results comparing against the majority baseline, which predicts capitalization is always informative, are shown in Table 3.8. Additionally, in Section 3.2 we show that features based on our capitalization classifier improve performance at named entity segmentation.

3.2 Segmenting Named Entities

We now discuss our approach to named entity recognition on Twitter data. As with POS tagging and shallow parsing, off the shelf named-entity recognizers perform poorly on tweets. For example, applying the Stanford Named Entity Recognizer to one of the examples from

[8] `http://www.chasen.org/~taku/software/TinySVM/`

Table 3.1 results in the following output:

> [Yess]$_{ORG}$! [Yess]$_{ORG}$! Its official [Nintendo]$_{LOC}$ announced today that they Will release the [Nintendo]$_{ORG}$ 3DS in north [America]$_{LOC}$ march 27 for $250

The OOV word 'Yess' is mistaken as a named entity. In addition, although the first occurrence of 'Nintendo' is correctly segmented, it is misclassified, whereas the second occurrence is improperly segmented – it should be the product "Nintendo 3DS". Finally "north America" should be segmented as a *LOCATION*, rather than just 'America'. In general, news-trained Named Entity Recognizers seem to rely heavily on capitalization, which we know to be unreliable in tweets.

Following Collins and Singer [34], Downey et al. [47] and Elsner et al. [56], we treat classification and segmentation of named entities as separate tasks. This allows us to more easily apply techniques better suited towards each task. For example, we are able to use discriminative methods for named entity segmentation and distantly supervised approaches for classification. While it might be beneficial to jointly model segmentation and (distantly supervised) classification using a joint sequence labeling and topic model similar to that proposed by Sauper et al. [160], we leave this for potential future work.

Because most words found in tweets are not part of an entity, we need a larger annotated dataset to effectively learn a model of named entities. We therefore use a randomly sampled set of 2,400 tweets for NER. All experiments (Tables 3.9, 4.2-4.4) report results using 4-fold cross validation.

3.2.1 Segmenting Named Entities

Because capitalization in Twitter is less informative than news, in-domain data is needed to train models which rely less heavily on capitalization, and also are able to utilize features provided by T-CAP.

We exhaustively annotated our set of 2,400 tweets (34K tokens) with named entities.[9] A convention on Twitter is to refer to other users using the @ symbol followed by their unique

[9]We found that including out-of-domain training data from the MUC competitions lowered performance at this task.

	P	R	F_1	F_1 inc.
Stanford NER	0.62	0.35	0.44	-
T-SEG(None)	0.71	0.57	0.63	43%
T-SEG(T-POS)	0.70	0.60	0.65	48%
T-SEG(T-POS, T-CHUNK)	0.71	0.61	0.66	50%
T-SEG(All Features)	0.73	0.61	0.67	52%

Table 3.9: Performance at segmenting entities varying the features used. "None" removes POS, Chunk, and capitalization features. Overall we obtain a 52% improvement in F_1 score over the Stanford Named Entity Recognizer.

username. We deliberately choose not to annotate @usernames as entities in our data set because they are both unambiguous, and trivial to identify with 100% accuracy using a simple regular expression, and would only serve to inflate our performance statistics. While there is ambiguity as to the type of @usernames (for example, they can refer to people or companies), we believe they could be more easily classified using features of their associated user's profile than contextual features of the text.

T-SEG models Named Entity Segmentation as a sequence-labeling task using IOB encoding for representing segmentations (each word either begins (B), is inside (I), or is outside (O) of a named entity). For example in the following tagged Tweet:

The/B Town/I might/O be/O one/O of/O the/O best/O movies/O ...

The Town can be read off as a named entity. The word *The* begins (B) the entity, *Town* is inside (I) and *might* is outside (O). We use Conditional Random Fields for learning and inference. Again we include orthographic, contextual and dictionary features; our dictionaries included a set of type lists gathered from Freebase. In addition, we use the Brown clusters and outputs of T-POS, T-CHUNK and T-CAP in generating features.

We report results at segmenting named entities in Table 3.9. Compared with the state-of-the-art news-trained Stanford Named Entity Recognizer [62], T-SEG obtains a 52% increase in F_1 score.

3.3 Limitations and Future Work

The performance we have described on syntactic annotation of Twitter is lower in each case than reported performance on grammatical texts such as news articles. Predicting syntactic annotations of Twitter's noisy and informal text is a more challenging task, due to its noisy nature, and diversity of different styles. In contrast news is grammatical and follows strict stylistic conventions.

Even though this task is more difficult in Twitter, we are still able to achieve good enough performance to be useful for information extraction applications. Furthermore, once we get past these noisy text challenges other tasks become easier due to Twitter's simple discourse structure. We leverage the tools described in this chapter to build a weakly supervised named entity categorization system in Chapter 4, and also to extract a calendar of popular events occuring in the near future in Chapter 7.

While annotating syntax in Twitter is a very difficult task, we believe that there is room for improvement. One possible avenue for improving performance at shallow syntactic annotation tasks is to apply recent work on text normalization [77]. We suspect that using text normalization systems as a reprocessing step on the input could lead to cascading errors, however using their outputs as additional features could be a very effective way to handle Twitter's diversity of style.

3.4 Related Work

There has been relatively little previous work on building NLP tools for Twitter or similar text styles. Locke and Martin [109] train a classifier to recognize named entities based on annotated Twitter data, handling the types *PERSON*, *LOCATION*, and *ORGANIZATION*. Developed in parallel to our work, Liu et al. [108] investigate NER on the same 3 types, in addition to *PRODUCT*s and present a semi-supervised approach using k-nearest neighbor. Also developed in parallel, Gimpell et al. [69] build a POS tagger for tweets using 20 coarse-grained tags. Benson et. al. [8] present a system which extracts artists and venues associated with musical performances. Recent work [77; 72] has proposed lexical normalization of tweets which may be useful as a preprocessing step

for the upstream tasks like POS tagging and NER. In addition Finin et. al. [61] investigate the use of Amazon's Mechanical Turk for annotating Named Entities in Twitter, Minkov et. al. [120] investigate person name recognizers in email, and Singh et. al. [166] apply a minimally supervised approach to extracting entities from text advertisements. In contrast to previous work, we have demonstrated the utility of features based on Twitter-specific POS taggers and Shallow Parsers in segmenting Named Entities.

3.5 Conclusions

We have demonstrated that existing tools for POS tagging, Chunking and Named Entity Recognition perform quite poorly when applied to noisy Twitter text. To address this challenge we have annotated tweets and built tools trained on unlabeled, in-domain and out-of-domain data, showing substantial improvement over their state-of-the art news-trained counterparts, for example, T-POS outperforms the Stanford POS Tagger, reducing error by 41%. We showed the benefit of features designed to work on noisy Twitter text including unsupervised word clusters and a novel capitalization reliability classifier. Additionally we have shown the benefits of features generated from T-POS and T-CHUNK in segmenting Named Entities.

Access to NLP tools which are capable of processing the short informal texts found on Twitter enables semantic processing and knowledge extraction from this realtime, inclusive stream of text. For example in Chapter 7, we leverage these tools to extract a realtime calendar of popular events occurring on Twitter. We also believe our NLP tools have the potential to benefit a wide variety of additional applications for example: disaster relief [126], sociological research, for instance studying the phenomenon of bullying [186], and sentiment analysis [184].

Our POS tagger, Chunker Named Entity Recognizer are available for use by the research community: `http://github.com/aritter/twitter_nlp`

Chapter 4

DISTANTLY SUPERVISED NAMED ENTITY CATEGORIZATION WITH CONSTRAINED LATENT VARIABLE MODELS

In Chapter 3, we presented a set of NLP tools adapted to Twitter, supporting a named entity segmenter which identifies the spans of entity mentions in this noisy, informal style of text. Part of the traditional named entity recognition task includes categorizing the entities into types; the traditional 3 types for named entity recognition in newswire text being *PERSON*, *LOCATION* and *ORGANIZATION*. Supervised learning is problematic for named entity categorization in Twitter, however, due to the diverse types of entities which are mentioned in addition to its terse style. In this chapter we therefore explore weakly supervised approaches to named entity categorization. We extend previous work on bootstrapping semantic categories [59; 23; 90; 174; 116], by applying constrained latent variable models. Probabilistic latent variable models provide a principled approach to estimating the type of new entities using existing dictionaries of entities as constraints in a probabilistic latent variable model. Because our approach is based on generative models, it provides a principled way to query the type of each entity mention in context; by leveraging large amounts of unlabeled data we are able to show large performance improvements over a supervised baseline.

4.1 Classifying Named Entities in Twitter

Classifying named entities in tweets is a difficult task for two reasons. First, tweets contain a plethora of distinctive named entity types (Companies, Products, Bands, Movies, and more). Almost all these types (except for People and Locations) are relatively infrequent, so even a large sample of manually annotated tweets will contain few training examples. Secondly, due to Twitter's 140 character limit, tweets often lack sufficient context to determine an entity's type without the aid of background knowledge.

To address these issues we introduce a novel approach to *distant supervision* [121] based

on Latent Dirichlet Allocation [13], which is able to leverage large amounts of unlabeled data in addition to large dictionaries of entities gathered from Freebase, and combines information about an entity's context across its mentions. Each entity's distribution over types is constrained based on an open-domain database (Freebase) as a source of supervision. This approach increases F_1 score by 25% relative to co-training [14; 194] on the task of classifying named entities in Tweets.

Because Twitter contains many distinctive, and infrequent entity types, gathering sufficient training data for named entity classification is a difficult task. In any random sample of tweets which is possible to annotate, many types will only occur a few times. Moreover, due to their terse nature, individual tweets often lack enough context to determine the type of the entities they contain. For example, consider following tweet:

> KKTNY in 45min.....

without any prior knowledge, there is not enough context to determine what type of entity "KKTNY" refers to, however by exploiting redundancy in the data [49], we can determine it is likely a reference to a television show as it often co-occurs with words such as *watching* and *premieres* in other contexts.[1]

In order to handle the problem of many infrequent types, we leverage large lists of entities and their types gathered from an open-domain ontology (Freebase) as a source of distant supervision, allowing use of large amounts of unlabeled data in learning.

4.2 Freebase

Although Freebase has very broad coverage, simply looking up entities and their types is inadequate for classifying named entities in context; this approach results in a 0.38 F-score, as described in Section 4.4. The Freebase dictionaries are highly ambiguous, so we therefore need some way to disambiguate the type of an entity in context. For example, according to Freebase, the string 'China' could refer to a country, a band, a person, or a film. This problem is very common: 35% of the entities in our data appear in more than one of

[1] Kourtney & Kim Take New York.

Type	Top 20 Entities not found in Freebase dictionaries
PRODUCT	nintendo ds lite, apple ipod, generation black, ipod nano, apple iphone, gb black, xperia, ipods, verizon media, mac app store, kde, hd video, nokia n8, ipads, iphone/ipod, galaxy tab, samsung galaxy, playstation portable, nintendo ds, vpn
TV-SHOW	pretty little, american skins, nof, order svu, greys, kktny, rhobh, parks & recreation, parks & rec, dawson 's creek, big fat gypsy weddings, big fat gypsy wedding, winter wipeout, jersey shores, idiot abroad, royle, jerseyshore, mr . sunshine, hawaii five-0, new jersey shore
FACILITY	voodoo lounge, grand ballroom, crash mansion, sullivan hall, memorial union, rogers arena, rockwood music hall, amway center, el mocambo, madison square, bridgestone arena, cat club, le poisson rouge, bryant park, mandalay bay, broadway bar, ritz carlton, mgm grand, olympia theatre, consol energy center

Table 4.1: Example type lists produced by LabeledLDA. No entities which are shown were found in Freebase; these are typically either too new to have been added, or are misspelled/abbreviated (for example rhobh="Real Housewives of Beverly Hills"). In a few cases there are segmentation errors.

our (mutually exclusive) Freebase dictionaries. Additionally, 30% of entities mentioned on Twitter do not appear in any Freebase dictionary, as they are either too new (for example a newly released videogame), or are misspelled or abbreviated (for example 'mbp' is often used to refer to the "mac book pro").

4.3 Distant Supervision with Topic Models

To model unlabeled entities and their possible types, we apply LabeledLDA [143], constraining each entity's distribution over topics based on its set of possible types according to Freebase. In contrast to previous weakly supervised approaches to Named Entity Classification, for example the Co-Training and Naïve Bayes (EM) models of Collins and Singer [34], LabeledLDA models each entity string as a mixture of types rather than using a single hidden variable to represent the type of each mention. This allows information about an entity's distribution over types to be shared across mentions, naturally handling ambiguous entity strings whose mentions could refer to different types.

Each entity string in our data is associated with a bag of context-words found within a context window around all of its mentions, and also within the entity itself. Each bag of context-words is associated with a distribution over topics, Multinomial(θ_e), and each topic is associated with a distribution over context-words, Multinomial(β_t). In addition, we maintain a one-to-one mapping between topics and Freebase type dictionaries. These dictionaries constrain θ_e, the distribution over topics for each entity string, based on its set of possible types, $FB[e]$. For example, θ_{Amazon} could correspond to a distribution over two types: *COMPANY*, and *LOCATION*, whereas θ_{Apple} might represent a distribution over *COMPANY*, and *FOOD*. For entities which aren't found in any of the Freebase dictionaries, we leave their topic distributions θ_e unconstrained. Note that in absence of any constraints LabeledLDA reduces to standard LDA, and a fully unsupervised setting similar to that presented by Elsner et. al. [56].

In detail, the generative story that models our data for distantly superivsed named entity classification is presented as Algorithm 4.

To infer values for the hidden variables, we apply Collapsed Gibbs sampling [73], where parameters are integrated out, and the $z_{e,i}$s are sampled directly. For details, refer to

Algorithm 4 Generative Story for Distantly Supervised Named Entity Categorization

for each type: $t = 1 \dots T$ **do**

Generate β_t according to symmetric Dirichlet distribution $\text{Dir}(\eta)$.

end for

for each entity string $e = 1 \dots |E|$ **do**

Generate θ_e over $FB[e]$ according to Dirichlet distribution $\text{Dir}(\alpha_{FB[e]})$.

for each word position $i = 1 \dots N_e$ **do**

Generate $z_{e,i}$ from $\text{Mult}(\theta_e)$.

Generate the word $w_{e,i}$ from $\text{Mult}(\beta_{z_{e,i}})$.

end for

end for

Chapter 2.

4.3.1 Classifying Entity Mentions in Context

In making predictions, we found it beneficial to consider θ_e^{train} as a prior distribution over types for entities which were encountered during (distantly supervised) training. In practice this sharing of information across contexts is very beneficial as there is often insufficient evidence in an isolated tweet to determine an entity's type. For entities which weren't encountered during training, we instead use a prior based on the distribution of types across all entities. One approach to classifying entities in context is to assume that θ_e^{train} is fixed, and that all of the words inside the entity mention and context, $\mathbf{w}$, are drawn based on a single topic, z, that is they are all drawn from Multinomial(β_z). We can then compute the posterior distribution over types in closed form with a simple application of Bayes rule:

$$P(z|\mathbf{w}) \quad \propto \quad \prod_{w \in \mathbf{w}} P(w|z:\beta) P(z:\theta_e^{\text{train}})$$

During development, however, we found that rather than making these assumptions, using Gibbs Sampling to estimate the posterior distribution over types performs slightly better. In order to make predictions, for each entity we use an informative Dirichlet prior

based on θ_e^{train} and perform 100 iterations of Gibbs Sampling holding the hidden topic variables in the training data fixed [193]. Fewer iterations are needed than in training since the type-word distributions, β have already been inferred.

In order to make predictions for entity mentions in context, we perform 100 iterations of Gibbs Sampling on the group of words inside the entity mention and a context window, holding the hidden topic variables in the (distantly supervised) training data fixed [193]. We need fewer sampling iterations than in training, since the type-word distributions, β_t have already been inferred.

For entities which have been encountered during training, we generate an informative Dirichlet prior using the estimate of their distribution over types across all contexts (θ_e^{train}). This has the effect of biasing the classification of an entity mention in context towards its more probable types. The local context can override this prior, however in cases where the local context is uninformative (a common situation due to lack of context in tweets), the prior suggests the most likely type which we found to be very useful in practice.

4.4 Experiments

To evaluate T-CLASS's ability to classify entity mentions in context, we annotated the 2,400 tweets with 10 types which are both popular on Twitter, and have good coverage in Freebase: *PERSON*, *GEO-LOCATION*, *COMPANY*, *PRODUCT*, *FACILITY*, *TV-SHOW*, *MOVIE*, *SPORTSTEAM*, *BAND*, and *OTHER*. Note that these type annotations are only used for evaluation purposes, and are not used for training T-CLASS, which relies only on distant supervision. In some cases, we combine multiple Freebase types to create a dictionary of entities representing a single type (for example the *COMPANY* dictionary contains Freebase types */business/consumer_company* and */business/brand*). Because our approach does not rely on any manually labeled examples, it is straightforward to extend it for different sets of types based on the needs of downstream applications.

4.4.1 Training

To gather unlabeled data for inference, we run T-SEG, our entity segmenter (described in Chapter 3), on 60M tweets, and keep the entities which appear 100 or more times.

This results in a set of 23,651 distinct entity strings. For each entity string, we collected words occurring in a context window of 3 words from all mentions in our data, and used a vocabulary of the 100K most frequent words. We ran Gibbs sampling for 1,000 iterations, using the last sample to estimate entity-type distributions θ_e, in addition to type-word distributions β_t. Table 4.1 displays the 20 entities (not found in Freebase) whose posterior distribution θ_e assigns highest probability to selected types.

4.4.2 Testing

For entities which were encountered during training, e_{seen}, we use an informative prior based on the inferred topic counts $C_1^{e_{\text{seen}}}, \ldots, C_T^{e_{\text{seen}}}$ from the training data:

$$\theta_{e_{\text{seen}}}^{\text{test}} \quad \sim \quad Dir(C_1^{e_{\text{seen}}}, \ldots, C_T^{e_{\text{seen}}})$$

For those entities which are new (haven't yet been encountered), we use an informative prior to encode the overall distribution over types in the training data. To avoid an overwhelming prior which prefers the most frequent class, we use concentration parameter of 1 (that is we normalize by the total number of words in the training data, N_{train}):

$$\theta_{e_{\text{new}}}^{\text{test}} \quad \sim \quad Dir(C_1/N_{\text{train}}, \ldots, C_T/N_{\text{train}})$$

We run 100 iterations of Gibbs sampling over each test "document" to infer posterior distributions over θ_e^{test} Each entity mention's type is predicted based on the topic with highest posterior probability.

4.4.3 Results

Table 4.2 presents the classification results of T-CLASS compared against a majority baseline which simply picks the most frequent class (*PERSON*), in addition to the Freebase baseline, which only makes predictions if an entity appears in exactly one dictionary (*i.e.*, appears unambiguous). T-CLASS also outperforms a simple supervised baseline which applies a

System	P	R	F_1
Majority Baseline	0.30	0.30	0.30
Freebase Baseline	**0.85**	0.24	0.38
Supervised Baseline	0.45	0.44	0.45
DL-Cotrain	0.54	0.51	0.53
LabeledLDA	0.72	**0.60**	**0.66**

Table 4.2: Named Entity Classification performance on the 10 types.

Type	LL	FB	CT	SP	N
PERSON	0.82	0.48	0.65	**0.83**	436
LOCATION	**0.74**	0.21	0.55	0.67	372
ORGANIZATION	**0.66**	0.52	0.55	0.31	319
overall	**0.75**	0.39	0.59	0.49	1127

Table 4.3: F_1 classification scores for the 3 MUC types *PERSON*, *LOCATION*, *ORGANIZATION*. Results are shown using LabeledLDA (LL), Freebase Baseline (FB), DL-Cotrain (CT) and Supervised Baseline (SP). N is the number of entities in the test set.

MaxEnt classifier using 4-fold cross validation over the 1,450 entities which were annotated for testing. Additionally we compare against the co-training algorithm of Collins and Singer [34] which also leverages unlabeled data and uses our Freebase type lists; for seed rules we use the "unambiguous" Freebase entities. Our results demonstrate that T-CLASS outperforms the baselines and achieves a 25% increase in F_1 score over co-training.

Tables 4.3 and 4.4 present a breakdown of F_1 scores by type, both collapsing types into the standard classes used in the MUC competitions (*PERSON*, *LOCATION*, *ORGANIZATION*), and using the 10 popular Twitter types described earlier.

We observe that T-CLASS outperforms the other methods except in the case of *MOVIE*s, where the Freebase baseline performs better. We also notice that classes like *BAND* and *FACILITY* are harder to identify, since they often have ambiguous names.

LabeledLDA is winning for two reasons: first, note that it is able to share information

Type	LL	FB	CT	SP	N
PERSON	0.82	0.48	0.65	**0.86**	436
GEO-LOC	**0.77**	0.23	0.60	0.51	269
COMPANY	**0.71**	0.66	0.50	0.29	162
FACILITY	**0.37**	0.07	0.14	0.34	103
PRODUCT	**0.53**	0.34	0.40	0.07	91
BAND	**0.44**	0.40	0.42	0.01	54
SPORTSTEAM	**0.53**	0.11	0.27	0.06	51
MOVIE	0.54	**0.65**	0.54	0.05	34
TV-SHOW	**0.59**	0.31	0.43	0.01	31
OTHER	**0.52**	0.14	0.40	0.23	219
overall	**0.66**	0.38	0.53	0.45	1450

Table 4.4: F_1 scores for classification broken down by type for LabeledLDA (LL), Freebase Baseline (FB), DL-Cotrain (CT) and Supervised Baseline (SP). N is the number of entities in the test set.

	P	R	F_1
DL-Cotrain-entity	0.47	0.45	0.46
DL-Cotrain-mention	0.54	0.51	**0.53**
LabeledLDA-entity	0.73	0.60	**0.66**
LabeledLDA-mention	0.57	0.52	0.54

Table 4.5: Comparing LabeledLDA and DL-Cotrain grouping unlabeled data by entities vs. mentions.

about an entity's likely types across mentions, which is very beneficial for the common case where there simply isn't enough context contained within a Tweet to disambiguate the type of entities it mentions. Secondly, because LabeledLDA makes use of the Freebase dictionaries as constraints, it is able to better exploit this highly ambiguous source of training data, rather than relying on unambiguous mentions for learning.

4.4.4 Entity Strings vs. *Entity Mentions*

DL-Cotrain and LabeledLDA use two different representations for the unlabeled data during learning. LabeledLDA groups together words across all mentions of an entity string, and infers a distribution over its possible types, whereas DL-Cotrain considers the entity mentions separately as unlabeled examples and predicts a type independently for each. In order to ensure that the difference in performance between LabeledLDA and DL-Cotrain is not simply due to this difference in representation, we compare both DL-Cotrain and LabeledLDA using both unlabeled datasets (grouping words by all mentions vs. keeping mentions separate) in Table 4.5. As expected, DL-Cotrain performs poorly when the unlabeled examples group mentions; this makes sense, since Co-Training uses a discriminative learning algorithm, so when trained on entities and tested on individual mentions, the performance decreases. Additionally, LabeledLDA's performance is poorer when considering mentions as "documents". This is likely due to the fact that there isn't enough context to effectively learn topics when the "documents" are very short (typically fewer than 10 words).

4.4.5 End to End Evaluation

Finally we present the end to end performance on segmentation and classification (T-NER) in Table 4.6. We observe that T-NER again outperforms co-training. Moreover, comparing against the Stanford Named Entity Recognizer on the 3 MUC types, T-NER doubles F_1 score.

System	P	R	F_1
COTRAIN-NER (10 types)	0.55	0.33	0.41
T-NER(10 types)	0.65	0.42	**0.51**
COTRAIN-NER (PLO)	0.57	0.42	0.49
T-NER(PLO)	0.73	0.49	**0.59**
Stanford NER (PLO)	0.30	0.27	0.29

Table 4.6: Performance at predicting both segmentation and classification. Systems labeled with PLO are evaluated on the 3 MUC types *PERSON*, *LOCATION*, *ORGANIZATION*.

4.5 Related Work

Previous work on Semantic Bootstrapping has taken a weakly-supervised approach to classifying named entities based on large amounts of unlabeled text [59; 23; 90; 174; 116]. In contrast, rather than predicting which classes an entity belongs to (e.g. a multi-label classification task), LabeledLDA estimates a *distribution* over its types, which is then useful as a prior when classifying mentions in context.

In addition there has been been work on Skip-Chain CRFs [172; 62] which enforce consistency when classifying multiple occurrences of an entity within a document. Using topic models (e.g. LabeledLDA) for classifying named entities has a similar effect, in that information about an entity's distribution of possible types is shared across its mentions.

We have taken a distantly supervised approach to Named Entity Classification which exploits large dictionaries of entities gathered from Freebase, requires no manually annotated data, and as a result is able to handle a larger number of types than previous work. Although we found manually annotated data to be very beneficial for named entity segmentation, we were motivated to explore approaches that don't rely on manual labels for classification due to Twitter's wide range of named entity types. Additionally, unlike previous work on NER in informal text, our approach allows the sharing of information across an entity's mentions which is quite beneficial due to Twitter's terse nature.

Finally, the idea of distant supervision was originally proposed in the context of relation

extraction [121] leveraging Freebase relations (as opposed to types) as a source of distant supervision.

4.6 Limitations and Future Work

Our proposed approach is capable of learning to categorize named entities from large amounts of naturally occurring unlabeled data in addition to highly ambiguous dictionaries. There are some limitations to this approach however which open up potential avenues for future work which we now outline.

First, because our approach is based on generative probabilistic models, it is generally not amenable to feature engineering. Our model makes strong independence assumptions about the features, so this approach is not applicable to situations where there are highly correlated and overlapping feature sets. This is the case for the task of extracting binary relations where commonly used features include dependency paths from parse trees in addition to word and part of speech sequences. In contrast conditionally trained models don't make independence assumptions about the features, and are therefore more appropriate for situations where feature engineering is particularly beneficial such as binary relation extraction. In Chapter 5, therefore, we investigate conditionally trained models for binary relation extraction.

The pipelined architecture described in Chapter 3 and the present chapter has several advantages: it makes inference tractable, and allows appropriate models to be applied to each task: conditional random fields are appropriate for the task of named entity segmentation as they enable feature engineering, whereas generative models are appropriate for weakly supervised named entity categorization. Despite these advantages, a limitation that stems from the pipelined approach, is the propagation of errors. Errors made by the segmenter cannot be corrected by the classifier and vice versa. In contrast a joint model of segmentation and weakly supervised named entity categorization could potentially produce better end-to-end performance by sharing information across these two tasks. Of course this will complicate the inference and learning, so it is an open question whether this approach will produce better results in the end, but this could be an interesting avenue for future work.

A final limitation we point out with our model is the issue of missing data which is

discussed in detail in Chapter 5. If a named entity is listed in our Freebase dictionaries, we assume it can only be classified as one of the possible categories in our data. That is, we make the assumption that Freebase has complete coverage for the set of possible types for any string that it contains. This assumption (which has so far been made by all distantly supervised approaches to learning), effectively leads to errors in the training data. In Chapter 5, we show that by relaxing this assumption we are able to dramatically improve precision and recall at extracting binary relations using distant supervision.

4.7 Conclusions

We have identified named entity classification as a particularly challenging task on Twitter. Due to their terse nature, tweets often lack enough context to identify the types of the entities they contain. In addition, a plethora of distinctive named entity types are present, necessitating large amounts of training data. To address both these issues we have presented and evaluated a distantly supervised approach based on LabeledLDA, a constrained latent variable model, which obtains a 25% increase in F_1 score over the co-training approach to Named Entity Classification suggested by Collins and Singer [34] when applied to Twitter.

By modeling large dictionaries of named entities from Freebase, in addition to a large corpus of named entities which is automatically extracted from Twitter using latent variables, we are able to build a state-of-the art named entity classification system for Twitter. We showed this approach outperforms both state-of-the art NLP tools which are tuned for newswire text in addition to a supervised baseline.

This is one instance of a semantic processing task where weakly supervised latent variable models perform better than supervised learning on small manually annotated datasets. More evidence that unsupervised and weakly supervised methods are appropriate for semantic processing are presented in later chapters.

We have made our named entity classification models which are trained on millions of Tweets available as part of a suite of NLP tools tuned for Twitter[2].

[2] `https://github.com/aritter/twitter_nlp`

Chapter 5

MODELING MISSING DATA IN DISTANT SUPERVISION

In Chapter 4 we presented a new approach to distant supervision based on topic models which is appropriate for highly ambiguous training data. This chapter addresses the question of missing data [107] in the context of distant supervision, which is a very general issue affecting distant supervision for named entity categorization and also extracting binary relations [185; 121; 148; 80; 173; 171].

Distant supervision algorithms learn information extraction models given only large readily available databases and text collections. Most previous work has used heuristics for generating labeled data, for example assuming that facts not contained in the database are not mentioned in the text, and facts in the database must be mentioned at least once. In this chapter, we propose a new latent-variable approach that models missing data. This provides a natural way to incorporate side information, for instance modeling the intuition that text will often mention rare entities which are likely to be missing in the database. Despite the added complexity introduced by reasoning about missing data, we demonstrate that a carefully designed local search approach to inference is very accurate and scales to large datasets. Experiments demonstrate improved performance for binary and unary relation extraction when compared to learning with heuristic labels, including on average a 27% increase in area under the precision recall curve in the binary case.

5.1 Introduction

This chapter addresses the issue of missing data [107] in the context of distant supervision. The goal of distant supervision is to learn to extract relations, for example binary relations [19; 185; 121] or unary relations such as named entity categories [34], from unlabeled text corpora using a large database of propositions involving the target relations as a distant source of supervision. In the case of binary relations, the intuition is that any sentence

Person	EMPLOYER
Bibb Latané	UNC Chapel Hill
Tim Cook	Apple
Susan Wojcicki	Google

True Positive	*"***Bibb Latané***, a professor at the* **University of North Carolina at Chapel Hill***, published the theory in 1981."*
False Positive	*"***Tim Cook** *praised* **Apple***'s record revenue..."*
False Negative	*"***John P. McNamara***, a professor at* **Washington State University***'s Department of Animal Sciences..."*

Figure 5.1: A small hypothetical database and heuristically labeled training data for the EMPLOYER relation.

which mentions a pair of entities (e_1 and e_2) that participate in a relation, r, is likely to express the proposition $r(e_1, e_2)$, so we can treat it as a positive training example of r. Figure 5.1 presents an example of this process.

One question which has received little attention in previous work is how to handle the situation where information is missing, either from the text corpus, or the database. As an example, let us assume the pair of entities (*John P. McNamara*, *Washington State University*) is absent from the EMPLOYER relation in our database. In this case, the sentence in Figure 5.1 (and others which mention the entity pair) is effectively treated as a negative example of the relation. This is an issue of practical concern, as most databases of interest are highly incomplete - this is the reason we need to extend them by extracting information from text in the first place.

As an example where the MAR assumption is violated, imagine estimating the probability of a coin flip landing heads, $\theta_H = P(H)$, from an observed sequence of flips with missing data which depends on each outcome: the result of a flip is hidden with probability 0.5 if the coin lands tails, and 0.2 if it is heads. It is clear that we need to explicitly model the process by which data is missing in order to obtain an accurate estimate of θ_H. A

similar problem arises when estimating the parameters of a relation extractor using distant supervision: whether a proposition is observed or missing in the text or database depends heavily on its truth value; given that a fact is true we have some chance to observe it, however we never observe which facts are false. In this case our data are *not missing at random*, because the probability of observing a proposition depends on its truth value.

We need to be cautious in how we handle missing data in distant supervision, because this is a case where data is *not missing at random* (NMAR). Whether a proposition is observed or missing in the text or database depends heavily on its truth value: given that it is true we have some chance to observe it, however we do not observe those which are false. To address this challenge, we propose a joint model of extraction from text and the process by which propositions are observed or missing in both the database and text. Our approach provides a natural way to incorporate *side information* in the form of a *missing data model*. For instance, popular entities such as Barack Obama already have good coverage in Freebase, so new extractions are more likely to be errors than those involving rare entities with poor coverage.

Our approach to missing data is general and can be combined with various IE solutions. As a proof of concept, we extend MultiR [80], a recent model for distantly supervised information extraction, to explicitly model missing data. These extensions complicate the MAP inference problem which is used as a subroutine in learning. This motivated us to explore a variety of approaches to inference in the joint extraction and missing data model. We explore both exact inference based on A* search and efficient approximate inference using local search. Our experiments demonstrate that with a carefully designed set of search operators, local search produces optimal solutions in most cases.

Experimental results demonstrate large performance gains over the heuristic labeling strategy on both binary relation extraction and weakly supervised named entity categorization. For example our model obtains a 27% increase in area under the precision recall curve on the sentence-level relation extraction task.

Symbol	Description
(e_1, e_2)	a pair of entities
$\mathbf{s}$	The set of sentences which mention e_1 and e_2
$\mathbf{z}$	The set of relations that are mentioned in $\mathbf{s}$
θ	Parameters of the extractor
$\phi(z_i, s_i; \theta)$	Sentence-level extraction factors
$f(z_i, s_i)$	Features
$\omega(\mathbf{z}, d_j)$	Aggregate hard constraint factors
d_j	Binary variable indicating whether the proposition $r_j(e_1, e_2)$ is mentioned in Freebase
t_j	Binary variable indicating whether the proposition $r_j(e_1, e_2)$ is mentioned in the text corpus
$\psi(t_j, d_j)$	Pairwise potentials which penalize disagreement between t_j and d_j

Table 5.1: A summary of notation used in this chapter.

5.2 Related Work

There has been much interest in distantly supervised[1] training of relation extractors using databases. For example, Craven and Kumlien [35] build a heuristically labeled dataset, using the Yeast Protein Database to label Pubmed abstracts with the *subcellular-localization* relation. Wu and Weld [185] heuristically annotate Wikipedia articles with facts mentioned in the infoboxes, enabling automated infobox generation for articles which do not yet contain them. Benson et. al. [8] use a database of music events taking place in New York City as a source of distant supervision to train event extractors from Twitter. Mintz et. al. [121] used a set of relations from Freebase as a distant source of supervision to learn to extract information from Wikipedia. Ridel et. al. [148], Hoffmann et. al. [80], and Surdeanu et. al. [171] presented a series of models casting distant supervision as a multiple-instance learning problem [45].

Recent work has begun to address the challenge of noise in heuristically labeled training data generated by distant supervision, and proposed a variety of strategies for correcting erroneous labels. Takamatsu et al. [173] present a generative model of the labeling process, which is used as a pre-processing step for improving the quality of labels before training relation extractors. Independently, Xu et. al. [187] analyze a random sample of 1834 sentences from the New York Times, demonstrating that most entity pairs expressing a Freebase relation correspond to false negatives. They apply pseudo-relevance feedback to add missing entries in the knowledge base before applying the MultiR model [80]. Min et al. [119] extend the MIML model of Surdeanu et. al. [171] using a semi-supervised approach assuming a fixed proportion of true positives for each entity pair. In contrast to previous work which has aimed to correct labeling noise, we cast this as a missing data problem in which information is not missing at random [107]. We propose a joint model of relation extraction and observation of facts in both the text in the database; this addresses both the problem of false negatives and false positives.

The issue of missing data has been extensively studied in the statistical literature [107; 67]. Most methods for handling missing data assume that variables are *missing at random*

[1] also referred to as weakly supervised

(MAR): whether a variable is observed does not depend on its value. In situations where the MAR assumption is violated (for example distantly supervised information extraction), ignoring the missing data mechanism will introduce bias. In this case it is necessary to jointly model the process of interest (e.g. information extraction) in addition to the missing data mechanism.

Another line of related work is iterative semantic bootstrapping [16; 2]. Carlson et. al. [22] exploit constraints between relations to reduce semantic drift in the bootstrapping process; such constraints are potentially complementary to our approach of modeling missing data.

5.3 A Latent Variable Model for Distantly Supervised Relation Extraction

In this section we review the MultiR model (due to Hoffmann et. al. [80]) for distant supervision in the context of extracting binary relations. This model is extended to handle missing data in Section 5.4. We focus on binary relations to keep discussions concrete; unary relation extraction is also possible.

Given a set of sentences, $\mathbf{s} = s_1, s_2, \ldots, s_n$, which mention a specific pair of entities (e_1 and e_2) our goal is to correctly predict which relation is mentioned in each sentence, or "NA" if none of the relations under consideration are mentioned. Unlike the standard supervised learning setup, we do not observe the latent sentence-level relation mention variables, $\mathbf{z} = z_1, z_2, \ldots, z_n$ [2] Instead we only observe *aggregate binary variables* for each relation, $\mathbf{d} = d_1, d_2, \ldots, d_k$, which indicate whether the proposition $r_j(e_1, e_2)$ is present in the database (Freebase). Of course the question which arises is: how do we relate the aggregate-level variables, d_j, to the sentence-level relation mentions, z_i? A sensible answer to this question is a simple deterministic-OR function. The deterministic-OR states that if there exists at least one i such that $z_i = j$, then $d_j = 1$. For example, if at least one sentence mentions that *"Barack Obama was born in Honolulu"*, then that fact is true in aggregate, if none of the sentences mentions the relation, then the fact is assumed false. The model also makes the converse assumption: if Freebase contains the relation BirthLocation(*Barack*

[2] These variables indicate which relation is mentioned between e_1 and e_2 in each sentence.

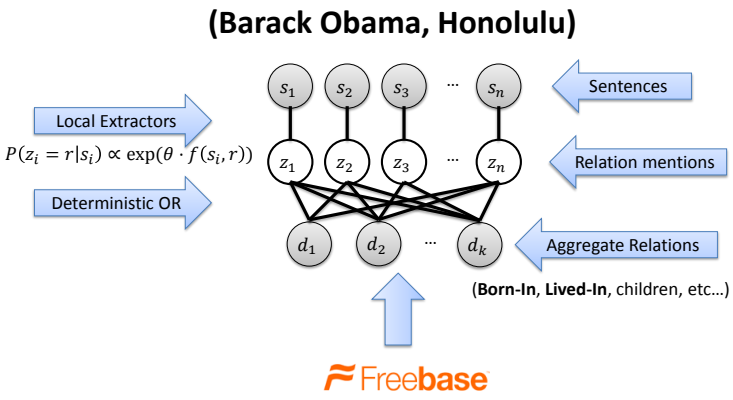

Figure 5.2: MultiR (Hoffmann et. al. 2011)

Obama, *Honolulu*), then we must extract it from at least one sentence. A summary of this model, which is due to Hoffmann et. al. [80] is presented in Figure 5.2.

5.3.1 Learning

To tune the parameters of the sentence-level relation mention classifier, θ, we maximize the likelihood of the facts observed in Freebase conditioned on the sentences in our text corpus:

$$\begin{aligned} \theta^* &= \arg\max_{\theta} P(\mathbf{d}|\mathbf{s};\theta) \\ &= \arg\max_{\theta} \prod_{e_1,e_2} \sum_{\mathbf{z}} P(\mathbf{z},\mathbf{d}|\mathbf{s};\theta) \end{aligned}$$

Here the conditional likelihood of a given entity pair is defined as follows:

$$\begin{aligned} P(\mathbf{z},\mathbf{d}|\mathbf{s};\theta) &= \prod_{i=1}^{n} \phi(z_i, s_i;\theta) \times \prod_{j=1}^{k} \omega(\mathbf{z}, d_j) \\ &= \prod_{i=1}^{n} e^{\theta \cdot f(z_i,s_i)} \times \prod_{j=1}^{k} \mathbf{1}_{\neg d_j \oplus \exists i: j=z_i} \end{aligned}$$

Where $\mathbf{1}_x$ is an indicator variable which takes the value 1 if x is true and 0 otherwise, the $\omega(\mathbf{z}, d_j)$ factors are hard constraints corresponding to the deterministic-OR function, and $f(z_i, s_i)$ is a vector of features extracted from sentence s_i and relation z_i.

An iterative gradient-ascent based approach is used to tune θ using a latent-variable perceptron-style additive update scheme [33; 99; 196]. The gradient of the conditional log likelihood, for a single pair of entities, e_1 and e_2, is as follows:[3]

$$\frac{\partial \log P(\mathbf{d}|\mathbf{s};\theta)}{\partial \theta} = \mathbf{E}_{P(\mathbf{z}|s,d;\theta)}\left(\sum_j f(s_j, z_j)\right) - \mathbf{E}_{P(\mathbf{z},\mathbf{d}|\mathbf{s};\theta)}\left(\sum_j f(s_j, z_j)\right)$$

These expectations are too difficult to compute in practice, so instead they are approximated as maximizations:

$$\frac{\partial \log P(\mathbf{d}|\mathbf{s};\theta)}{\partial \theta} \approx \sum_j f(s_j, z_j^{*\text{DB}}) - \sum_j f(s_j, z_j^*)$$

Computing this approximation to the gradient requires solving two inference problems corresponding to the two maximizations:

$$\begin{aligned} \mathbf{z}^{*\text{DB}} &= \arg\max_{\mathbf{z}} P(\mathbf{z}|\mathbf{s}, \mathbf{d}; \theta) \\ \mathbf{z}^* &= \arg\max_{\mathbf{z}} P(\mathbf{z}, \mathbf{d}|\mathbf{s}; \theta) \end{aligned}$$

The MAP solution for the second term is easy to compute: because $\mathbf{d}$ and $\mathbf{z}$ are deterministically related, we can simply find the highest scoring relation, r, for each sentence, s_i, according to the sentence-level factors, ϕ, independently:

$$\begin{aligned} z_i &= \arg\max_r \phi(r, s_i) \\ &= \arg\max_r \exp\left(\theta \cdot f(r, s_i)\right) \end{aligned}$$

The first term, is more difficult, however, as this requires finding the best assignment to the sentence-level hidden variables $\mathbf{z} = z_1 \ldots z_n$ conditioned on the observed sentences and facts in the database. Hoffmann et. al. [80] show how this reduces to a well-known weighted edge cover problem which can be solved exactly in polynomial time.

5.4 Modeling Missing Data

The model presented in Section 5.3 makes two assumptions which correspond to hard constraints:

[3] For details see Koller and Friedman [87], Chapter 20.

1. If a fact is not found in the database it can't be mentioned in the text.

2. If a fact is in the database it must be mentioned in at least one sentence.

These assumptions drive the learning, however if there is information missing from either the text or the database this leads to errors in the training data (false positives, and false negatives respectively).

In order to gracefully handle the problem of missing data, we propose to extend the model presented in Section 5.3 by splitting the aggregate level variables, $\mathbf{d}$, into two parts: $\mathbf{t}$ which represents whether a fact is mentioned in the text (in at least one sentence), and $\mathbf{d}'$ which represents whether the fact is mentioned in the database. We introduce pairwise potentials $\psi(t_j, d_j)$ which penalize disagreement between t_j and d_j, that is:

$$\psi(t_j, d_j) = \begin{cases} -\alpha_{\text{MIT}} & \text{if } t_j = 0 \text{ and } d_j = 1 \\ -\alpha_{\text{MID}} & \text{if } t_j = 1 \text{ and } d_j = 0 \\ 0 & \text{otherwise} \end{cases}$$

Where α_{MIT} (**M**issing **I**n **T**ext) and α_{MID} (**M**issing **I**n **D**atabase) are parameters of the model which can be understood as penalties for missing information in the text and database respectively. We refer to this model as DNMAR (for **D**istant Supervision with Data **N**ot **M**issing **A**t **R**andom). A graphical model representation is presented in Figure 5.3.

This model can be understood as relaxing the two hard constraints mentioned above into soft constraints. As we show in Section 5.7, simply relaxing these hard constraints into soft constraints and setting the two parameters α_{MIT}, and α_{MID} by hand on development data results in a large improvement to precision at comparable recall over MultiR on two different applications of distant supervision: binary relation extraction and named entity categorization.

Inference in this model becomes more challenging however, because the constrained inference problem no longer reduces to a weighted edge cover problem as before. In Section 5.5, we present an inference technique for the new model which is time and memory efficient and almost always finds an exact MAP solution.

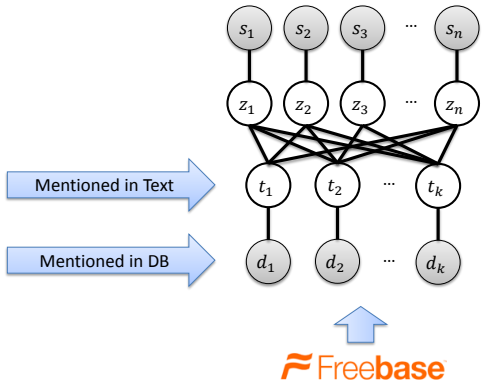

Figure 5.3: DNMAR

The learning proceeds analogously to what was described in section 5.3.1, with the exception that we now maximize over the additional aggregate-level hidden variables $\mathbf{t}$, which have been introduced. The parameter updates are now as follows:

$$\frac{\partial P(\mathbf{d}|\mathbf{s};\theta)}{\partial \theta} \approx \max_{\mathbf{z},\mathbf{t}\ P(\mathbf{z},\mathbf{t}|\mathbf{s},\mathbf{d};\theta)} \left(\sum_j f(s_j, z_j) \right) - \max_{\mathbf{z},\mathbf{t}\ P(\mathbf{z},\mathbf{t},\mathbf{d}|\mathbf{s};\theta)} \left(\sum_j f(s_j, z_j) \right)$$

As before, MAP inference is a subroutine in learning, both for the unconstrained case corresponding to the second term (which is again trivial to compute), and for the constrained case which is more challenging as it no longer reduces to a weighted edge cover problem as before.

5.5 MAP Inference

The only difference in the new inference problem is the addition of $\mathbf{t}$; $\mathbf{z}$ and $\mathbf{t}$ are deterministically related, so we can simply find a MAP assignment to $\mathbf{z}$, from which $\mathbf{t}$ follows. The resulting inference problem can be viewed as optimization under soft constraints, where the objective includes terms for each fact *not* in Freebase which is extracted from the text: $-\alpha_{\text{MID}}$, and an effective reward for extracting a fact which *is* contained in Freebase: α_{MIT}.

The solution to the MAP inference problem is the value of $\mathbf{z}$ which maximizes the

following objective:

$$\begin{aligned}
\mathbf{z}^{*\text{DB}} &= \arg\max_{\mathbf{z}} P(\mathbf{z}|\mathbf{d};\theta,\alpha) \\
&= \arg\max_{\mathbf{z}} \prod_{i=1}^{n} \phi(z_i, s_i;\theta) \times \prod_{j=1}^{k} \omega(\mathbf{z}, t_j)\psi(t_j, d_j;\alpha) \\
&= \arg\max_{\mathbf{z}} \prod_{i=1}^{n} e^{\theta \cdot f(z_i, s_i)} \times \prod_{j=1}^{k} e^{-\alpha_{\text{MIT}} \mathbf{1}_{d_j \wedge \neg\exists i: j=z_i} - \alpha_{\text{MID}} \mathbf{1}_{\neg d_j \wedge \exists i: j=z_i}} \\
&= \arg\max_{\mathbf{z}} \sum_{i=1}^{n} \theta \cdot f(z_i, s_i) + \sum_{j=1}^{k} \left(\alpha_{\text{MIT}} \mathbf{1}_{d_j \wedge \exists i: j=z_i} - \alpha_{\text{MID}} \mathbf{1}_{\neg d_j \wedge \exists i: j=z_i} \right) \qquad (5.1)
\end{aligned}$$

the last step comes from taking the log.

Note that α_{MIT} functions as a *reward* for extracting a fact in Freebase as opposed to a penalty for not extracting it. This formulation makes the optimization algorithms easier to describe and implement, and the objective is equivalent up to an additive constant. The MAP values of $\mathbf{z}$, will therefore be the same under both objectives.

Whether we choose to set the parameters α_{MIT} and α_{MID} to fixed values (Section 5.4), or incorporate side information through a missing data model (Section 5.6), inference becomes more challenging than in the model where facts observed in Freebase are treated as hard constraints (Section 5.3); the hard constraints are equivalent to setting $\alpha_{\text{MID}} = \alpha_{\text{MIT}} = \infty$.

We now present exact and approximate approaches to inference. Standard search methods such as A* and branch and bound have high computation and memory requirements and are therefore only feasible on problems with few variables; they are, however, guaranteed to find an optimal solution.[4] Approximate methods scale to large problem sizes, but we loose the guarantee of finding an optimal solution. After showing how to find guaranteed exact solutions for small problem sizes (e.g. up to 200 variables), we present an inference algorithm based on local search which is empirically shown to find optimal solutions in almost every case by comparing its solutions to those found by A*.

---

[4]Each entity pair defines an inference problem where the number of variables is equal to the number of sentences which mention the pair.

5.5.1 A Search*

We cast exact MAP inference in the DNMAR model as an application of A* search. Each partial hypothesis, h, in the search space corresponds to a partial assignment of the first m variables in $\mathbf{z}$; to expand a hypothesis, we generate k new hypotheses, where k is the total number of relations. Each new hypothesis h' contains the same partial assignment to $z_1, \ldots, z_m$ as h, with each h' having a different value of $z_{m+1} = r$.

A* operates by maintaining a priority queue of hypotheses to expand, with each hypothesis' priority determined by an admissible heuristic. The heuristic represents an upper bound on the score of the best solution with h's partial variable assignment under the objective from Equation 5.1. In general, a tighter upper bound corresponds to a better heuristic and faster solution. To upper bound our objective, we start with the $\phi(z_i, s_i)$ factors from the partial assignment. Unassigned variables ($i > k$), are set to their maximum possible value, $z_i = \max_r \phi(r, s_i)$ independently. Next to account for the effect the aggregate $\psi(t_j, d_j)$ factors on the unassigned variables, we consider independently changing each unassigned z_i variable for each $\psi(t_j, d_j)$ factor to improve the overall score. This approach can lead to inconsistencies, but provides us with a good upper bound for the best possible solution with a partial assignment to $z_1, \ldots, z_k$.

5.5.2 Local Search

While A* is guaranteed to find an exact solution, its time and memory requirements prohibit use on large problems involving many variables. As a more scalable alternative we propose a greedy hill climbing method [158], which starts with a full assignment to $\mathbf{z}$, and repeatedly moves to the best neighboring solution $\mathbf{z}'$ according to the objective in Equation 5.1. The neighborhood of $\mathbf{z}$ is defined by a set of *search operators*. If none of the neighboring solutions has a higher score, then we have reached a (local) maximum at which point the algorithm terminates with the current solution which may or may not correspond to a global maximum. This process is repeated using a number of *random restarts*, and the best local maximum is returned as the solution.

Algorithm 5 Guaranteed Exact MAP inference

function DNMAR-A*(e_1,e_2,$s_1 \dots s_n, d_1 \dots d_k$)

Initialize priority queue Q

Insert the empty hypothesis, h_{null}, into Q

while Q is not empty **do**

Remove the current best hypothesis, $h^*_{z_1 \dots z_m}$, from Q

if $m = n$ **then**

Return $h^*_{z_1 \dots z_m}$

end if

for all Relations r **do**

Create a new hypothesis $h'_{z_1 \dots z_m, r}$

$p = \text{heuristic}(h')$ ▷ heuristic() is an upper bound on Equation 5.1

Insert h' into Q with priority p

end for

end while

end function

Algorithm 6 Approximate MAP inference (almost always produces exact solution)

function DNMAR-LS(e_1,e_2,$s_1 \dots s_n, d_1 \dots d_k$)

choose a random $\mathbf{z}^* = z_1^* \dots z_n^*$ ▷ Random initialization

while True **do**

for $i = 1 \to n$ **do**

for $j = 1 \to k$ **do** ▷ Standard Search Operator

$\mathbf{z}_{ij}^1 = z_1^* \dots z_{i-1}^*, j, z_{i+1}^*, \dots z_n^*$

end for

end for

for $j_1 = 1 \to k$ **do**

for $j_2 = 1 \to k$ **do** ▷ Aggregate Search Operator

$\mathbf{z}_{j_1 j_2}^2 = z^*$

for $z \in \mathbf{z}_{j_1 j_2}^2 | z = j_1$ **do**

$z = j_2$

end for

end for

end for

$z' = \arg\max_{z \in z^1 \cup z^2} \text{score}(z)$ ▷ Choose the best neighboring solution

if score(z') > score(z^*) **then**

$z^* = z'$

else

Return z^* ▷ Local maximum

end if

end while

end function

Search Operators: We start with a standard search operator, which considers changing each relation-mention variable, z_i, individually to maximize the overall score. During each iteration, all z_is are considered, and the one which produces the largest improvement to the overall score is changed to form the neighboring solution, $\mathbf{z}'$. Unfortunately, this definition of the solution neighborhood is prone to poor local optima because it is often required to traverse many low scoring states before changing one of the aggregate variables, t_j, and achieving a higher score from the associated aggregate factor, $\psi(t_j, d_j)$. For example, consider a case where the proposition $r(e_1, e_2)$ is not in Freebase, but is mentioned many times in the text, and imagine the current solution contains no mention $z_i = r$. Any neighboring solution which assigns a mention to r will include the penalty α_{MID}, which could outweigh the benefit from changing any individual z_i to r: $\phi(r, s_i) - \phi(z_i, s_i)$. If multiple mentions were changed to r however, together they could outweigh the penalty for extracting a fact not in Freebase, and produce an overall higher score.

To avoid the problem of getting stuck in local optima, we propose an additional search operator which considers changing *all* variables, z_i, which are currently assigned to a specific relation r, to a new relation r', resulting in an additional $(k-1)^2$ possible neighbors, in addition to the $n \times (k-1)$ neighbors which come from the standard search operator. This aggregate-level search operator allows for more global moves which help to avoid local optima, similar to the type-level sampling approach for MCMC [100].

During each iteration, we consider all $n \times (k-1) + (k-1)^2$ possible neighboring solutions generated by both search operators, and pick the one with biggest overall improvement, or terminate the algorithm if no improvements can be made over the current solution. 20 random restarts were used for each inference problem. We found this approach to almost always find an optimal solution. In over 100,000 problems with 200 or fewer variables from the New York Times dataset used in Section 5.7, an optimal solution was missed in only 3 cases which was verified by comparing against optimal solutions found using A*. Without including the aggregate-level search operator, local search almost always gets stuck in a local maximum.

5.5.3 Discussion

Algorithm 5 can be easily modified to produce approximate solutions efficiently by setting a maximum beam width. When the length of the queue becomes larger than the maximum width, the lowest scoring hypotheses are dropped. We found the approximate inference method described in Algorithm 6 to more reliably find exact solutions while maintaining scalability, however.

5.6 Incorporating Side Information

In Section 5.4, we relaxed the hard constraints made by MultiR as soft constraints, which allows for missing information in either the text or database, enabling errors in the distantly supervised training data to be naturally corrected as a side-effect of learning. We made the simplifying assumption, however, that all facts are equally likely to be missing from the text or database, which is encoded in the choice of 2 fixed parameters α_{MIT}, and α_{MID}. Is it possible to improve performance by incorporating side information in the form of a missing data model [107], taking into account how likely each fact is to be observed in the text and the database conditioned on its truth value? In our setting, the missing data model corresponds to choosing the values of α_{MIT} and α_{MID} dynamically based on the entities and relations involved.

Popular Entities: Consider two entities: Barack Obama, the 44th president of the United States, and Donald Parry, a professional rugby league footballer of the 1980s[5]. Since Obama is much more well-known than Parry, we wouldn't be very surprised to see information missing from Freebase about Parry, but it would seem odd if true facts were missing about Obama.

We can encode these intuitions by choosing entity-specific values of α_{MID}:

$$\alpha_{\text{MID}}^{(e_1,e_2)} = -\gamma \min\left(c(e_1), c(e_2)\right)$$

where $c(e_i)$ is the count of e_i in Freebase, which is used as an estimate of its coverage.

Well Aligned Relations: Given that a pair of entities, e_1 and e_2, participating in a

[5] `http://en.wikipedia.org/wiki/Donald_Parry`

Freebase relation, r, appear together in a sentence s_i, the chance that s_i expresses r varies greatly depending on r. If a sentence mentions a pair of entities which participate in both the CountryCapitol relation and the LocationContains relation (for example Moscow and Russia), it is more likely that a random sentence will express LocationContains than CountryCapitol.

We can encode this preference for matching certain relations over others by adjusting α^r_{MIT} on a per-relation basis. We choose a different value of α^r_{MIT} for each relation based on quick inspection of the data, and estimating the number of true positives. Relations such as *contains*, *place_lived*, and *nationality* which contain a large number of true positive matches are assigned a large value of $\alpha^r_{\text{MIT}} = \gamma_{\text{large}}$, those with a medium number such as *capitol*, *place_of_death* and *administrative_divisions* were assigned a medium value γ_{medium}, and those relations with few matches were assigned a small value γ_{small}. These 3 parameters were tuned on held out development data.

5.7 Experiments

In Section 5.5, we presented a scalable approach to inference in the DNMAR model which almost always finds an optimal solution. Of course the real question is: does modeling missing data improve performance at extracting information from text? In this section we present experimental results showing large improvements in both precision and recall on two distantly supervised learning tasks: binary relation extraction and named entity categorization.

5.7.1 Binary Relation Extraction

For the sake of comparison to previous work we evaluate performance on the New York Times text, features and Freebase relations developed by Riedel et. al. [148] which was also used by Hoffmann et. al. [80]. This dataset is constructed by extracting named entities from 1.8 million New York Times articles, which are then match against entities in Freebase. Sentences which contain pairs of entities participating in one or more relations are then used as training examples for those relations. The sentence-level features include word sequences

appearing in context with the pair of entities, in addition to part of speech sequences, and dependency paths from the Malt parser [132].

Baseline

To evaluate the effect of modeling missing data in distant supervision, we compare against the MultiR model for distant supervision [80], a state of the art approach for binary relation extraction which is the most similar previous work, and models facts in Freebase as hard constraints disallowing the possibility of missing information in either the text or the database. To make our experiment as controlled as possible and rule-out the possibility of differences in performance due to implementation details, we compare against our own re-implementation of MultiR which reproduces Hoffmann et. al.'s performance, and shares as much code as possible with the DNMAR model.

Experimental Setup

We evaluate binary relation extraction using two evaluations. We first evaluate on a sentence-level extraction task using a manually annotated dataset provided by Hoffmann et. al. [80][6]. This dataset consists of sentences paired with human judgments on whether each expresses a specific relation. Secondly, we perform an automatic evaluation which compares propositions extracted from text against held-out data from Freebase.

Results

Sentential Extraction: Figure 5.4 presents precision and recall curves for the sentence-level relation extraction task on the same manually annotated data presented by Hoffmann et. al. [80]. By explicitly modeling the possibility of missing information in both the text and the database we achieve a 17% increase in area under the precision recall curve. Incorporating additional side information in the form of a missing data model, as described in Section 5.6, produces even better performance, yielding a 27% increase over the baseline

[6] `http://raphaelhoffmann.com/mr/`

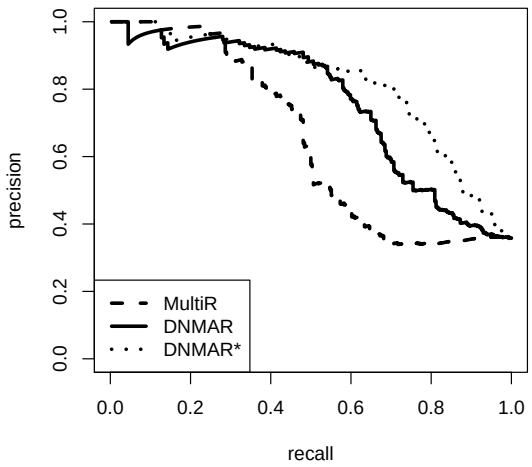

Figure 5.4: Overall precision and Recall at the sentence-level extraction task comparing against human judgments. DNMAR* incorporates side-information as discussed in Section 5.6.

in area under the curve. The differences between each pair of systems is significant with p-value less than 0.05 according to a paired t-test.

Per-relation precision and recall curves are presented in Figure 5.6. For certain relations, for instance */location/us_state/capital*, there simply isn't enough overlap between the information contained in Freebase and facts mentioned in the text to learn anything useful. For these relations, entity pair matches are unlikely to actually express the relation; for instance, in the following sentence from the data:

NHPF , which has its **Louisiana** office in **Baton Rouge** , gets the funds ...

although Baton Rouge is the capital of Louisiana, the */location/us_state/capital* relation is not expressed in this sentence. Another interesting observation which we can make from Figure 5.6, is that the benefit from modeling missing data varies from one relation to another. Some relations, for instance */people/person/place_of_birth*, have relatively good coverage in both Freebase and the text, and therefore we do not see as much gain from modeling missing

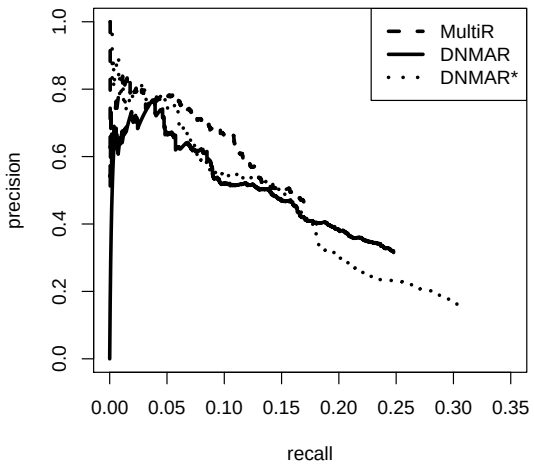

Figure 5.5: Aggregate-level automatic evaluation comparing against held-out data from Freebase. DNMAR* incorporates side-information as discussed in Section 5.6.

data. Other relations, such as */location/location/contains*, and */people/person/place_lived* have poorer coverage making our missing data model very beneficial.

Aggregate Extraction: Following previous work, we evaluate precision and recall against held-out data from Freebase in Figure 5.5. As mentioned by Mintz et. al. [121], this automatic evaluation underestimates precision because many facts correctly extracted from the text are missing in the database and therefore judged as incorrect. Riedel et. al. [149] further argues that this evaluation is biased because frequent entity pairs are more likely to contain facts in Freebase, so systems which rank extractions involving popular entities higher will achieve better performance independently of how accurate their predictions are. Indeed in Figure 5.5 we see that the precision of our system which models missing data is generally lower than the system which assumes no data is missing from Freebase, although we do roughly double the recall. By better modeling missing data we achieve lower precision on this automatic held-out evaluation as the system using hard constraints is explicitly trained to predict facts which occur in Freebase (not those which are mentioned in the text but unlikely to appear in the database).

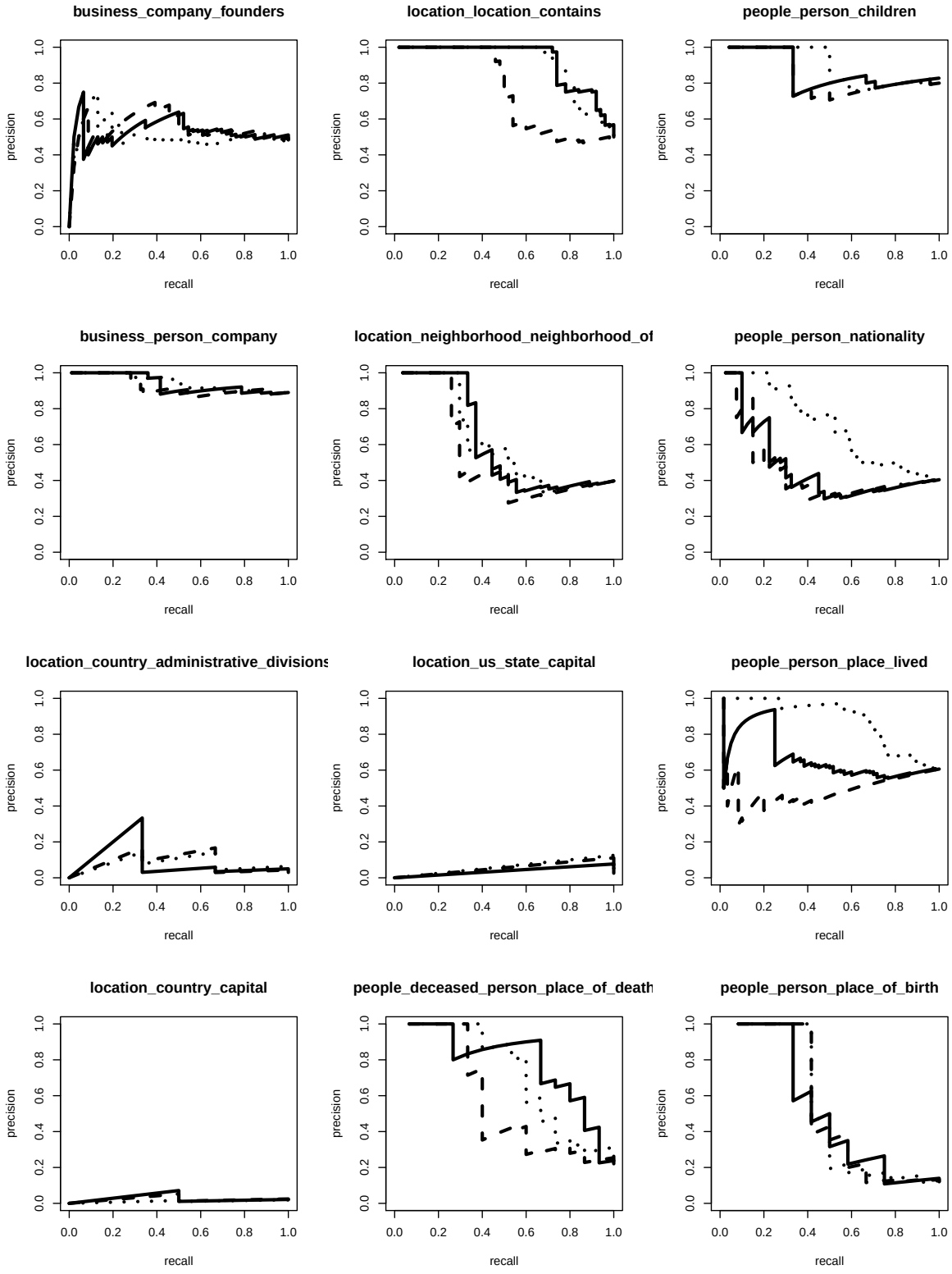

Figure 5.6: Per-relation precision and recall on the sentence-level relation extraction task. The dashed line corresponds to MultiR, DNMAR is the solid line, and DNMAR*, which incorporates side-information, is represented by the dotted line.

5.7.2 Named Entity Categorization

As mentioned previously, the problem of missing data in distant (weak) supervision is a very general issue; thus far we have investigated this problem solely in the context of extracting binary relations using distant supervision. We now turn to the problem of weakly supervised named entity recognition [34; 59; 174].

Experimental Setup

To demonstrate the effect of modeling missing data in the distantly supervised named entity categorization task, we adapt the MultiR and DNMAR models to the Twitter named entity categorization dataset which was presented in Chapter 4. The models described so far are applied unchanged: rather than modeling a set of relations in Freebase between a pair of entities, e_1 and e_2, we now model a set of possible Freebase categories associated with a single entity e. This is a natural extension of distant supervision from binary to unary relations. The unlabeled data and features described in Chapter 4 are used for training the model, and their manually annotated Twitter named entity dataset is used for evaluation.

Results

Precision and recall at weakly supervised named entity categorization comparing MultiR against DNMAR is presented in Figure 5.7. We observe substantial improvement in precision at comparable recall by explicitly modeling the possibility of missing information in the text and database. The missing data model leads to a 107% increase in area under the precision-recall curve (from 0.16 to 0.34), but still falls short of the results presented in Chapter 4, indicating the mixed membership model is more appropriate for highly ambiguous training data, where we only have one entity to anchor the relation, than MultiR.[7]

Note that the more informative missing data models which incorporate side-information for binary relation extraction presented in Section 5.6 are not appropriate for distantly

[7]This makes sense intuitively, for example imagine the situation where entity pairs are allowed to take any relation; in this case the model presented in Chapter 4 reduces to Latent Dirichlet Allocation, a standard topic model, which we would expect to find reasonable categories in the data. It seems unlikely however that MultiR, which is a discriminatively trained model, would successfully identify topics or categories in unlabeled data.

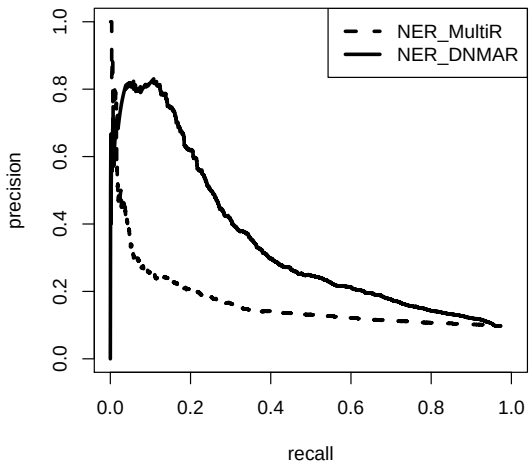

Figure 5.7: Precision and Recall at the named entity categorization task

supervised named entity categorization. For instance, relying on entity frequency as a measure of information missing in the database is not very informative for unary relations (e.g. if an entity is missing, then we don't have any information). We leave the problem of developing more informative missing data models which incorporate side-information for weakly supervised named entity categorization as future work.

5.8 Conclusions

In this chapter we have investigated the problem of missing data in distant supervision; we introduced a joint model of information extraction and missing data which relaxes the hard constraints used in previous work to generate heuristic labels, and provides a natural way to incorporate side information through a missing data model. Efficient inference breaks in the new model, so we presented an approach based on A* search which is guaranteed to find exact solutions, however exact inference is not computationally tractable for large problems. To address the challenge of large problem sizes, we proposed a scalable inference algorithm based on local search, which includes a set of aggregate search operators allowing for long-distance jumps in the solution space to avoid local maxima; this approach was

experimentally demonstrated to find exact solutions in almost every case. Finally we evaluated the performance of our model on the tasks of binary relation extraction and named entity categorization showing large performance gains in each case.

In future work we would like to apply our approach to modeling missing data to additional models, for instance the model of Surdeanu et. al. [171] and the mixed membership model for distant supervision presented in Chapter 4, and also explore new missing data models.

Chapter 6

MIXED MEMBERSHIP MODELS FOR SELECTIONAL PREFERENCE

In Chapters 4 and 5, we investigated a distantly supervised approach to learning that leverages large structured data sources and unlabeled text corpora for training information extraction models. This approach is attractive, since it doesn't rely on manually annotated text corpora, which are relatively small in addition to being expensive and time consuming to produce. In certian situations, however, we don't have access to appropriate structured data sources with sufficient coverage to make this approach feasable. Furthermore, we often don't even have an appropriate set of categories to describe the data in advance.

In this chapter we present an approach to modeling problems in lexical semantics based on latent variable models, using the task of inferring argument types or selectional preferences as a case study. Latent variable models, which leverage large volumes of unlabeled data, are appropriate for problems in lexical semantics because we typically don't know in advance what set of categories are most appropriate. In addition there are many possible argument categories to choose from, and the data is highly unbalanced.

Automatically inferring arugment categories or selectional preferences is a well-known NLP task with broad applicability. We present an approach to selectional preferences based on latent variable models that automatically discovers an appropriate set of argument categories, in addition to the preferences for thousands of relations simultaneously. We evaluate our model of selectional preferences at the task of textual inference, filterng improper applications of inference rules that violate the preferences of the involved relations.

Our model of selectional preferences described in this chapter is adapted to automatically infer significant categories in millions of automatically extracted events from Twitter in Chapter 7.

6.1 Selectional Preferences

Selectional Preferences describe the set of admissible argument values for a relation. For instance, locations are likely to appear in the second argument of the relation *X is headquartered in Y* and companies or organizations in the first. A large, high-quality database of preferences has the potential to improve the performance of a wide range of NLP tasks including semantic role labeling [68], pronoun resolution [10], textual inference [135], word-sense disambiguation [146], and many more. Therefore, much attention has been focused on automatically computing them based on a corpus of relation instances.

Resnik [145] presented the earliest work in this area, describing an information-theoretic approach that inferred selectional preferences based on the WordNet hypernym hierarchy. Recent work [57; 10] has moved away from generalization to known classes, instead utilizing distributional similarity between nouns to generalize beyond observed relation-argument pairs. This avoids problems like WordNet's poor coverage of proper nouns and is shown to improve performance. These methods, however, no longer produce the generalized class for an argument.

In this chapter we describe a novel approach to computing selectional preferences by making use of unsupervised topic models. Our approach is able to combine benefits of both kinds of methods: it retains the generalization and human-interpretability of class-based approaches and is also competitive with the direct methods on predictive tasks.

Unsupervised topic models, such as latent Dirichlet allocation (LDA) [13] and its variants are characterized by a set of hidden topics, which represent the underlying semantic structure of a document collection. For our problem these topics offer an intuitive interpretation – they represent the (latent) set of classes that store the preferences for the different relations. Thus, topic models are a natural fit for modeling our relation data.

In particular, our system, called LDA-SP, uses LinkLDA [58], an extension of LDA that simultaneously models *two* sets of distributions for each topic. These two sets represent the two arguments for the relations. Thus, LDA-SP is able to capture information about the pairs of topics that commonly co-occur. This information is very helpful in guiding inference.

We run LDA-SP to compute preferences on a massive dataset of binary relations $r(a_1, a_2)$ extracted from the Web by TEXTRUNNER [5]. Our experiments demonstrate that LDA-SP significantly outperforms state of the art approaches obtaining an 85% increase in recall at precision 0.9 on the standard pseudo-disambiguation task.

Additionally, because LDA-SP is based on a formal probabilistic model, it has the advantage that it can naturally be applied in many scenarios. For example, we can obtain a better understanding of similar relations (Table 6.1), filter out incorrect inferences based on querying our model (Section 6.4.3), as well as produce a repository of class-based preferences with a little manual effort as demonstrated in Section 6.4.4. In all these cases we obtain high quality results, for example, massively outperforming Pantel et al.'s approach in the textual inference task.[1]

6.2 Related Work

Previous work on selectional preferences can be broken into four categories: class-based approaches [145; 98; 32; 135], similarity based approaches [37; 57], discriminative [10], and generative probabilistic models [156].

6.2.1 Class-based Preferences

First proposed by Resnik [145], class-based methods are the most studied of the four. They make use of a pre-defined set of classes, either manually produced (e.g. WordNet), or automatically generated [137]. For each relation, some measure of the overlap between the classes and observed arguments is used to identify those that best describe the arguments. These techniques produce a human-interpretable output, but often suffer in quality due to an incoherent taxonomy, inability to map arguments to a class (poor lexical coverage), and word sense ambiguity.

---

[1] Our repository of selectional preferences is available at `http://www.cs.washington.edu/research/ldasp`.

6.2.2 Similarity-based Preferences

Because of these limitations researchers have investigated non-class based approaches, which attempt to directly classify a given noun-phrase as plausible/implausible for a relation. Of these, the *similarity based approaches* make use of a distributional similarity measure between arguments and evaluate a heuristic scoring function:

$$S_{\text{rel}}(\text{arg}) = \sum_{\text{arg}' \in \text{Seen(rel)}} \text{sim}(\text{arg}, \text{arg}') \cdot \text{wt}_{\text{rel}}(\text{arg})$$

Erk [57] showed the advantages of this approach over Resnik's information-theoretic class-based method on a pseudo-disambiguation evaluation. These methods obtain better lexical coverage, but are unable to obtain any abstract representation of selectional preferences.

6.2.3 Generative Probabilistic Models

Our solution fits into the general category of *generative probabilistic models*, which model each relation/argument combination as being generated by a latent class variable. Background material on generative models are presented in Chapter 2. These classes are automatically learned from the data. This retains the class-based flavor of the problem, without the knowledge limitations of the explicit class-based approaches. Rooth et al. [156], proposed a model in which each class corresponds to a multinomial over relations and arguments and EM is used to learn the parameters of the model. In contrast, we use a LinkLDA framework in which each relation is associated with a corresponding multinomial distribution over classes, and each argument is drawn from a class-specific distribution over words; LinkLDA captures co-occurrence of classes in the two arguments. Additionally we perform full Bayesian inference using collapsed Gibbs sampling, in which parameters are integrated out [74].

Topic Models

Topic models such as LDA [13] and its variants have recently begun to see use in many NLP applications such as summarization [42], document alignment and segmentation [30],

and inferring class-attribute hierarchies [144]. Our particular model, LinkLDA, has been applied to a few NLP tasks such as simultaneously modeling the words appearing in blog posts and users who will likely respond to them [189], modeling topic-aligned articles in different languages [118], and word sense induction [17].

We highlight two systems, developed independently of our own, which apply LDA-style models to similar tasks. Ó Séaghdha [133] proposes a series of LDA-style models for the task of computing selectional preferences. This work learns selectional preferences between the following grammatical relations: verb-object, noun-noun, and adjective-noun. It also focuses on jointly modeling the generation of both predicate and argument, and evaluation is performed on a set of human-plausibility judgments obtaining impressive results against Keller and Lapata's [83] Web hit-count based system. Van Durme and Gildea [180] proposed applying LDA to general knowledge templates extracted using the KNEXT system [162]. In contrast, our work uses LinkLDA and focuses on modeling multiple arguments of a relation (*e.g.*, the subject and direct object of a verb).

Finally we point out some related work on automatically inferring semantic networks from text [86] using second order Markov Logic Networks. While not applied to the task of computing selectional preferences, this relational clustering system produced similar output.

6.2.4 Discriminative Methods

Bergsma *et. al.* [10] proposed the first *discriminative approach* to selectional preferences. Their insight that pseudo-negative examples could be used as training data allows the application of an SVM classifier, which makes use of many features in addition to the relation-argument co-occurrence frequencies used by other methods. They automatically generated positive and negative examples by selecting arguments having high and low mutual information with the relation. Since it is a discriminative approach it is amenable to feature engineering, but needs to be retrained and tuned for each task. On the other hand, generative models produce complete probability distributions of the data, and hence can be integrated with other systems and tasks in a more principled manner (see Sections 6.4.2 and 6.4.3). Additionally, unlike LDA-SP Bergsma et al.'s system doesn't produce human-

interpretable topics. Finally, we note that LDA-SP and Bergsma's system are potentially complimentary – the output of LDA-SP could be used to generate higher-quality training data for Bergsma, potentially improving their results.

6.3 Topic Models for Selectional Preferences

We present a series of topic models for the task of computing selectional preferences. These models vary in the amount of independence they assume between a_1 and a_2. At one extreme is IndependentLDA, a model which assumes that both a_1 and a_2 are generated completely independently. On the other hand, JointLDA, the model at the other extreme (Figure 6.1) assumes both arguments of a specific extraction are generated based on a single hidden variable z. LinkLDA (Figure 6.2) lies between these two extremes, and as demonstrated in Section 6.4, it is the best model for our relation data.

We are given a set R of binary relations and a corpus $\mathcal{D} = \{r(a_1, a_2)\}$ of extracted instances for these relations. Our task is to compute, for each argument a_i of each relation r, a set of usual argument values (noun phrases) that it takes. For example, for the relation *is headquartered in* the first argument set will include companies like *Microsoft, Intel, General Motors* and second argument will favor locations like *New York, California, Seattle*.

6.3.1 IndependentLDA

We first describe the straightforward application of LDA to modeling our corpus of extracted relations. In this case two separate LDA models are used to model a_1 and a_2 independently.

In the generative model for our data, each relation r has a corresponding multinomial over topics θ_r, drawn from a Dirichlet. For each extraction, a hidden topic z is first picked according to θ_r, and then the observed argument a is chosen according to the multinomial β_z.

Our approach can be described in terms of topic modeling terminology as follows: we treat each relation as a document whose contents consist of a bags of words corresponding to all the noun phrases observed as arguments of the relation in our corpus. Formally, LDA generates each argument in the corpus of relations as presented in Algorithm 9.

Algorithm 7 Generative story for IndependentLDA

for each topic $t = 1 \dots T$ **do**

Generate β_t according to symmetric Dirichlet distribution Dir(η).

end for

for each relation $r = 1 \dots |R|$ **do**

Generate θ_r according to Dirichlet distribution Dir(α).

for each tuple $i = 1 \dots N_r$ **do**

Generate $z_{r,i}$ from Multinomial(θ_r).

Generate the argument $a_{r,i}$ from multinomial $\beta_{z_{r,i}}$.

end for

end for

One weakness of IndependentLDA is that it doesn't jointly model a_1 and a_2 together. Clearly this is undesirable, as information about which topics one of the arguments favors can help inform the topics chosen for the other. For example, class pairs such as (*team*, *game*), (*politician*, *political issue*) form much more plausible selectional preferences than, say, (*team*, *political issue*), (*politician*, *game*).

6.3.2 JointLDA

As a more tightly coupled alternative, we first propose JointLDA, whose graphical model is depicted in Figure 6.1. The key difference in JointLDA (versus LDA) is that instead of one, it maintains *two* sets of topics (latent distributions over words) denoted by β and γ, one for classes of each argument. A topic id k represents a pair of topics, β_k and γ_k, that co-occur in the arguments of extracted relations. Common examples include *(Person, Location), (Politician, Political issue), etc.* The hidden variable $z = k$ indicates that the noun phrase for the first argument was drawn from the multinomial β_k, and that the second argument was drawn from γ_k. The per-relation distribution θ_r is a multinomial over the topic ids and represents the selectional preferences, both for arg1s and arg2s of a relation r.

Although JointLDA has many desirable properties, it has some drawbacks as well. Most notably, in JointLDA topics correspond to pairs of multinomials (β_k, γ_k); this leads to a

situation in which multiple redundant distributions are needed to represent the same underlying semantic class. For example consider the case where we we need to represent the following selectional preferences for our corpus of relations: (*person*, *location*), (*person*, *organization*), and (*person*, *crime*). Because JointLDA requires a separate pair of multinomials for each topic, it is forced to use 3 separate multinomials to represent the class *person*, rather than learning a single distribution representing *person* and choosing 3 different topics for a_2. This results in poor generalization because the data for a single class is divided into multiple topics.

In order to address this problem while maintaining the sharing of influence between a_1 and a_2, we next present LinkLDA, which represents a compromise between IndependentLDA and JointLDA. LinkLDA is more flexible than JointLDA, allowing different topics to be chosen for a_1, and a_2, however still models the generation of topics from the same distribution for a given relation.

Algorithm 8 Generative story for JointLDA

for each topic $t = 1 \dots T$ **do**
  Generate β_t according to symmetric Dirichlet distribution Dir(η_1).
  Generate γ_t according to symmetric Dirichlet distribution Dir(η_2).
end for
for each relation $r = 1 \dots |R|$ **do**
  Generate θ_r according to Dirichlet distribution Dir(α).
  for each tuple $i = 1 \dots N_r$ **do**
    Generate $z_{r,i}$ from Multinomial(θ_r).
    Generate argument 1, $a^1_{r,i}$, from multinomial $\beta_{z_{r,i}}$.
    Generate argument 2, $a^2_{r,i}$, from multinomial $\gamma_{z_{r,i}}$.
  end for
end for

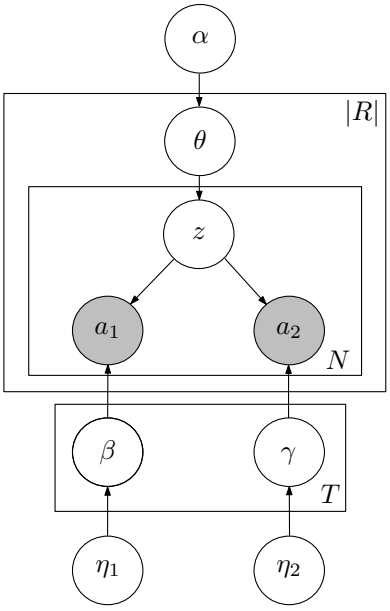

Figure 6.1: JointLDA

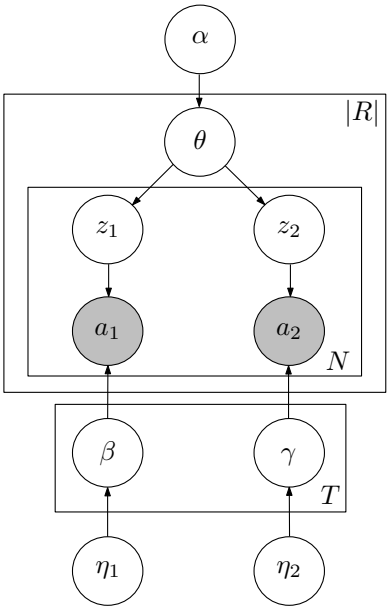

Figure 6.2: LinkLDA

6.3.3 LinkLDA

Figure 6.2 illustrates the LinkLDA model in the plate notation, which is analogous to the model in [58]. In particular note that each a_i is drawn from a different hidden topic z_i, however the z_is are drawn from the same distribution θ_r for a given relation r. To facilitate learning related topic pairs between arguments we employ a sparse prior over the per-relation topic distributions. Because a few topics are likely to be assigned most of the probability mass for a given relation it is more likely (although not necessary) that the same topic number k will be drawn for both arguments.

Algorithm 9 Generative story for LinkLDA

for each topic $t = 1 \dots T$ **do**
  Generate β_t according to symmetric Dirichlet distribution Dir(η_1).
  Generate γ_t according to symmetric Dirichlet distribution Dir(η_2).
end for
for each relation $r = 1 \dots |R|$ **do**
  Generate θ_r according to Dirichlet distribution Dir(α).
  for each tuple $i = 1 \dots N_r$ **do**
    Generate $z^1_{r,i}$ from Multinomial(θ_r).
    Generate $z^2_{r,i}$ from Multinomial(θ_r).
    Generate argument 1, $a^1_{r,i}$, from multinomial $\beta_{z^1_{r,i}}$.
    Generate argument 2, $a^2_{r,i}$, from multinomial $\gamma_{z^2_{r,i}}$.
  end for
end for

When comparing LinkLDA with JointLDA the better model may not seem immediately clear. On the one hand, JointLDA jointly models the generation of both arguments in an extracted tuple. This allows one argument to help disambiguate the other in the case of ambiguous relation strings. LinkLDA, however, is more flexible; rather than requiring both arguments to be generated from one of $|Z|$ possible pairs of multinomials (β_z, γ_z), LinkLDA allows the arguments of a given extraction to be generated from $|Z|^2$ possible pairs. Thus,

instead of imposing a hard constraint that $z_1 = z_2$ (as in JointLDA), LinkLDA simply assigns a higher probability to states in which $z_1 = z_2$, because both hidden variables are drawn from the same (sparse) distribution θ_r. LinkLDA can thus re-use argument classes, choosing different combinations of topics for the arguments if it fits the data better. In Section 6.4 we show experimentally that LinkLDA outperforms JointLDA (and IndependentLDA) by wide margins. We use LDA-SP to refer to LinkLDA in all the experiments below.

6.3.4 Inference

For all the models we use collapsed Gibbs sampling for inference in which each of the hidden variables are sampled sequentially conditioned on a full-assignment to all others, integrating out the parameters [74]. This produces robust parameter estimates, as it allows computation of expectations over the posterior distribution as opposed to estimating maximum likelihood parameters. In addition, the integration allows the use of sparse priors, which are typically more appropriate for natural language data. In all experiments we use hyperparameters $\alpha = \eta_1 = \eta_2 = 0.1$. We generated initial code for our samplers using the Hierarchical Bayes Compiler [43].

6.3.5 Advantages of Topic Models for Selectional Preferences

There are several advantages to using topic models for our task. First, they naturally model the class-based nature of selectional preferences, but don't take a pre-defined set of classes as input. Instead, they compute the classes automatically. This leads to better lexical coverage since the issue of matching a new argument to a known class is side-stepped. Second, the models naturally handle ambiguous arguments, as they are able to assign different topics to the same phrase in different contexts. Inference in these models is also scalable – linear in both the size of the corpus as well as the number of topics. In addition, there are several scalability enhancements such as SparseLDA [191], and an approximation of the Gibbs Sampling procedure can be efficiently parallelized [127]. Finally we note that, once a topic distribution has been learned over a set of training relations, one can efficiently apply inference to unseen relations [191].

6.4 Experiments

We perform three main experiments to assess the quality of the preferences obtained using topic models. The first is a task-independent evaluation using a pseudo-disambiguation experiment (Section 6.4.2), which is a standard way to evaluate the quality of selectional preferences [156; 57; 10]. We use this experiment to compare the various topic models as well as the best model with the known state of the art approaches to selectional preferences. Secondly, we show significant improvements to performance at an end-task of textual inference in Section 6.4.3. Finally, we report on the quality of a large database of Wordnet-based preferences obtained after manually associating our topics with Wordnet classes (Section 6.4.4).

6.4.1 Generalization Corpus

For all experiments we make use of a corpus of $r(a_1, a_2)$ tuples, which was automatically extracted by TextRunner [5] from 500 million Web pages.

To create a *generalization corpus* from this large dataset. We first selected 3,000 relations from the middle of the tail (we used the 2,000-5,000 most frequent ones)[2] and collected all instances. To reduce sparsity, we discarded all tuples containing an NP that occurred fewer than 50 times in the data. This resulted in a vocabulary of about 32,000 noun phrases, and a set of about 2.4 million tuples in our generalization corpus.

We inferred topic-argument and relation-topic multinomials (β, γ, and θ) on the generalization corpus by taking 5 samples at a lag of 50 after a burn in of 750 iterations. Using multiple samples introduces the risk of *topic drift* due to lack of identifiability, however we found this to not be a problem in practice. During development we found that the topics tend to remain stable across multiple samples after sufficient burn in, and multiple samples improved performance. Table 6.1 lists sample topics and high ranked words for each (for both arguments) as well as relations favoring those topics.

[2]Many of the most frequent relations have very weak selectional preferences, and thus provide little signal for inferring meaningful topics. For example, the relations *has* and *is* can take just about any arguments.

Topic t	Arg1	Relations which assign highest probability to t	Arg2
18	The residue - The mixture - The reaction mixture - The solution - the mixture - the reaction mixture - the residue - The reaction - the solution - The filtrate - the reaction - The product - The crude product - The pellet - The organic layer - Thereto - This solution - The resulting solution - Next - The organic phase - The resulting mixture - C.)	was treated with, is treated with, was poured into, was extracted with, was purified by, was diluted with, was filtered through, is disolved in, is washed with	EtOAc - CH2Cl2 - H2O - CH.sub.2Cl.sub.2 - H.sub.2O - water - MeOH - NaHCO3 - Et2O - NHCl - CHCl.sub.3 - NHCl - dropwise - CH2Cl.sub.2 - Celite - Et.sub.2O - Cl.sub.2 - NaOH - AcOEt - CH2C12 - the mixture - saturated NaHCO3 - SiO2 - H2O - N hydrochloric acid - NHCl - preparative HPLC - to0 C
151	the Court - The Court - the Supreme Court - The Supreme Court - this Court - Court - The US Supreme Court - the court - This Court - the US Supreme Court - The court - Supreme Court - Judge - the Court of Appeals - A federal judge	will hear, ruled in, decides, upholds, struck down, overturned, sided with, affirms	the case - the appeal - arguments - a case - evidence - this case - the decision - the law - testimony - the State - an interview - an appeal - cases - the Court - that decision - Congress - a decision - the complaint - oral arguments - a law - the statute
211	President Bush - Bush - The President - Clinton - the President - President Clinton - President George W. Bush - Mr. Bush - The Governor - the Governor - Romney - McCain - The White House - President - Schwarzenegger - Obama	hailed, vetoed, promoted, will deliver, favors, denounced, defended	the bill - a bill - the decision - the war - the idea - the plan - the move - the legislation - legislation - the measure - the proposal - the deal - this bill - a measure - the program - the law - the resolution - efforts - the agreement - gay marriage - the report - abortion
224	Google - Software - the CPU - Clicking - Excel - the user - Firefox - System - The CPU - Internet Explorer - the ability - Program - users - Option - SQL Server - Code - the OS - the BIOS	will display, to store, to load, processes, cannot find, invokes, to search for, to delete	data - files - the data - the file - the URL - information - the files - images - a URL - the information - the IP address - the user - text - the code - a file - the page - IP addresses - PDF files - messages - pages - an IP address

Table 6.1: Example argument lists from the inferred topics. For each topic number t we list the most probable values according to the multinomial distributions for each argument (β_t and γ_t). The middle column reports a few relations whose inferred topic distributions θ_r assign highest probability to t.

6.4.2 Task Independent Evaluation

We first compare the three LDA-based approaches to each other and two state of the art similarity based systems [57] (using mutual information and Jaccard similarity respectively). These similarity measures were shown to outperform the generative model of Rooth et al. [156], as well as class-based methods such as Resnik's. In this pseudo-disambiguation experiment an observed tuple is paired with a pseudo-negative, which has both arguments randomly generated from the whole vocabulary (according to the corpus-wide distribution over arguments). The task is, for each relation-argument pair, to determine whether it is observed, or a random distractor.

Test Set

For this experiment we gathered a primary corpus by first randomly selecting 100 high-frequency relations *not* in the generalization corpus. For each relation we collected all tuples containing arguments in the vocabulary. We held out 500 randomly selected tuples as the test set. For each tuple $r(a_1, a_2)$ in the held-out set, we removed all tuples in the training set containing either of the *rel-arg* pairs, *i.e.*, any tuple matching $r(a_1, *)$ or $r(*, a_2)$. Next we used collapsed Gibbs sampling to infer a distribution over topics, θ_r, for each of the relations in the primary corpus (based solely on tuples in the training set) using the topics from the generalization corpus.

For each of the 500 observed tuples in the test-set we generated a pseudo-negative tuple by randomly sampling two noun phrases from the distribution of NPs in both corpora.

Prediction

Our prediction system needs to determine whether a specific relation-argument pair is admissible according to the selectional preferences or is a random distractor (D). Following previous work, we perform this experiment independently for the two relation-argument pairs (r, a_1) and (r, a_2).

We first compute the probability of observing a_1 for first argument of relation r given that it is not a distractor, $P(a_1|r, \neg D)$, which we approximate by its probability given an

estimate of the parameters inferred by our model, marginalizing over hidden topics t. The analysis for the second argument is similar.

$$\begin{aligned} P(a_1|r, \neg D) \approx P_{LDA}(a_1|r) &= \sum_{t=0}^{T} P(a_1|t)P(t|r) \\ &= \sum_{t=0}^{T} \beta_t(a_1)\theta_r(t) \end{aligned}$$

A simple application of Bayes Rule gives the probability that a particular argument is not a distractor. Here the distractor-related probabilities are independent of r, *i.e.*, $P(D|r) = P(D)$, $P(a_1|D, r) = P(a_1|D)$, *etc.* We estimate $P(a_1|D)$ according to their frequency in the generalization corpus.

$$\begin{aligned} P(\neg D|r, a_1) &= \frac{P(\neg D|r)P(a_1|r, \neg D)}{P(a_1|r)} \\ &\approx \frac{P(\neg D)P_{LDA}(a_1|r)}{P(D)P(a_1|D) + P(\neg D)P_{LDA}(a_1|r)} \end{aligned}$$

Results

Figure 6.3 plots the precision-recall curve for the pseudo-disambiguation experiment comparing the three different topic models. LDA-SP, which uses LinkLDA, substantially outperforms both IndependentLDA and JointLDA.

Next, in figure 6.4, we compare LDA-SP with mutual information and Jaccard similarities using both the generalization and primary corpus for computation of similarities. We find LDA-SP significantly outperforms these methods. Its edge is most noticed at high precisions; it obtains 85% more recall at 0.9 precision compared to mutual information. Overall LDA-SP obtains an 15% increase in the area under precision-recall curve over mutual information. All three systems' AUCs are shown in Table 6.2; LDA-SP's improvements over both Jaccard and mutual information are highly significant with a significance level less than 0.01 using a paired t-test.

In addition to a superior performance in selectional preference evaluation LDA-SP also produces a set of coherent topics, which can be useful in their own right. For instance, one

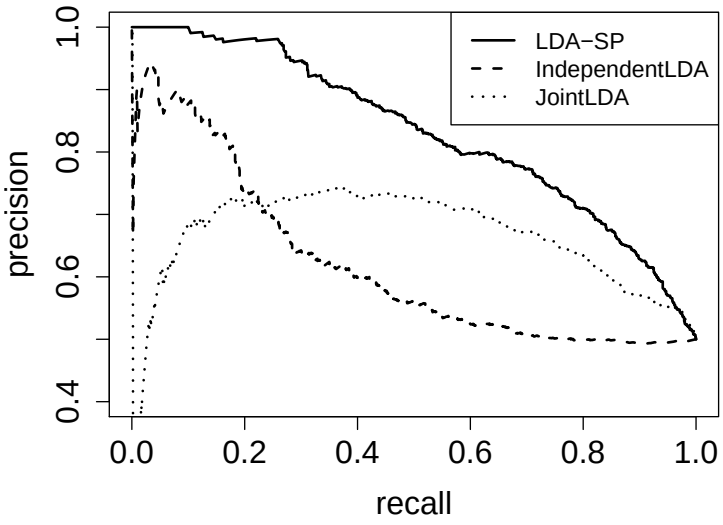

Figure 6.3: Comparison of LDA-based approaches on the pseudo-disambiguation task. LDA-SP (LinkLDA) substantially outperforms the other models.

	LDA-SP	MI-Sim	Jaccard-Sim
AUC	0.833	0.727	0.711

Table 6.2: Area under the precision recall curve. LDA-SP's AUC is significantly higher than both similarity-based methods according to a paired t-test with a significance level below 0.01.

could use them for tasks such as set-expansion [24] or automatic thesaurus induction [59; 91].

6.4.3 End Task Evaluation

We now evaluate LDA-SP's ability to improve performance at an end-task. We choose the task of improving textual entailment by learning selectional preferences for inference rules and filtering inferences that do not respect these. This application of selectional preferences was introduced by Pantel *et. al.* [135]. For now we stick to inference rules of the form $r_1(a_1, a_2) \Rightarrow r_2(a_1, a_2)$, though our ideas are more generally applicable to more complex rules. As an example, the rule (X *defeats* Y) $\Rightarrow$ (X *plays* Y) holds when X and Y are both sports teams, however fails to produce a reasonable inference if X and Y are *Britain* and

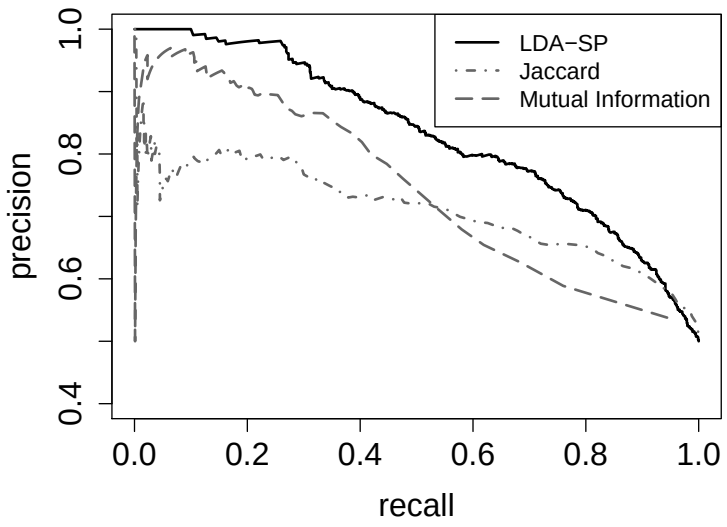

Figure 6.4: Comparison to similarity-based selectional preference systems. LDA-SP obtains 85% higher recall than mutual information at precision 0.9.

Nazi Germany respectively.

Filtering Inferences

In order for an inference to be plausible, both relations must have similar selectional preferences, and further, the arguments must obey the selectional preferences of both the antecedent r_1 and the consequent r_2.[3] Pantel et al. [135] made use of these intuitions by producing a set of class-based selectional preferences for each relation, then filtering out any inferences where the arguments were incompatible with the intersection of these preferences. In contrast, we take a probabilistic approach, evaluating the quality of a specific inference by measuring the probability that the arguments in both the antecedent and the consequent were drawn from the same hidden topic in our model. Note that this probability captures both the requirement that the antecedent and consequent have similar selectional preferences, and that the arguments from a particular instance of the rule's application match their overlap.

[3]Similarity-based and discriminative methods are not applicable to this task as they offer no straightforward way to compare the similarity between selectional preferences of two relations.

We use $z_{r_i,j}$ to denote the topic that generates the j^{th} argument of relation r_i. The probability that the two arguments a_1, a_2 were drawn from the same hidden topic factorizes as follows due to the conditional independences in our model:[4]

$$P(z_{r_1,1} = z_{r_2,1}, z_{r_1,2} = z_{r_2,2}|a_1, a_2) = \\ P(z_{r_1,1} = z_{r_2,1}|a_1)P(z_{r_1,2} = z_{r_2,2}|a_2)$$

To compute each of these factors we simply marginalize over the hidden topics:

$$P(z_{r_1,j} = z_{r_2,j}|a_j) = \sum_{t=1}^{T} P(z_{r_1,j} = t|a_j)P(z_{r_2,j} = t|a_j)$$

where $P(z = t|a)$ can be computed using Bayes rule. For example,

$$\begin{aligned} P(z_{r_1,1} = t|a_1) &= \frac{P(a_1|z_{r_1,1} = t)P(z_{r_1,1} = t)}{P(a_1)} \\ &= \frac{\beta_t(a_1)\theta_{r_1}(t)}{P(a_1)} \end{aligned}$$

Experimental Conditions

In order to evaluate LDA-SP's ability to filter inferences based on selectional preferences we need a set of inference rules between the relations in our corpus. We therefore mapped the DIRT Inference rules [104], (which consist of pairs of dependency paths) to TEXTRUNNER relations as follows. We first gathered all instances in the generalization corpus, and for each $r(a_1, a_2)$ created a corresponding simple sentence by concatenating the arguments with the relation string between them. Each such simple sentence was parsed using Minipar [103]. From the parses we extracted all dependency paths between nouns that contain only words present in the TEXTRUNNER relation string. These dependency paths were then matched against each pair in the DIRT database, and all pairs of associated relations were collected producing about 26,000 inference rules.

[4]Note that all probabilities are conditioned on an estimate of the parameters θ, β, γ from our model, which are omitted for compactness.

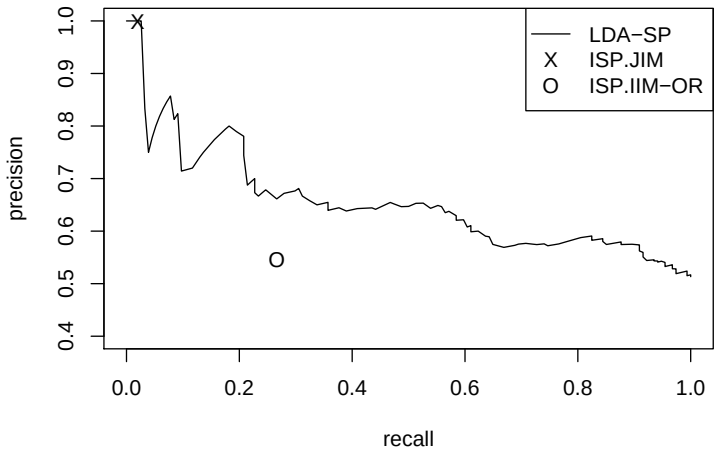

Figure 6.5: Precision and recall on the inference filtering task.

Following Pantel et al. [135] we randomly sampled 100 inference rules. We then automatically filtered out any rules which contained a negation, or for which the antecedent and consequent contained a pair of antonyms found in WordNet (this left us with 85 rules). For each rule we collected 10 random instances of the antecedent, and generated the consequent. We randomly sampled 300 of these inferences to hand-label.

Results

In figure 6.5 we compare the precision and recall of LDA-SP against the top two performing systems described by Pantel et al. (ISP.IIM-$\vee$ and ISP.JIM, both using the CBC clusters [137]). We find that LDA-SP achieves both higher precision and recall than ISP.IIM-$\vee$ on the inference filtering evaluation. It is also able to achieve the high-precision point of ISP.JIM and can trade precision to get a much larger recall.

In addition we demonstrate LDA-SP's ability to rank inference rules by measuring the Kullback Leibler Divergence[5] between the topic-distributions of the antecedent and consequent, θ_{r_1} and θ_{r_2} respectively. Table 6.3 shows the top 10 and bottom 10 rules out of the 26,000 ranked by KL Divergence after automatically filtering antonyms (using WordNet) and negations. For slight variations in rules (*e.g.*, symmetric pairs) we mention only one

[5]KL-Divergence is an information-theoretic measure of the similarity between two probability distributions, and defined as follows: $KL(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$.

example to show more variety.

6.4.4 A Repository of Class-Based Preferences

Finally we explore LDA-SP's ability to produce a repository of human interpretable class-based selectional preferences. As an example, for the relation *was born in*, we would like to infer that the plausible arguments include (*person*, *location*) and (*person*, *date*).

Since we already have a set of topics, our task reduces to mapping the inferred topics to an equivalent class in a taxonomy (*e.g.*, WordNet). We experimented with automatic methods such as Resnik's, but found them to have all the same problems as directly applying these approaches to the SP task.[6] Guided by the fact that we have a relatively small number of topics (600 total, 300 for each argument) we simply chose to label them manually. By labeling this small number of topics we can infer class-based preferences for an arbitrary number of relations.

In particular, we applied a semi-automatic scheme to map topics to WordNet. We first applied Resnik's approach to automatically shortlist a few candidate WordNet classes for each topic. We then manually picked the best class from the shortlist that best represented the 20 top arguments for a topic (similar to Table 6.1). We marked all incoherent topics with a special symbol $\emptyset$. This process took one of the authors about 4 hours to complete.

To evaluate how well our topic-class associations carry over to unseen relations we used the same random sample of 100 relations from the pseudo-disambiguation experiment.[7] For each argument of each relation we picked the top two topics according to frequency in the 5 Gibbs samples. We then discarded any topics which were labeled with $\emptyset$; this resulted in a set of 236 predictions. A few examples are displayed in table 6.4.

We evaluated these classes and found the accuracy to be around 0.88. We contrast this with Pantel's repository,[8] the only other released database of selectional preferences to our

[6] Perhaps recent work on automatic coherence ranking [129] and labeling [117] could produce better results.

[7] Recall that these 100 were not part of the original 3,000 in the generalization corpus, and are, therefore, representative of new "unseen" relations.

[8] `http://demo.patrickpantel.com/Content/LexSem/paraphrase.htm`

Top 10 Inference Rules Ranked by LDA-SP		
antecedent	consequent	KL-div
will begin at	will start at	0.014999
shall review	shall determine	0.129434
may increase	may reduce	0.214841
walk from	walk to	0.219471
consume	absorb	0.240730
shall keep	shall maintain	0.264299
shall pay to	will notify	0.290555
may apply for	may obtain	0.313916
copy	download	0.316502
should pay	must pay	0.371544
Bottom 10 Inference Rules Ranked by LDA-SP		
antecedent	consequent	KL-div
lose to	shall take	10.011848
should play	could do	10.028904
could play	get in	10.048857
will start at	move to	10.060994
shall keep	will spend	10.105493
should play	get in	10.131299
shall pay to	leave for	10.131364
shall keep	return to	10.149797
shall keep	could do	10.178032
shall maintain	have spent	10.221618

Table 6.3: Top 10 and Bottom 10 ranked inference rules ranked by LDA-SPafter automatically filtering out negations and antonyms (using WordNet).

arg1 class	relation	arg2 class
politician#1	was running for	leader#1
people#1	will love	show#3
organization#1	has responded to	accusation#2
administrative_unit#1	has appointed	administrator#3

Table 6.4: Class-based Selectional Preferences.

knowledge. We evaluated the same 100 relations from his website and tagged the top 2 classes for each argument and evaluated the accuracy to be roughly 0.55.

We emphasize that tagging a pair of class-based preferences is a highly subjective task, so these results should be treated as preliminary. Still, these early results are promising. We wish to undertake a larger scale study soon.

6.5 Limitations

LinkLDA captures commonly co-occurring argument-class-pairs, which we found to be quite useful for modeling selectional preferences. One limitation of this model, however is its inability to capture correlations between classes in the same argument. For instance, relations which appear with *person* as the first argument, also often apply to *organization*s. This limitation is due to the use of the Dirichlet distribution as a prior over per-relation topic distributions. One potential avenue for addressing this limitation would be to replace the Dirichlet with a logistic normal distribution [12], which is capable of modeling correlations between topics. This approach has the potential to better model the data, although would also complicate inference as the logistic normal is not conjugate to the multinomial distribution as is the Dirichlet. It is therefore left as an open empirical question which approach will work best for modeling selectional preferences.

6.6 Conclusions and Future Work

This chapter presented an application of topic modeling to the problem of automatically computing selectional preferences. Our method, LDA-SP, learns a distribution over top-

ics for each relation while simultaneously grouping related words into these topics. This approach is capable of producing human interpretable classes, however, avoids the drawbacks of traditional class-based approaches (poor lexical coverage and ambiguity). LDA-SP achieves state-of-the-art performance on predictive tasks such as pseudo-disambiguation, and filtering incorrect inferences.

Our repository of selectional preferences for 10,000 relations is available at `http://www.cs.washington.edu/research/ldasp`.

Our approach automatically induces an appropriate set of argument types in addition to the preferences for each relation. This is very advantageous in practice since neither the set of types, nor the preferences are obvious in advance for large heterogeneous text corpora such as Twitter and the web, making supervised approaches based on manually annotated data infesable for this task. This same approach is adapted to the task of unsupervised event type categorization in Twitter in Chapter 7.

Chapter 7

EXTRACTING AN OPEN-DOMAIN CALENDAR OF EVENTS FROM MICROBLOG TEXT

We now have access to syntactic annotation tools adapted to noisy informal text which were developed in Chpater 3. We also have a set probabilistic models for open-domain information extraction in large heterogenous text corpora developed in Chapters 4, 5, 6. What new kinds of data analysis applications could this set of tools and techniques for processing large, noisy, heterogeneous text corpora enable? In this chapter we provide one possible answer to this question by demonstrating how to extract an open-domain calendar of popular events occuring in the near future from Twitter.

To do this, we leverage the named entity recognizer presented in Chapter 3. We also develop an event extractor following similar techniques: annotating a corpus of tweets with event phrases, which are then used to train in-domain sequence models. In addition we leverage the in-domain part of speech tags from Chapter 3 as input to a system which is able to automatically resolve temporal expressions to determine when the events take place. Because Twitter users can talk about pretty much anything, and it is apriori unclear which set of event categories are most important to capture in this data, we adapt the model of selectional preferences described in Chapter 6 to automatically induce an appropriate set of categories for popular events discussed on Twitter. This approach has the advantage that it is agnostic to the correct set of categories in the data, and also provides a model based on large volumes of unlabeled data which can be queried to categorize entities in context, outperforming a supervised baseline.

7.1 Open Domain Event Extraction from Twitter

Previous work in event extraction [75; 3; 188; 66; 142; 46; 26] has focused largely on news articles, as historically this genre of text has been the best source of information on current events. In the meantime, social networking sites such as Facebook and Twitter have become

Entity	Event Phrase	Date	Type
Steve Jobs	died	10/6/11	Death
iPhone	announcement	10/4/11	ProductLaunch
GOP	debate	9/7/11	PoliticalEvent
Amanda Knox	verdict	10/3/11	Trial

Table 7.1: Examples of events extracted by TwiCal.

an important complementary source of such information. While status messages contain a wealth of useful information, they are very disorganized motivating the need for automatic extraction, aggregation and categorization. Although there has been much interest in tracking trends or memes in social media [96; 105], little work has addressed the challenges arising from extracting structured representations of events from short or informal texts.

Extracting useful structured representations of events from this disorganized corpus of noisy text is a challenging problem. On the other hand, individual tweets are short and self-contained and are therefore not composed of complex discourse structure as is the case for texts containing narratives. In this chapter we demonstrate that **open-domain event extraction** from Twitter is indeed feasible, for example our highest-confidence extracted future events are 90% accurate as demonstrated in Section 7.7.

Twitter has several characteristics which present unique *challenges* and *opportunities* for the task of open-domain event extraction.

Challenges: Twitter users frequently mention mundane events in their daily lives (such as what they ate for lunch) which are only of interest to their immediate social network. In contrast, if an event is mentioned in newswire text, it is safe to assume it is of general importance. Individual tweets are also very terse, often lacking sufficient context to categorize them into topics of interest (e.g. Sports, Politics, ProductRelease etc...). Further because Twitter users can talk about whatever they choose, it is unclear in advance which set of event types are appropriate. Finally, tweets are written in an informal style causing NLP tools designed for edited texts to perform extremely poorly.

Opportunities: The short and self-contained nature of tweets means they have very simple discourse and pragmatic structure, issues which still challenge state-of-the-art NLP systems. For example in newswire, complex reasoning about relations between events (e.g. *before* and *after*) is often required to accurately relate events to temporal expressions [110; 28]. The volume of Tweets is also much larger than the volume of news articles, so redundancy of information can be exploited more easily.

To address Twitter's noisy style, we follow recent work on NLP in noisy text [152; 108; 69], annotating a corpus of Tweets with events, which is then used as training data for sequence-labeling models to identify event mentions in millions of messages.

Because of the terse, sometimes mundane, but highly redundant nature of tweets, we were motivated to focus on extracting an aggregate representation of events which provides additional context for tasks such as event categorization, and also filters out mundane events by exploiting redundancy of information. We propose identifying important events as those whose mentions are strongly associated with references to a unique date as opposed to dates which are evenly distributed across the calendar.

Twitter users discuss a wide variety of topics, making it unclear in advance what set of event types are appropriate for categorization. To address the diversity of events discussed on Twitter, we introduce a novel approach to discovering important event types and categorizing aggregate events within a new domain.

Supervised or semi-supervised approaches to event categorization would require first designing annotation guidelines (including selecting an appropriate set of types to annotate), then annotating a large corpus of events found in Twitter. This approach has several drawbacks, as it is apriori unclear what set of types should be annotated; a large amount of effort would be required to manually annotate a corpus of events while simultaneously refining annotation standards.

We propose an approach to open-domain event categorization based on latent variable models that uncovers an appropriate set of types which match the data. The automatically discovered types are subsequently inspected to filter out any which are incoherent and the

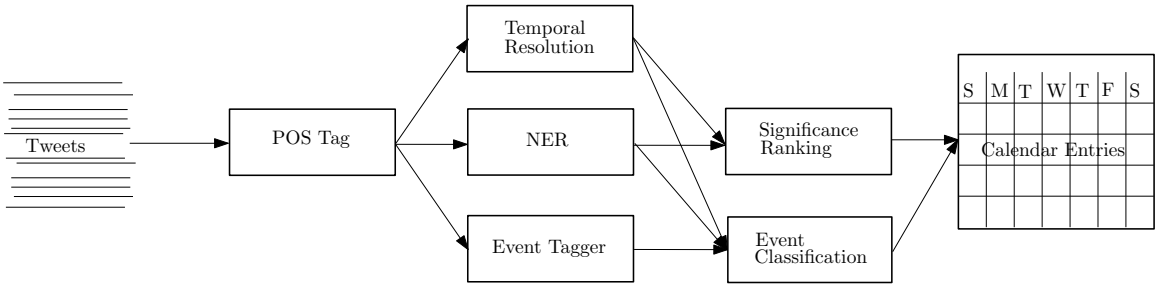

Figure 7.1: Processing pipeline for extracting events from Twitter.

rest are annotated with informative labels;[1] examples of types discovered using our approach are listed in figure 7.3. The resulting set of types are then applied to categorize hundreds of millions of extracted events without the use of any manually annotated examples. By leveraging large quantities of unlabeled data, our approach results in a 14% improvement in F_1 score over a supervised baseline which uses the same set of types.

7.2 System Overview

TwiCal extracts a 4-tuple representation of events which includes a named entity, event phrase, calendar date, and event type (see Table 7.1). This representation was chosen to closely match the way important events are typically mentioned in Twitter.

While including location information as part of our representation could be quite useful (for example enabling querying for events taking place in a specific city), we do not address the location resolution problem as part of this work. Accurate and complete extraction of location information for these events would likely require linking entities mentioned in the text of individual tweets with databases of geo-located entities [38]. Also note that location is not important for many types of prominent events discussed on Twitter (for example product releases, movie premieres, celebrity deaths and so on).

An overview of the various components of our system for extracting events from Twitter is presented in Figure 7.1. Given a raw stream of tweets, our system extracts named

[1]This annotation and filtering takes minimal effort. One of the authors spent roughly 30 minutes inspecting and annotating the automatically discovered event types.

entities in association with event phrases and unambiguous dates which are involved in significant events. First the tweets are POS tagged, then named entities and event phrases are extracted, temporal expressions resolved, and the extracted events are categorized into types. Finally we measure the strength of association between each named entity and date based on the number of tweets they co-occur in, in order to determine whether an event is significant.

NLP tools, such as named entity segmenters and part of speech taggers which were designed to process edited texts (e.g. news articles) perform very poorly when applied to Twitter text due to its noisy and unique style. To address these issues, we utilize a named entity tagger and part of speech tagger trained on in-domain Twitter data presented in Chapter 3. We also develop an event tagger trained on in-domain annotated data.

7.3 Extracting Event Mentions

In order to extract event mentions from Twitter's noisy text, we first annotate a corpus of tweets, which is then used to train sequence models to extract events. While we apply an established approach to sequence-labeling tasks in noisy text [152; 108; 69], this is the first work to extract event-referring phrases in Twitter.

Event phrases can consist of many different parts of speech as illustrated in the following examples:

- **Verbs:** Apple to *Announce* iPhone 5 on October 4th?! YES!
- **Nouns:** iPhone 5 *announcement* coming Oct 4th
- **Adjectives:** WOOOHOO *NEW* IPHONE TODAY! CAN'T WAIT!

These phrases provide important context, for example extracting the entity, **Steve Jobs** and the event phrase **died** in connection with October 5th, is much more informative than simply extracting **Steve Jobs**. In addition, event mentions are helpful in upstream tasks such as categorizing events into types, as described in Section 7.6.

In order to build a tagger for recognizing events, we annotated 1,000 tweets (19,484 tokens) with event phrases, following annotation guidelines similar to those developed for

	precision	recall	F1
TwiCal-Event	0.56	0.74	0.64
No POS	0.48	0.70	0.57
Timebank	0.24	0.11	0.15

Table 7.2: Precision and recall at event phrase extraction. All results are reported using 4-fold cross validation over the 1,000 manually annotated tweets (about 19K tokens). We compare against a system which doesn't make use of features generated based on our Twitter trained POS Tagger, in addition to a system trained on the Timebank corpus which uses the same set of features.

the Event tags in Timebank [142]. We treat the problem of recognizing event triggers as a sequence labeling task, using Conditional Random Fields for learning and inference [94]. Linear Chain CRFs model dependencies between the predicted labels of adjacent words, which are beneficial for extracting multi-word event phrases. We use contextual, dictionary, and orthographic features, and also include features based on our Twitter-tuned POS tagger [152], and dictionaries of event terms gathered from WordNet by Sauri et al. [161].

The precision and recall at segmenting event phrases are reported in Table 7.2. Our classifier, TwiCal-Event, obtains an F-score of 0.64. To demonstrate the need for in-domain training data, we compare against a baseline of training our system on the Timebank corpus.

7.4 Extracting and Resolving Temporal Expressions

In addition to extracting events and related named entities, we also need to extract when they occur. In general there are many different ways users can refer to the same calendar date, for example "next Friday", "August 12th", "tomorrow" or "yesterday" could all refer to the same day, depending on when the tweet was written. To resolve temporal expressions we make use of TempEx [111], which takes as input a reference date, some text, and parts of speech (from our Twitter-trained POS tagger) and marks temporal expressions with unambiguous calendar references. Although this mostly rule-based system was designed for use on newswire text, we find its precision on Tweets (94% - estimated over as sample of 268

extractions) is sufficiently high to be useful for our purposes. TempEx's high precision on Tweets can be explained by the fact that some temporal expressions are relatively unambiguous. Although there appears to be room for improving the recall of temporal extraction on Twitter by handling noisy temporal expressions (for example see Chapter 3 for a list of over 50 spelling variations on the word "tomorrow"), we leave adapting temporal extraction to Twitter as future work (see Chapter 8).

7.5 Ranking Events

Simply using frequency to determine which events are significant is insufficient, because many tweets refer to common events in user's daily lives. As an example, users often mention what they are eating for lunch, therefore entities such as *McDonalds* occur relatively frequently in association with references to most calendar days. Important events can be distinguished as those which have strong association with a unique date as opposed to being spread evenly across days on the calendar. To extract significant events of general interest from Twitter, we thus need some way to measure the strength of association between an entity and a date.

In order to measure the association strength between an entity and a specific date, we utilize the G^2 log likelihood ratio statistic. G^2 has been argued to be more appropriate for text analysis tasks than χ^2 [52]. Although Fisher's Exact test would produce more accurate p-values [122], given the amount of data with which we are working (sample size greater than 10^{11}), it proves difficult to compute Fisher's Exact Test Statistic, which results in floating point overflow even when using 64-bit operations. The G^2 test works sufficiently well in our setting, however, as computing association between entities and dates produces less sparse contingency tables than when working with pairs of entities (or words).

The G^2 test is based on the likelihood ratio between a model in which the entity is conditioned on the date, and a model of independence between entities and date references. For a given entity e and date d this statistic can be computed as follows:

$$G^2 = \sum_{x \in \{e, \neg e\}, y \in \{d, \neg d\}} O_{x,y} \times \ln\left(\frac{O_{x,y}}{E_{x,y}}\right)$$

Where $O_{e,d}$ is the observed fraction of tweets containing both e and d, $O_{e,\neg d}$ is the observed

fraction of tweets containing e, but not d, and so on. Similarly $E_{e,d}$ is the expected fraction of tweets containing both e and d assuming a model of independence.

7.6 Categorizing Events Extracted from Microblogs

To categorize the extracted events into types we propose an approach based on latent variable models which infers an appropriate set of event types to match our data, and also classifies events into types by leveraging large amounts of unlabeled data.

Supervised or semi-supervised classification of event categories is problematic for a number of reasons. First, it is *a priori* unclear which categories are appropriate for Twitter. Secondly, a large amount of manual effort is required to annotate tweets with event types. Third, the set of important categories (and entities) is likely to shift over time, or within a focused user demographic. Finally many important categories are relatively infrequent, so even a large annotated dataset may contain just a few examples of these categories, making classification difficult.

For these reasons we were motivated to investigate unsupervised approaches that will automatically induce event types which match the data. We adopt an approach based on latent variable models inspired by the model of selectional preferences presented in Chapter 6 [154; 163; 88; 180; 155], and recent work on unsupervised information extraction [7; 192; 26].

Each event indicator phrase in our data, e, is modeled as a mixture of types. For example the event phrase "cheered" might appear as part of either a PoliticalEvent, or a SportsEvent. Each type corresponds to a distribution over named entities n involved in specific instances of the type, in addition to a distribution over dates d on which events of the type occur. Including calendar dates in our model has the effect of encouraging (though not requiring) events which occur on the same date to be assigned the same type. This is helpful in guiding inference, because distinct references to the same event should also have the same type.

The generative story for our data is based on LinkLDA [58], and is presented as Algorithm 10. This approach has the advantage that information about an event phrase's type distribution is shared across its mentions, while ambiguity is also naturally preserved.

Sports	7.45%	Conflict	0.69%
Party	3.66%	Prize	0.68%
TV	3.04%	Legal	0.67%
Politics	2.92%	Death	0.66%
Celebrity	2.38%	Sale	0.66%
Music	1.96%	VideoGameRelease	0.65%
Movie	1.92%	Graduation	0.63%
Food	1.87%	Racing	0.61%
Concert	1.53%	Fundraiser/Drive	0.60%
Performance	1.42%	Exhibit	0.60%
Fitness	1.11%	Celebration	0.60%
Interview	1.01%	Books	0.58%
ProductRelease	0.95%	Film	0.50%
Meeting	0.88%	Opening/Closing	0.49%
Fashion	0.87%	Wedding	0.46%
Finance	0.85%	Holiday	0.45%
School	0.85%	Medical	0.42%
AlbumRelease	0.78%	Wrestling	0.41%
Religion	0.71%	OTHER	53.45%

Figure 7.2: Complete list of automatically discovered event types with percentage of data covered. Interpretable types representing significant events cover roughly half of the data.

Label	Top 5 Event Phrases	Top 5 Entities
Sports	tailgate - scrimmage - tailgating - homecoming - regular season	espn - ncaa - tigers - eagles - varsity
Concert	concert - presale - performs - concerts - tickets	taylor swift - toronto - britney spears - rihanna - rock
Perform	matinee - musical - priscilla - seeing - wicked	shrek - les mis - lee evans - wicked - broadway
TV	new season - season finale - finished season - episodes - new episode	jersey shore - true blood - glee - dvr - hbo
Movie	watch love - dialogue theme - inception - hall pass - movie	netflix - black swan - insidious - tron - scott pilgrim
Sports	inning - innings - pitched - homered - homer	mlb - red sox - yankees - twins - dl
Politics	presidential debate - osama - presidential candidate - republican debate - debate performance	obama - president obama - gop - cnn - america
TV	network news broadcast - airing - primetime drama - channel - stream	nbc - espn - abc - fox - mtv
Product	unveils - unveiled - announces - launches - wraps off	apple - google - microsoft - uk - sony
Meeting	shows trading - hall - mtg - zoning - briefing	town hall - city hall - club - commerce - white house
Finance	stocks - tumbled - trading report - opened higher - tumbles	reuters - new york - u.s. - china - euro
School	maths - english test - exam - revise - physics	english - maths - german - bio - twitter
Album	in stores - album out - debut album - drops on - hits stores	itunes - ep - uk - amazon - cd
TV	voted off - idol - scotty - idol season - dividend-paying	lady gaga - american idol - america - beyonce - glee
Religion	sermon - preaching - preached - worship - preach	church - jesus - pastor - faith - god
Conflict	declared war - war - shelling - opened fire - wounded	libya - afghanistan - #syria - syria - nato
Politics	senate - legislation - repeal - budget - election	senate - house - congress - obama - gop
Prize	winners - lotto results - enter - winner - contest	ipad - award - facebook - good luck - winners
Legal	bail plea - murder trial - sentenced - plea - convicted	casey anthony - court - india - new delhi - supreme cou
Movie	film festival - screening - starring - film - gosling	hollywood - nyc - la - los angeles - new york
Death	live forever - passed away - sad news - condolences - burried	michael jackson - afghanistan - john lennon - young peace
Sale	add into - 50% off - up - shipping - save up	groupon - early bird - facebook - @etsy - etsy
Drive	donate - tornado relief - disaster relief - donated - raise money	japan - red cross - joplin - june - africa

Figure 7.3: Example event types discovered by our model. For each type t, we list the top 5 entities which have highest probability given t, and the 5 event phrases which assign highest probability to t.

Algorithm 10 Generative story for our data involving event types as hidden variables. Bayesian Inference techniques are applied to invert the generative process and infer an appropriate set of types to describe the observed events.

for each event type $t = 1 \dots T$ **do**

Generate β_t^n according to symmetric Dirichlet distribution Dir(η_n).

Generate β_t^d according to symmetric Dirichlet distribution Dir(η_d).

end for

for each unique event phrase $e = 1 \dots |E|$ **do**

Generate θ_e according to Dirichlet distribution Dir(α).

for each entity which co-occurs with e, $i = 1 \dots N_e$ **do**

Generate $z_{e,i}^n$ from Multinomial(θ_e).

Generate the entity $n_{e,i}$ from Multinomial($\beta_{z_{e,i}^n}$).

end for

for each date which co-occurs with e, $i = 1 \dots N_d$ **do**

Generate $z_{e,i}^d$ from Multinomial(θ_e).

Generate the date $d_{e,i}$ from Multinomial($\beta_{z_{d,i}^n}$).

end for

end for

For inference we use collapsed Gibbs Sampling [74] where each hidden variable, z_i, is sampled in turn, and parameters are integrated out. Example types are displayed in Figure 7.3. To estimate the distribution over types for a given event, a sample of the corresponding hidden variables is taken from the Gibbs Markov chain after sufficient burn in. Prediction for new data is performed using a streaming approach to inference [193].

7.6.1 Evaluation

To evaluate the ability of our model to classify significant events, we gathered 65 million extracted events of the form listed in Figure 7.1 (not including the type). We then ran Gibbs Sampling with 100 types for 1,000 iterations of burn-in, keeping the hidden variable assignments found in the last sample.[2]

One of the authors manually inspected the resulting types and assigned them labels such as SPORTS, POLITICS, MUSICRELEASE and so on, based on their distribution over entities, and the event words which assign highest probability to that type. Out of the 100 types, we found 52 to correspond to coherent event types which referred to significant events;[3] the other types were either incoherent, or covered types of events which are not of general interest, for example there was a cluster of phrases such as *applied*, *call*, *contact*, *job interview*, etc... which correspond to users discussing events related to searching for a job. Such event types which do not correspond to significant events of general interest were simply marked as *OTHER*. A complete list of labels used to annotate the automatically discovered event types along with the coverage of each type is listed in figure 7.2. Note that this assignment of labels to types only needs to be done once and produces a labeling for an arbitrarily large number of event instances. Additionally the same set of types can easily be used to classify new event instances using streaming inference techniques [193]. One interesting direction for future work is automatic labeling and coherence evaluation of automatically discovered event types analogous to recent work on topic models [130; 95].

---

[2]To scale up to larger datasets, we performed inference in parallel on 40 cores using an approximation to the Gibbs Sampling procedure analogous to that presented by Newmann et. al. [128].

[3]After labeling some types were combined resulting in 37 distinct labels.

	Precision	Recall	F_1
TwiCal-Classify	0.85	0.55	0.67
Supervised Baseline	0.61	0.57	0.59

Table 7.3: Precision and recall of event type categorization at the point of maximum F_1 score.

In order to evaluate the ability of our model to classify aggregate events, we grouped together all (entity,date) pairs which occur 20 or more times the data, then annotated the 500 with highest association (see Section 7.5) using the event types discovered by our model.

To help demonstrate the benefits of leveraging large quantities of unlabeled data for event classification, we compare against a supervised Maximum Entropy baseline which makes use of the 500 annotated events using 10-fold cross validation. For features, we treat the set of event phrases that co-occur with each (entity, date) pair as a bag-of-words, and also include the associated entity. Because many event categories are infrequent, there are often few or no training examples for a category, leading to low performance.

Figure 7.4 compares the performance of our unsupervised approach to the supervised baseline, via a precision-recall curve obtained by varying the threshold on the probability of the most likely type. In addition table 7.3 compares precision and recall at the point of maximum F-score. Our unsupervised approach to event categorization achieves a 14% increase in maximum F_1 score over the supervised baseline.

Figure 7.5 plots the maximum F_1 score as the amount of training data used by the baseline is varied. It seems likely that with more data, performance will reach that of our approach which does not make use of any annotated events, however our approach both automatically discovers an appropriate set of event types and provides an initial classifier with minimal effort, making it useful as a first step in situations where annotated data is not immediately available.

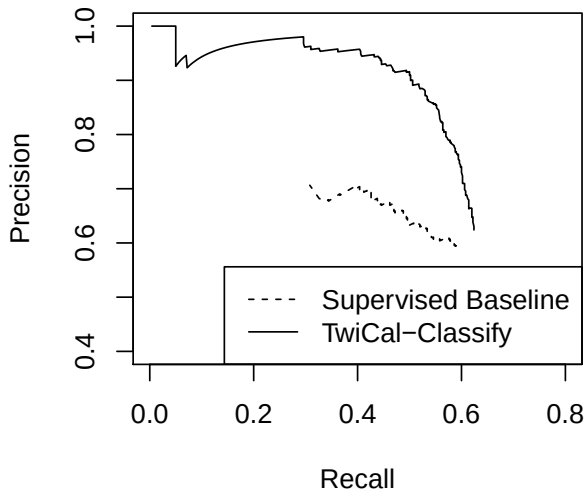

Figure 7.4: Precision and recall predicting event types.

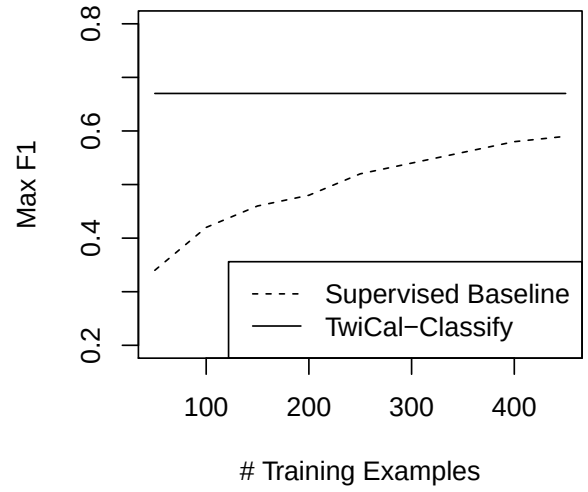

Figure 7.5: Maximum F_1 score of the supervised baseline as the amount of training data is varied.

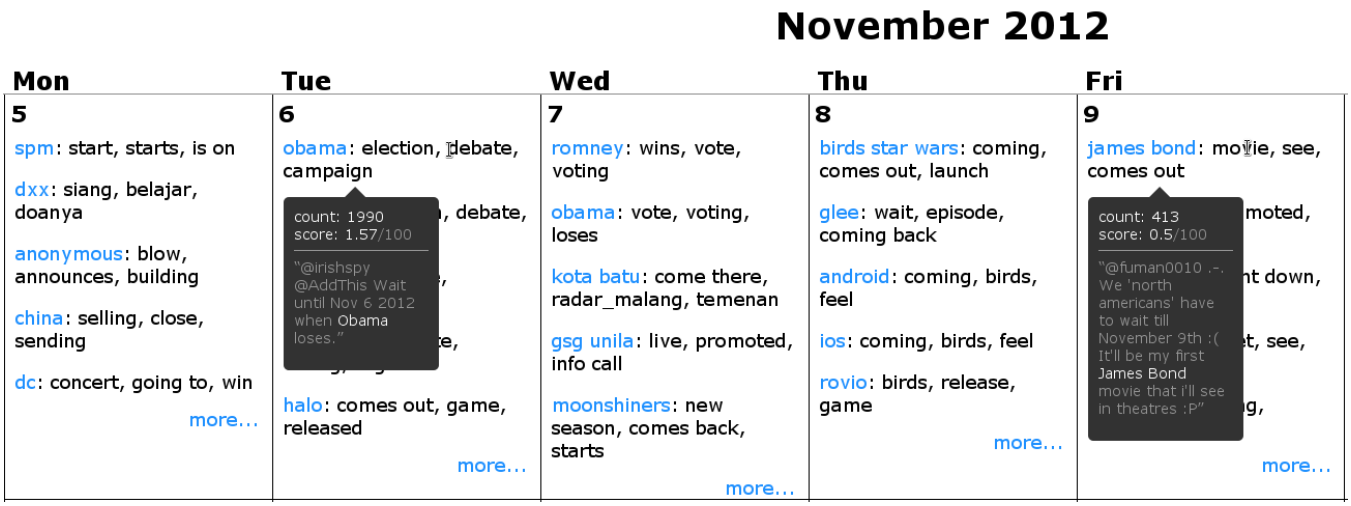

Figure 7.6: Screenshot of popular events extracted from Twitter taken on October 26, 2012. Examples include the US presidential elections on November 6, and the release of the new James Bond movie on November 9.

7.6.2 Measuring Similarity Between Event Phrases

Because our approach is based on generative probabilistic models (as opposed to a discriminatively trained approach), the learned model may be useful for additional purposes beyond categorizing events into types. As a demonstration of our learned model's generality, we investigated its ability to generate paraphrases of extracted event phrases in Tweets similar to those found in the DIRT database [104]. These paraphrases could be useful for a variety of purposes, for example, answering natural language questions about events discussed in Tweets, or summarization. To evaluate our model's ability to generate paraphrases, we took all event phrases which occurred 1,000 or more times in our data, and measured KL divergence[4] between each pair's distribution over event types. The 10 event phrases which have lowest KL-divergence to 10 randomly sampled phrases are listed in Table 7.4.

Input Phrase	Top 10 Most Similar Phrases by KL Divergence
closed	canceled, vacation, prepare, wind, cancelled, alert, delayed, delayed, schedule, holiday
sell	published, sold, offer, progress, convering, discovered, issue, credit, jump, delivered
watch	watching, season, watched, showing, excited, stream, series, catch, missed, shown
service	running, services, preaching, praise, run, prayer, praying, blessing, beginning, marathon
exam	quiz, study, test, studying, teaching, lesson, class, lecture, homework, workout
listen	listened, listening, songs, dropping, repeat, drop, buy, song, downloaded, download
lesson	studying, write, homework, workout, wore, finish, worked, learned, asked, finished
loving	enjoyed, waking up, exhausted, confused, finish, boring, doing, annoying, spent, finished
passed	dies, died, death, funeral, crashed, rest, anniversary, war, killed, passing
begins	continues, returning, beginning, return, stops, living, prepare, begin, evening, move

Table 7.4: Example paraphrases based on KL divergence between event phrases' type distributions, θ.

	precision				
# calendar entries	ngram baseline	entity + date	event phrase	event type	entity + date + event + type
100	0.50	0.90	0.86	0.72	0.70
500	0.46	0.66	0.56	0.54	0.42
1,000	0.44	0.52	0.42	0.40	0.32

Table 7.5: Evaluation of precision at different recall levels (generated by varying the threshold of the G^2 statistic).

7.7 Experiments

To estimate the quality of the calendar entries generated using our approach we manually evaluated a sample of the top 100, 500 and 1,000 calendar entries occurring within a 2-week future window of November 3rd.

7.7.1 Data

For evaluation purposes, we gathered roughly the 100 million most recent tweets on November 3rd 2011 (collected using the Twitter Streaming API[5], and tracking a broad set of temporal keywords, including "today", "tomorrow", names of weekdays, months, etc.).

We extracted named entities in addition to event phrases, and temporal expressions from the text of each of the 100M tweets. We then added the extracted triples to the dataset used for inferring event types described in Section 7.6, and performed 50 iterations of Gibbs sampling for predicting event types on the new data, holding the hidden variables in the original data constant. This *streaming* approach to inference is similar to that presented by Yao et al. [193].

We then ranked the extracted events as described in Section 7.5, and randomly sampled 50 events from the top ranked 100, 500, and 1,000. We annotated the events with 4 separate criteria:

[4]KL-Divergence is an information theoretic measure of similarity between two probability distributions which is defined as follows $\sum_i P_1(i) \log \frac{P_1(i)}{P_2(i)}$

[5]`https://dev.twitter.com/docs/streaming-api`

1. Is there a significant event involving the extracted entity which will take place on the extracted date?

2. Is the most frequently extracted event phrase informative?

3. Is the event's type correctly classified?

4. Are each of (1-3) correct? That is, does the event contain a correct entity, date, event phrase, and type?

Note that if (1) is marked as incorrect for a specific event, subsequent criteria are always marked incorrect.

7.7.2 Baseline

To demonstrate the importance of natural language processing and information extraction techniques in extracting informative events, we compare against a simple baseline which does not make use of our named entity recognizer or our event recognizer; instead, it considers all 1-4 grams in each tweet as candidate calendar entries, relying on the G^2 test to filter out phrases which have low association with each date.

7.7.3 Results

The results of the evaluation are displayed in table 7.5. The table shows the precision of the systems at different yield levels (number of aggregate events). These are obtained by varying the thresholds in the G^2 statistic. Note that the baseline is only comparable to the third column, i.e., the precision of (entity, date) pairs, since the baseline is not performing event identification and classification. Although in some cases ngrams do correspond to informative calendar entries, the precision of the ngram baseline is extremely low compared with our system.

In many cases the ngrams don't correspond to salient entities related to events; they often consist of single words which are difficult to interpret, for example "Breaking" which is part of the movie "Twilight: Breaking Dawn" released on November 18. Although the word

"Breaking" has a strong association with November 18, by itself it is not very informative to present to a user.[6]

Our high-confidence calendar entries are surprisingly high quality. If we limit the data to the 100 highest ranked calendar entries over a two-week date range in the future, the precision of extracted (entity, date) pairs is quite good (90%) - an 80% increase over the ngram baseline. As expected precision drops as more calendar entries are displayed, but remains high enough to display to users (in a ranked list). In addition to being less likely to come from extraction errors, highly ranked entity/date pairs are more likely to relate to popular or important events, and are therefore of greater interest to users.

In addition we present a sample of extracted future events on a calendar in figure 7.6 in order to give an example of how they might be presented to a user. We present the top 5 entities associated with each date, in addition to the most frequently extracted event phrase, and highest probability event type.

7.7.4 Error Analysis

We found 2 main causes for why entity/date pairs were uninformative for display on a calendar, which occur in roughly equal proportion:

Segmentation Errors Some extracted "entities" or ngrams don't correspond to named entities or are generally uninformative because they are mis-segmented. Examples include "RSVP", "Breaking" and "Yikes".

Weak Association between Entity and Date In some cases, entities are properly segmented, but are uninformative because they are not strongly associated with a specific event on the associated date, or are involved in many different events which happen to occur on that day. Examples include locations such as "New York", and frequently mentioned entities, such as "Twitter".

[6] In addition, we notice that the ngram baseline tends to produce many near-duplicate calendar entries, for example: "Twilight Breaking", "Breaking Dawn", and "Twilight Breaking Dawn". While each of these entries was annotated as correct, it would be problematic to show this many entries describing the same event to a user.

7.8 Related Work

While we are the first to study open domain event extraction within Twitter, there are two key related strands of research: extracting specific types of events from Twitter, and extracting open-domain events from news [142].

Recently there has been much interest in information extraction and event identification within Twitter. Benson et al. [8] use distant supervision to train a relation extractor which identifies artists and venues mentioned within tweets of users who list their location as New York City. Sakaki et al. [159] train a classifier to recognize tweets reporting earthquakes in Japan; they demonstrate their system is capable of recognizing almost all earthquakes reported by the Japan Meteorological Agency. Additionally there is recent work on detecting events or tracking topics [105] in Twitter which does not extract structured representations, but has the advantage that it is not limited to a narrow domain. Petrović et al. investigate a streaming approach to identifying tweets which are the first to report a breaking news story using Locally Sensitive Hash Functions [138]. Becker et al. [6], Popescu et al. [141; 140] and Lin et al. [101] investigate discovering clusters of related words or tweets which correspond to events in progress. In contrast to previous work on Twitter event identification, our approach is independent of event type or domain and is thus more widely applicable. Additionally, our work focuses on extracting a calendar of events (including those occurring in the future), extracting event-referring expressions and categorizing events into types.

Also relevant is work on identifying events [93; 40; 11], and extracting timelines [106] from news articles.[7] Twitter status messages present both unique challenges and opportunities when compared with news articles. Twitter's noisy text presents serious challenges for NLP tools. On the other hand, it contains a higher proportion of references to present and future dates. Tweets do not require complex reasoning about relations between events in order to place them on a timeline as is typically necessary in long texts containing narratives [169]. Additionally, unlike news, tweets often discus mundane events which are not of general interest, so it is crucial to exploit redundancy of information to assess whether an event is significant.

---

[7] `http://newstimeline.googlelabs.com/`

Previous work on open-domain information extraction [4; 183; 60] has mostly focused on extracting relations (as opposed to events) from web corpora and has also extracted relations based on verbs. In contrast, this work extracts events, using tools adapted to Twitter's noisy text, and extracts event phrases which are often adjectives or nouns, for example: *Super Bowl* **Party** *on Feb 5th.*

Finally we note that there has recently been increasing interest in applying NLP techniques to short informal messages such as those found on Twitter. For example, recent work has explored Part of Speech tagging [69], geographical variation in language found on Twitter [54; 55], modeling informal conversations [150; 151; 39], and also applying NLP techniques to help crisis workers with the flood of information following natural disasters [124; 97; 126].

7.9 Conclusions

We have presented a scalable and open-domain approach to extracting and categorizing events from status messages. We evaluated the quality of these events in a manual evaluation showing a clear improvement in performance over an ngram baseline.

We proposed a novel approach to categorizing events in an open-domain text genre with unknown types. Our approach based on latent variable models first discovers event types which match the data, which are then used to classify aggregate events without any annotated examples. Because this approach is able to leverage large quantities of unlabeled data, it outperforms a supervised baseline by 14%.

A possible avenue for future work is extraction of even richer event representations, while maintaining domain independence. For example: grouping together related entities, classifying entities in relation to their roles in the event, thereby, extracting a frame-based representation of events.

Our work makes a significant headway into our general goal of reducing information overload by creating aggregate representations for the end-user. A possible avanue for future work is adding functionality for personalized calendars based on the events of interest. We also wish to induce semantic frames for open-domain events via unsupervised approaches, using ideas similar to [26].

Chapter 8

CONCLUSIONS AND FUTURE WORK

In this thesis we explored the challenges and opportunities which arise when extracting information form large, heterogeneous and informal text corpora such as Twitter and the Web. We developed a set of NLP tools which have been tuned to work on noisy Twitter text, and have been made available online.[1] These tools were then leveraged to extract a calendar of popular events occurring in the near future from Twitter. The highly diverse nature of information expressed in these corpora was addressed through a series of unsupervised and weakly supervised methods which are able to learn from large quantities of unlabeled text, and which are based on probabilistic latent variable models. We showed that for semantic processing tasks where the number of categories is large and not known a-priori, the ability to exploit large unlabeled text corpora is necessary, and relying on small manually annotated text corpora and supervised learning is not practical.

In Chapter 3 we showed that off-the-shelf NLP tools are insufficient for handling Twitter's unique and noisy style. Issues such as misspellings and abbreviations, unreliable capitalization, and unique grammar cause news-trained NLP tools to perform very poorly on Tweets. In response we developed a set of Twitter-tuned NLP tools which are trained on an in-domain annotated corpus. These shallow syntactic annotation tasks are more challenging in noisy text than grammatical texts such as news articles, however once we get past the noisy text issues, other challenges become more manageable due to Twitter's simple discourse structure.

Discourse is one of the more challenging and unsolved problems in NLP. How to link information across sentences is not solved in the same way that we know how to take and individual sentence and extract its meaning. By working with short informal messages from Twitter we are able to sidestep some of these challenging discourse issues. Exploiting these

[1] `https://github.com/aritter/twitter_nlp`

intuitions about Twitter's simple discourse structure, and our Twitter-tuned NLP tools we were able to automatically extract a calendar of popular events occurring in the near future in Chapter 7.

In addition this thesis described a series of new probabilistic models for open-domain information extraction. We presented a new approach to distant supervision based on topic models which is capable of handling the situation of highly ambiguous training data, for example as arises in the case of weakly supervised named entity categorization. We investigated the issue of missing data in distant supervision, presenting an extension to the model of Hoffmann et. al. [80] which results in large improvements to precision and recall. In addition we showed how to model selectional preferences using topic models, and demonstrated the effectiveness of this approach on a pseudodisambiguation evaluation as well as through evaluation on the task of filtering improper applications of inference rules in context. Finally we proposed a latent variable model which is capable of automatically discovering an appropriate set of types to describe events automatically extracted from Twitter.

8.1 Future Work

Moving forward, there are many unsolved and fundamental challenges to be addressed towards the construction of scalable natural language understanding systems. This section focuses specifically on the challenges and opportunities that arise from user-generated text, time-sensitive information contained in text and the ever increasing availability of structured data and unstructured text. Several possible directions for future work are outlined below.

8.1.1 Populating the Realtime Semantic Web

The semantic web vision is beginning to take shape based on efforts such as Freebase, which present a consistent schema across large volumes of structured data from a huge variety of domains. Such systems currently don't address realtime data, however. Imagine the ability to receive alerts based on queries which link information across both realtime text and structured data. For example consider a system which is able to alert you instantly when performances by Grammy nominated artists in your hometown are announced, enabling

you to buy tickets before they sell out. Information extracted from text will likely play an even more crucial role in the population of the realtime semantic web, as people are generally more likely to mention new information in text than to fill out structured data forms describing it.

8.1.2 Forecasting the News

Given that we can extract events occurring in the future, an exciting question is whether it is possible to predict which will turn into important news stories before they take place. More concisely stated: *How well can we predict the topic of the front-page article of the New York Times for an arbitrary date in the near future?* This is actually quite easy in some cases, for example I would bet that on the Wednesday after the first Monday in November of any election year, the top story will be mostly about who won the U.S. elections. For other dates the answer may be less obvious however, as multiple events could compete for the top spot. A tool which can accurately predict the most important news in advance would be useful for information analysts, journalists, event planners, political campaigns or advertisers. Realtime text such as newspapers, blogs, Facebook and Twitter provide us with a source of information for making such predictions. Perhaps the number and source of future event mentions could serve as an indicator of their importance, in addition to features such as the category of event, entities involved, or user sentiment. To learn a model which makes such predictions, we have access to virtually limitless amounts of naturally occurring training data in the form of important news articles paired with corpora of historically written texts which reference them in future tense.

8.1.3 Planetary Scale Language Grounding

Children learn language by hearing it used in appropriate context, not by observing large corpora of linguistic annotations. Previous work on grounded language acquisition has focused on limited environments, however the emergence of large quantities of user-generated realtime text presents us with a unique opportunity to ground domain-independent language in a diverse set of sensor measurements taken from the real world at scale. We would like to

extend our previous work building semantic models from large quantities of unlabeled text to include non-textual sources of information and thereby ground the meaning of these distributional semantics in real-world sensor data. Imagine a system which can link comments on a recent news story with shifts in political polling numbers, realtime sports commentary with box score data from baseball games, comments on the weather with meteorological data, mentions of earthquakes with spikes in seismographic data, and complaints about traffic jams with public data on traffic flow. *Is it possible to ground language meaning in sensor data at the scale of all important events taking place in the world?*

8.1.4 Event Reference Resolution in Microblogs

An individual tweet can refer to an event in a wide variety of different ways. Because tweets are short, they are likely to mention only a subset of entities involved in the event, and might not even mention the date on which it occurs. Also there are many event phrases which can be used to refer to the same type of event, for example either "game" or "plays" might be used to refer to a sports event.

To address these issues, an important task is to cluster together all mentions of the same distinct event. More accurate grouping of mentions into co-referring events will be useful for many purposes, such as providing a better estimate of the number of references to an individual event (and thus a better estimate of it's importance), deduplication of events,[2] in addition to providing more complete information about each event for upstream tasks such as schema discovery (see below). While there has been previous work on event reference resolution in news articles, this is likely an even more important problem in the context of Tweets, which are not organized into documents.

8.1.5 Event Schema Discovery

The entities involved in a given event can be seen as filling a particular field in the schema determined by the event's type. For example, a ProductRelease event might be expected to involve a *Company* and *Product* (for instance *Apple* and *iPhone 5*). For systems focused

[2]Our current representation of events, while simple and effective, does lead to duplicates in situations where multiple entities are involved in the same event.

on extracting events within a narrow domain [8], the schema is easily specified in advance. Because our events and types are extracted in a fully open-domain manner, however, we have no way of knowing in advance what schema is appropriate for each of the types discovered by our model [153].

Recent work by Cafarella et. al. [20] has explored schema discovery for open-domain relation extraction using a data-mining style algorithm. We would like to investigate an analogous approach to schema discovery for open-domain events extracted from Twitter. Additionally we would like to explore an approach based on probabilistic generative models, which could enable schema induction to be performed jointly with event type categorization and event reference resolution, potentially leading to better performance than a pipelined approach to these problems.

8.1.6 Learning to Resolve Temporal References Through Weak Supervision

In order to place events on a calendar, we need to extract and resolve temporal expressions. For example, given the phrase "next Friday" we should be able to determine the unique calendar date which is referenced because we have the time at which the tweet was written. Currently we are using TempEx for this purpose [111]. TempEx is a rule-based system for extracting and resolving temporal expressions designed for use in newswire text, which we have found to have high precision when applied to tweets. TempEx's high precision can be explained by the fact that many temporal expressions are relatively unambiguous, however the creative spelling variations often employed on Twitter (for example "nxt fri") lead to low recall.

A natural approach to improving recall would be to annotate a large corpus of tweets with temporal expressions and manually resolve them to unique calendar dates for use as in-domain training data by machine learning models. This annotation would be extremely tedious and time-consuming, however, as non-trivial effort is required to calculate the reference date for each annotated temporal expression.

As an alternative to manual annotation, we would like to investigate a distantly supervised approach to extracting and resolving temporal expressions. First, note that it

is easy to retrospectively identify when important events take place based on frequency spikes; significant {`Entity,Date`} tuples can readily be identified using a statistical test. Once significant events are identified, we can apply a *distant supervision* assumption [121] which states that tweets written within a fixed window around the date of the event[3] which mention a key entity involved in the event should also refer to the date on which the event takes place. As a concrete example, assume the entity "LinkedIn" is frequently mentioned on May 19, 2011 (the date of their IPO); we could then gather all tweets which mention "LinkedIn" near the 19th, for example:

Tweeted on May 16: LinkedIn will go public on Thursday. I guess we'll see what happens.

Tweeted on May 20: LinkedIn debuted on the NYSE yesterday - currently at $99.50/share (premarket). WAY too overpriced: http://bit.ly/mtV8oO Thoughts?

We can consider each such {`Tweet,ReferenceDate,TargetDate`} tuple as a (distantly supervised) training example. Negative examples can consist of dates within a window other than the target date, and feature templates can consist of conjunctions of lexical features from the tweet and properties of the target date. Example features generated using these templates could include:

- TIME = PAST $\wedge$ WORD = "debuted"
- DOW = WEDNESDAY $\wedge$ WORD = "wed".

In practice the distant supervision assumption presented above may be too unrealistic to be effective because tweets which mention an event won't always mention the date on which it takes place. We therefore may need to introduce additional latent variables which determine whether each word is part of a temporal expression. One possible approach would be to model temporal expressions using a constrained Hidden Markov Model, where a subset of the hidden states are arbitrarily mapped to date properties in advance (e.g. TIME = PAST, DOW = WEDNESDAY, WEEK = NEXT, etc...) and during training the transition

[3]For example += 2 weeks.

matrices of the HMMs are constrained such that the probability of transitioning into each of these states is zero, except if the associated target date has that property active. Parameters would be shared across HMMs except for those set to zero due to constraints. Learning could proceed in a straightforward fashion using EM (the forward-backward algorithm) or sampling-based inference [70]. During inference (decoding) the HMM would be left unconstrained. After decoding, temporal expressions with associated properties of the target date could be read off based on the inferred values of hidden variables. If necessary, feature engineering (and overlapping correlated features such as prefixes, suffixes and dictionary-based features) could be accommodated by modeling the emission distributions with locally normalized logistic regression models [9]. Note that an additional (distantly supervised) classification step would likely be necessary as date properties are often left semantically ambiguous, even though they are pragmatically well defined. For example in the absence of an explicit past tense marker a phrase like "the Wednesday meeting" is usually assumed to refer to the future.

BIBLIOGRAPHY

[1] Steven Abney and Marc Light. Hiding a semantic class hierarchy in a markov model. In *In Proceedings of the ACL Workshop on Unsupervised Learning in Natural Language Processing*, 1998.

[2] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, 2000.

[3] James Allan, Ron Papka, and Victor Lavrenko. On-line new event detection and tracking. In *SIGIR*, 1998.

[4] Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open information extraction from the web. In *In IJCAI*, 2007.

[5] Michele Banko and Oren Etzioni. The tradeoffs between open and traditional relation extraction. In *ACL-08: HLT*, 2008.

[6] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. In *ICWSM*, 2011.

[7] Cosmin Bejan, Matthew Titsworth, Andrew Hickl, and Sanda Harabagiu. Nonparametric bayesian models for unsupervised event coreference resolution. In *NIPS*. 2009.

[8] Edward Benson, Aria Haghighi, and Regina Barzilay. Event discovery in social media feeds. In *The 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, USA, 2011. To appear.

[9] Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. Painless unsupervised learning with features. In *NAACL*, 2010.

[10] Shane Bergsma, Dekang Lin, and Randy Goebel. Discriminative learning of selectional preference from unlabeled text. In *EMNLP*, 2008.

[11] Steven Bethard and James H. Martin. Identification of event mentions and their semantic class. In *EMNLP*, 2006.

[12] David M. Blei and John D. Lafferty. Correlated topic models. In *NIPS*, 2005.

[13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 2003.

[14] Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled sata with co-training. In *COLT*, pages 92–100, 1998.

[15] Sergey Brin. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*. 1999.

[16] Sergey Brin. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*. 1999.

[17] Samuel Brody and Mirella Lapata. Bayesian word sense induction. In *EACL*, pages 103–111, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[18] Peter F. Brown, Peter V. deSouza, Robert L. Mercer, Vincent J. Della Pietra, and Jenifer C. Lai. Class-based n-gram models of natural language. *Comput. Linguist.*, 1992.

[19] Razvan Bunescu and Raymond Mooney. Learning to extract relations from the web using minimal supervision. In *Annual meeting-association for Computational Linguistics*, 2007.

[20] Michael J. Cafarella, Dan Suciu, and Oren Etzioni. Navigating extracted data with schema discovery. In *WebDB*, 2007.

[21] Andrew Carlson, Justin Betteridge, Estevam R. Hruschka, and Tom M. Mitchell. Coupling semi-supervised learning of categories and relations. In *NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing*, 2009.

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

[23] Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka, Jr., and Tom M. Mitchell. Coupled semi-supervised learning for information extraction. In *Proceedings of the third ACM international conference on Web search and data mining*, WSDM '10, 2010.

[24] Andrew Carlson, Justin Betteridge, Richard C. Wang, Estevam R. Hruschka Jr., and Tom M. Mitchell. Coupled semi-supervised learning for information extraction. In *WSDM 2010*, 2010.

[25] Nathanael Chambers and Dan Jurafsky. 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, ACL-IJCNLP '09, pages 602–610, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[26] Nathanael Chambers and Dan Jurafsky. Template-based information extraction without the templates. In *Proceedings of ACL*, 2011.

[27] Nathanael Chambers, Shan Wang, and Dan Jurafsky. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Association for Computational Linguistics, 2007.

[28] Nathanael Chambers, Shan Wang, and Dan Jurafsky. Classifying temporal relations between events. In *ACL*, 2007.

[29] Eugene Charniak, Curtis Hendrickson, Neil Jacobson, and Mike Perkowitz. Equations for part-of-speech tagging. In *AAAI*, pages 784–789, 1993.

[30] Harr Chen, S. R. K. Branavan, Regina Barzilay, and David R. Karger. Global models of document structure using latent permutations. In *NAACL*, 2009.

[31] Massimiliano Ciaramita and Mark Johnson. Explaining away ambiguity: Learning verb selectional preference with bayesian networks. In *COLING*, 2000.

[32] Stephen Clark and David Weir. Class-based probability estimation using a semantic hierarchy. *Comput. Linguist.*, 2002.

[33] Michael Collins. Discriminative training methods for hidden markov models: theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*, 2002.

[34] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Empirical Methods in Natural Language Processing*, 1999.

[35] Mark Craven, Johan Kumlien, et al. Constructing biological knowledge bases by extracting information from text sources. In *ISMB*, 1999.

[36] Aron Culotta and Andrew McCallum. Confidence estimation for information extraction. In *Proceedings of HLT-NAACL 2004: Short Papers on XX*, HLT-NAACL '04, pages 109–112, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

[37] Ido Dagan, Lillian Lee, and Fernando C. N. Pereira. Similarity-based models of word cooccurrence probabilities. In *Machine Learning*, 1999.

[38] Nilesh Dalvi, Ravi Kumar, and Bo Pang. Object matching in tweets with spatial models. In *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012.

[39] Cristian Danescu-Niculescu-Mizil, Michael Gamon, and Susan Dumais. Mark my words! Linguistic style accommodation in social media. In *Proceedings of WWW*, pages 745–754, 2011.

[40] Anish Das Sarma, Alpa Jain, and Cong Yu. Dynamic relationship and event discovery. In *WSDM*, 2011.

[41] Hal Daumé III. Fast search for dirichlet process mixture models. *arXiv preprint arXiv:0907.1812*, 2009.

[42] Hal Daumé III and Daniel Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 2006.

[43] Hal Daume III. hbc: Hierarchical bayes compiler. http://hal3.name/hbc. 2007.

[44] Leon Derczynski, Diana Maynard, Niraj Aswani, and Kalina Bontcheva. Microblog-genre noise and impact on semantic annotation accuracy. In *HT*, pages 21–30, 2013.

[45] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 1997.

[46] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation. *LREC*, 2004.

[47] Doug Downey, Matthew Broadhead, and Oren Etzioni. Locating complex named entities in web text. In *Proceedings of the 20th international joint conference on Artifical intelligence*, 2007.

[48] Doug Downey, Oren Etzioni, and Stephen Soderland. A probabilistic model of redundancy in information extraction. In *Proceedings of the 19th international joint conference on Artificial intelligence*, 2005.

[49] Doug Downey, Oren Etzioni, and Stephen Soderland. Analysis of a probabilistic model of redundancy in unsupervised information extraction. *Artif. Intell.*, 174(11):726–748, 2010.

[50] Doug Downey, Stefan Schoenmackers, and Oren Etzioni. Sparse information extraction: Unsupervised language models to the rescue. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007.

[51] Gregory Druck, Gideon Mann, and Andrew McCallum. Leveraging existing resources using generalized expectation criteria. In *Proceedings of the Neural Information Processing Systems (NIPS) Workshop on Learning Problem Design*, 2007.

[52] Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.*, 1993.

[53] Jacob Eisenstein. What to do about bad language on the internet. In *Proceedings of NAACL-HLT*, pages 359–369, 2013.

[54] Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. A latent variable model for geographic lexical variation. In *EMNLP*, 2010.

[55] Jacob Eisenstein, Noah A. Smith, and Eric P. Xing. Discovering sociolinguistic associations with structured sparsity. In *ACL-HLT*, 2011.

[56] Micha Elsner, Eugene Charniak, and Mark Johnson. Structured generative models for unsupervised named-entity clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, 2009.

[57] Katrin Erk. A simple, similarity-based model for selectional preferences. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 2007.

[58] Elena Erosheva, Stephen Fienberg, and John Lafferty. Mixed-membership models of scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*, 2004.

[59] Oren Etzioni, Michael Cafarella, Doug Downey, Ana maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alex Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 2005.

[60] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *EMNLP*, 2011.

[61] Tim Finin, Will Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. Annotating named entities in Twitter data with crowdsourcing. In *Proceedings of the NAACL Workshop on Creating Speech and Text Language Data With Amazon's Mechanical Turk*. Association for Computational Linguistics, June 2010.

[62] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, 2005.

[63] Michael Fleischman and Eduard Hovy. Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics - Volume 1*, COLING '02, pages 1–7, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[64] Radu Florian. Named entity recognition as a house of cards: classifier stacking. In *Proceedings of the 6th conference on Natural language learning - Volume 20*, COLING-02, 2002.

[65] Eric N. Forsythand and Craig H. Martell. Lexical and discourse analysis of online chat dialog. In *Proceedings of the International Conference on Semantic Computing*, 2007.

[66] Evgeniy Gabrilovich, Susan Dumais, and Eric Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In *WWW*, 2004.

[67] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2003.

[68] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Comput. Linguist.*, 2002.

[69] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *ACL*, 2011.

[70] Sharon Goldwater and Tom Griffiths. A fully bayesian approach to unsupervised part-of-speech tagging. In *ACL*, 2007.

[71] Joshua T. Goodman. A bit of progress in language modeling. Technical report, Microsoft Research, 2001.

[72] Stephan Gouws, Donald Metzler, Congxing Cai, and Eduard Hovy. Contextual bearing on linguistic variation in social media. In *ACL Workshop on Language in Social Media*, Portland, Oregon, USA, 2011. To appear.

[73] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, April 2004.

[74] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proc Natl Acad Sci U S A*, 2004.

[75] Ralph Grishman and Beth Sundheim. Message understanding conference - 6: A brief history. In *Proceedings of the International Conference on Computational Linguistics*, 1996.

[76] Mark Hachman. Humanity's tweets: Just 20 terabytes. In *PCMAG.COM*, 2011.

[77] Bo Han and Timothy Baldwin. Lexical normalisation of short text messages: Makn sens a #twitter. In *The 49th Annual Meeting of the Association for Computational Linguistics*, Portland, Oregon, USA, 2011. To appear.

[78] Gregor Heinrich. Parameter estimation for text analysis. Technical report, 2004.

[79] Gregor Heinrich. Parameter estimation for text analysis. Technical report, University of Leipzig, 2008.

[80] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 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*, 2011.

[81] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 1999.

[82] Feng Jiao, Shaojun Wang, Chi-Hoon Lee, Russell Greiner, and Dale Schuurmans. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 209–216, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[83] Frank Keller and Mirella Lapata. Using the web to obtain frequencies for unseen bigrams. *Comput. Linguist.*, 2003.

[84] Kevin Knight. Bayesian Inference with Tears. Technical report, 2009.

[85] Catherine Kobus, François Yvon, and Géraldine Damnati. Normalizing sms: are two metaphors better than one ? In *COLING*, pages 441–448, 2008.

[86] Stanley Kok and Pedro Domingos. Extracting semantic networks from text via relational clustering. In *ECML PKDD*, 2008.

[87] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[88] Zornitsa Kozareva and Eduard Hovy. Learning arguments and supertypes of semantic relations using recursive patterns. In *ACL*, 2010.

[89] Zornitsa Kozareva and Eduard Hovy. Not all seeds are equal: Measuring the quality of text mining seeds. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2010.

[90] Zornitsa Kozareva and Eduard H. Hovy. Not all seeds are equal: Measuring the quality of text mining seeds. In *HLT-NAACL*, 2010.

[91] Zornitsa Kozareva, Ellen Riloff, and Eduard Hovy. Semantic class learning from the web with hyponym pattern linkage graphs. In *ACL-08: HLT*, 2008.

[92] Zornitsa Kozareva, Sonia Vazquez, and Andres Montoyo. Domain information for fine-grained person name categorization. In *Proceedings of the 9th international conference on Computational linguistics and intelligent text processing*, CICLing'08, pages 311–321, Berlin, Heidelberg, 2008. Springer-Verlag.

[93] Giridhar Kumaran and James Allan. Text classification and named entities for new event detection. In *SIGIR*, 2004.

[94] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[95] Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *ACL*, 2011.

[96] Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *KDD*, 2009.

[97] William Lewis, Robert Munro, and Stephan Vogel. Crisis mt: Developing a cookbook for mt in crisis situations. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 2011.

[98] Hang Li and Naoki Abe. Generalizing case frames using a thesaurus and the mdl principle. *Comput. Linguist.*, 1998.

[99] Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 2006.

[100] Percy Liang, Michael I Jordan, and Dan Klein. Type-based mcmc. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010.

[101] Cindy X. Lin, Bo Zhao, Qiaozhu Mei, and Jiawei Han. PET: a statistical model for popular events tracking in social communities. In *KDD*, 2010.

[102] Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 17th international conference on Computational linguistics*, 1998.

[103] Dekang Lin. Dependency-based evaluation of minipar. In *Proc. Workshop on the Evaluation of Parsing Systems*, 1998.

[104] Dekang Lin and Patrick Pantel. Dirt-discovery of inference rules from text. In *KDD*, 2001.

[105] Jimmy Lin, Rion Snow, and William Morgan. Smoothing techniques for adaptive online language models: Topic tracking in tweet streams. In *KDD*, 2011.

[106] Xiao Ling and Daniel S. Weld. Temporal information extraction. In *AAAI*, 2010.

[107] Roderick J A Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, Inc., New York, NY, USA, 1986.

[108] Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. Recognizing named entities in tweets. In *ACL*, 2011.

[109] Brian Locke and James Martin. Named entity recognition: Adapting to microblogging. In *Senior Thesis, University of Colorado*, 2009.

[110] Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. Machine learning of temporal relations. In *ACL*, 2006.

[111] Inderjeet Mani and George Wilson. Robust temporal processing of news. In *ACL*, 2000.

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

[113] Mitchell P. Marcus, Beatrice Santorini, and Mary A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 1994.

[114] Mausam, Michael Schmitz, Robert Bart, Stephen Soderland, and Oren Etzioni. 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*, 2012.

[115] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. In *http://mallet.cs.umass.edu*, 2002.

[116] Tara McIntosh. Unsupervised discovery of negative categories in lexicon bootstrapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, 2010.

[117] Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In *KDD*, 2007.

[118] David Mimno, Hanna M. Wallach, Jason Naradowsky, David A. Smith, and Andrew McCallum. Polylingual topic models. In *EMNLP*, 2009.

[119] Bonan Min, Ralph Grishman, Li Wan, Chang Wang, and David Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of NAACL-HLT*, pages 777–782, 2013.

[120] Einat Minkov, Richard C. Wang, and William W. Cohen. Extracting personal names from email: applying named entity recognition to informal text. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 443–450, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[121] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP 2009*, 2009.

[122] Robert C. Moore. On log-likelihood-ratios and the significance of rare events. In *EMNLP*, 2004.

[123] Christoph Müller and Michael Strube. Multi-level annotation of linguistic data with MMAX2. In Sabine Braun, Kurt Kohn, and Joybrato Mukherjee, editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*, pages 197–214. Peter Lang, Frankfurt a.M., Germany, 2006.

[124] Robert Munro. Subword and spatiotemporal models for identifying actionable information in Haitian Kreyol. In *CoNLL*, 2011.

[125] R. Nallapati and W. Cohen. Link-plsa-lda: A new unsupervised model for topics and influence of blogs. In *ICWSM*, 2008.

[126] Graham Neubig, Yuichiroh Matsubayashi, Masato Hagiwara, and Koji Murakami. Safety information mining - what can NLP do in a disaster -. In *IJCNLP*, 2011.

[127] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed algorithms for topic models. *JMLR*, 2009.

[128] David Newman, Arthur U. Asuncion, Padhraic Smyth, and Max Welling. Distributed inference for latent dirichlet allocation. In *NIPS*, 2007.

[129] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *NAACL-HLT*, 2010.

[130] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *HLT-NAACL*, 2010.

[131] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2002.

[132] Joakim Nivre, Johan Hall, and Jens Nilsson. Memory-based dependency parsing. In *Proceedings of CoNLL*, 2004.

[133] Diarmuid Ó Séaghdha. Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010.

[134] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig latin: a not-so-foreign language for data processing. In *SIGMOD*, 2008.

[135] Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard H. Hovy. Isp: Learning inferential selectional preferences. In *HLT-NAACL*, 2007.

[136] Patrick Pantel, Eric Crestan, Arkady Borkovsky, Ana M. Popescu, and Vishnu Vyas. Web-scale distributional similarity and entity set expansion. In *EMNLP*, 2009.

[137] Patrick Andre Pantel. *Clustering by committee*. PhD thesis, University of Alberta, Edmonton, Alta., Canada, 2003.

[138] Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to twitter. In *HLT-NAACL*, 2010.

[139] Hoifung Poon and Pedro Domingos. Joint inference in information extraction. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1*, pages 913–918. AAAI Press, 2007.

[140] Ana-Maria Popescu and Marco Pennacchiotti. Dancing with the stars, nba games, politics: An exploration of twitter users' response to events. In *ICWSM*, 2011.

[141] Ana-Maria Popescu, Marco Pennacchiotti, and Deepa Arun Paranjpe. Extracting events and event descriptions from twitter. In *WWW*, 2011.

[142] J. Pustejovsky, P. Hanks, R. Sauri, A. See, R. Gaizauskas, A. Setzer, D. Radev, B. Sundheim, D. Day, L. Ferro, and M. Lazo. The TIMEBANK corpus. In *Proceedings of Corpus Linguistics 2003*, 2003.

[143] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: a supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 248–256, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[144] Joseph Reisinger and Marius Pasca. Latent variable models of concept-attribute attachment. 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*, 2009.

[145] P. Resnik. Selectional constraints: an information-theoretic model and its computational realization. *Cognition*, 1996.

[146] Philip Resnik. Selectional preference and sense disambiguation. In *Proc. of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How?*, 1997.

[147] Philip Resnik and Eric Hardisty. Gibbs sampling for the uninitiated, 2009.

[148] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *ECML/PKDD (3)*, pages 148–163, 2010.

[149] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. Relation extraction with matrix factorization and universal schemas. In *Proceedings of NAACL-HLT*, 2013.

[150] Alan Ritter, Colin Cherry, and Bill Dolan. Unsupervised modeling of twitter conversations. In *HLT-NAACL*, 2010.

[151] Alan Ritter, Colin Cherry, and William B. Dolan. Data-driven response generation in social media. In *EMNLP*, 2011.

[152] Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. Named entity recognition in tweets: An experimental study. *EMNLP*, 2011.

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

[154] Alan Ritter, Mausam, and Oren Etzioni. A latent dirichlet allocation method for selectional preferences. In *ACL*, 2010.

[155] Kirk Roberts and Sanda M. Harabagiu. Unsupervised learning of selectional restrictions and detection of argument coercions. In *EMNLP*, 2011.

[156] Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. Inducing a semantically annotated lexicon via em-based clustering. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, 1999.

[157] Dan Roth and Wen-tau Yih. Global Inference for Entity and Relation Identification via a Linear Programming Formulation. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. The MIT press, 2007.

[158] Stuart J. Russell, Peter Norvig, John F. Candy, Jitendra M. Malik, and Douglas D. Edwards. *Artificial intelligence: a modern approach*. 1996.

[159] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In *WWW*, 2010.

[160] Christina Sauper, Aria Haghighi, and Regina Barzilay. Incorporating content structure into text analysis applications. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 377–387, Morristown, NJ, USA, 2010. Association for Computational Linguistics.

[161] Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. Evita: a robust event recognizer for qa systems. In *HLT-EMNLP*, 2005.

[162] Lenhart Schubert and Matthew Tong. Extracting and evaluating general world knowledge from the brown corpus. In *In Proc. of the HLT-NAACL Workshop on Text Meaning*, pages 7–13, 2003.

[163] Diarmuid Ó. Séaghdha. Latent variable models of selectional preference. In *ACL*, ACL '10, 2010.

[164] Satoshi Sekine. Named entity: History and future. In *New York University Technical Report 04-021*, 2004.

[165] Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, 2003.

[166] Sameer Singh, Dustin Hillard, and Chris Leggetter. Minimally-supervised extraction of entities from text advertisements. In *Human Language Technologies: Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL HLT)*, 2010.

[167] Sameer Singh, Karl Schultz, and Andrew McCallum. *Bi-directional Joint Inference for Entity Resolution and Segmentation Using Imperatively-Defined Factor Graphs*, volume 5782/2009 of *Lecture Notes in Computer Science*, ir Bi-directional Joint Inference for Entity Resolution and Segmentation Using Imperatively-Defined Factor Graphs, pages 414–429. Springer Berlin / Heidelberg, September, 2009.

[168] Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine learning*, 1999.

[169] Fei Song and Robin Cohen. Tense interpretation in the context of narrative. In *Proceedings of the ninth National conference on Artificial intelligence - Volume 1*, AAAI'91, 1991.

[170] Fabian Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO - a large ontology from Wikipedia and WordNet. *Elsevier Journal of Web Semantics*, 2008.

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

[172] Charles Sutton. Collective segmentation and labeling of distant entities in information extraction, 2004.

[173] Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, 2012.

[174] Partha Pratim Talukdar and Fernando Pereira. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1473–1481. Association for Computational Linguistics, 2010.

[175] Erik F. Tjong Kim Sang and Sabine Buchholz. Introduction to the conll-2000 shared task: chunking. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning - Volume 7*, ConLL '00, 2000.

[176] Kristina Toutanova and Mark Johnson. A bayesian lda-based model for semi-supervised part-of-speech tagging. In *Advances in Neural Information Processing Systems 20*. 2008.

[177] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, 2003.

[178] Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. EMNLP '00, 2000.

[179] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, 2010.

[180] Benjamin Van Durme and Daniel Gildea. Topic models for corpus-centric knowledge generalization. In *Technical Report TR-946, Department of Computer Science, University of Rochester, Rochester*, 2009.

[181] Vishnu Vyas, Patrick Pantel, and Eric Crestan. Helping editors choose better seed sets for entity set expansion. In *CIKM*, 2009.

[182] Vishnu Vyas, Patrick Pantel, and Eric Crestan. Helping editors choose better seed sets for entity set expansion. In *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009.

[183] Daniel S. Weld, Raphael Hoffmann, and Fei Wu. Using wikipedia to bootstrap open information extraction. *SIGMOD Rec.*, 2009.

[184] Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Alan Ritter, Sara Rosenthal, and Veselin Stoyanov. Semeval-2013 task 2: Sentiment analysis in twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2013.

[185] Fei Wu and Daniel S. Weld. Autonomously semantifying wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007.

[186] Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. Learning from bullying traces in social media. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012.

[187] Wei Xu, Raphael Hoffmann Le Zhao, and Ralph Grishman. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of ACL*, 2013.

[188] Yiming Yang, Tom Pierce, and Jaime Carbonell. A study of retrospective and online event detection. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, 1998.

[189] Tae Yano, William W. Cohen, and Noah A. Smith. Predicting response to political blog posts with topic models. In *NAACL*, 2009.

[190] Tae Yano, William W. Cohen, and Noah A. Smith. Predicting response to political blog posts with topic models. In *NAACL*, 2009.

[191] L. Yao, D. Mimno, and A. Mccallum. Efficient methods for topic model inference on streaming document collections. In *KDD*, 2009.

[192] Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew McCallum. Structured relation discovery using generative models. In *EMNLP*, 2011.

[193] Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.

[194] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*, ACL '95, 1995.

[195] Fabio Massimo Zanzotto, Marco Pennaccchiotti, and Kostas Tsioutsiouliklis. Linguistic redundancy in twitter. In *EMNLP*, 2011.

[196] Luke S. Zettlemoyer and Michael Collins. Online learning of relaxed ccg grammars for parsing to logical form. In *EMNLP-CoNLL*, 2007.