Unsupervised Induction of Frame-Based Linguistic Forms

by

Francis Michael Ostrowski Ferraro

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

Baltimore, Maryland

October, 2017

Abstract

This thesis studies the use of bulk, structured, linguistic annotations in order to perform unsupervised induction of meaning for three kinds of linguistic forms: words, sentences, and documents. The primary linguistic annotation I consider throughout this thesis are frames, which encode core linguistic, background or societal knowledge necessary to understand abstract concepts and real-world situations. I begin with an overview of linguistically-based structured meaning representation; I then analyze available large-scale natural language processing (NLP) and linguistic resources and corpora for their abilities to accommodate bulk, automatically-obtained frame annotations.

I then proceed to induce meanings of the different forms, progressing from the word level, to the sentence level, and finally to the document level. I first show how to use these bulk annotations in order to better encode linguistic- and cognitive science-backed semantic expectations within word forms. I then demonstrate a straightforward approach for learning large lexicalized and refined syntactic fragments, which encode and memoize commonly used phrases and linguistic constructions. Next, I

ABSTRACT

consider two unsupervised models for document and discourse understanding; one is a purely generative approach that naturally accommodates layer annotations and is the first to capture and unify a complete frame hierarchy. The other conditions on limited amounts of external annotations, imputing missing values when necessary, and can more readily scale to large corpora. These discourse models help improve document understanding and type-level understanding.

Primary Reader: Benjamin Van Durme, *Johns Hopkins University*

Secondary Reader: Mark Dredze, *Johns Hopkins University*

Tertiary Reader: Jordan Boyd-Graber, *University of Maryland, College Park*

Acknowledgments

This thesis would not have been possible without the support of many people. To my committee, Ben Van Durme, Mark Dredze, and Jordan Boyd-Graber: thank you very much for your suggestions. Ben, as my advisor, thank you for having faith in me, giving me the freedom to explore, and reorienting me when needed. You've helped me grow into a researcher, and helped me achieve my goals.

Colloquially, Johns Hopkins is said to be a hub of NLP; I agree. I was very lucky to be able to do my Ph.D. at JHU, be surrounded by so many great people, and have the support of three different organizations: the Department of Computer Science, the Johns Hopkins University Center for Language and Speech Processing, and the Johns Hopkins Human Language Technology Center of Excellence. While I have definitely benefited from interacting with the faculty and students across these three organizations, I would like to say special thanks to Matt Post and Jason Eisner. Both Matt and Jason taught me so much about teaching and mentorship; they provided support and counseled me from my start to my end at JHU. I would also like to acknowledge and thank Ruth Scally, Debbie Deford, Cathy Thornton, Laura Graham,

and Zachary Burwell, who ensure that the organizations all run smoothly. Without working machines, there are no experiments: I would like to thank everyone who keeps the machines running.

To my classmates, labmates, and colleagues, present and former, including Nick Andrews, Tongfei Chen, Ryan Cotterell, Matt Gormley, Gaurav Kumar, Keith Levin, Chandler May, Courtney Napoles, Michael Paul, Nanyun Peng, Adam Poliak, Pushpendre Rastogi, Darcey Riley, Rachel Rudinger, Keisuke Sakaguchi, Adam Teichert, Tim Vieira, Aaron White, Travis Wolfe, Xuchen Yao: whether we got to collaborate or just get coffee, thank you for your support.

Thank you to the NSF for the Graduate Research Fellowship, which materially gave me the freedom to forge my own agenda, and to the CLSP, HLTCOE, and the Maryland Advanced Research Computing Center (MARCC) for the compute and storage resources that I needed to complete this thesis.

To my family, thank you for the love and support you have shown.

Dedication

To ƨ99.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Motivation

As we become more conversant with, and rely more heavily on, automated assistants or information extraction systems such as *Google Now* or *Siri*, those systems will need to have a deeper understanding of our habits, experiences and expectations. We're surrounded by this information. Some of it we record and engage with explicitly: writing that work email which details an important process, chronicling our experiences into online journals and blogs, or reading the engaging, in-depth reporting of troubling situations and events from around the world. Other parts of this information are societal and encompass what we experience, but that are not necessarily discussed actively: "knowing" what to expect at a party, or reading between the lines in missives and responding appropriately.

Frames were originally meant to schematize the above information, and more (Minsky, 1974; Schank and Abelson, 1977; Fillmore, 1967, 1976, 1982, i.a.). They are

structured abstractions over *concepts*: they can describe (or prescribe) what conditions should be true for a given concept to be evoked; they can categorize and specify types of participants, objects, or other situations that can be part of a concept; and they can describe ways that we can communicate this information to one another. In some cases, those concepts might be words, e.g., what is likely to happen during a "blizzard"; phrases or sentences, e.g., what actions, and how were they completed, did Chris likely take given the sentence "Chris extracted the car from the snowbank"; or entire communications, e.g., after reading a weather report of impending snow, what actions, such as stocking up on essential items, might Chris take. In short, they encapsulate what we can call *common background knowledge*. However, specifying this type of information—be it manually or automatically—has been and continues to be a challenge (Sundheim, 1996; Strassel et al., 2017, i.a.).

In this thesis I explore methods to induce frame-based meaning representations of words, sentences, and documents in an unsupervised manner. I merge linguistic, artificial intelligence and cognitive science theory with modern machine learning and large scale natural language annotations in order to obtain and analyze these representations. These induced meaning representations, explicitly or implicitly, are structured and can offer a prioritization of expectations. I will predominantly, but not exclusively, focus on *events*, rather than people or participants.[1] These events will be those that are reported on in text.

[1]Of course, participants *are* a key part of understanding events.

Many (but not all) of the questions I ask and answer will tend more to the explanatory side: how well do certain induced representations align with human-provided expectations? What kinds of deep syntactic structures can be learned? And how can we better integrate lower-level information with higher-level information (possibly specified with errors, if at all) in order to better explain entire collections of documents? However, the explanatory questions are posed with end applications in mind; I conclude this thesis by exploring one such information extraction application.

In the rest of this chapter, I will consider potential applications of frames in §1.1: if we could, to a modest extent, accurately and comprehensively encode expectations about what will happen and who might participate in certain events, what future scientific questions might we be able to answer, or what future systems or tools might we be able to build? The applications discussed in this chapter are meant to inspire; they will not be answered directly in this thesis. In §1.2 I then briefly explore the challenges of extracting event knowledge; this exploration is a combination of (qualitatively) comparing different kinds of documents, computing lexical bias statistics, and simply examining event participant occurrences for different genres of documents. These analyses question an often unstated, but critical, assumption—namely that statistics acquired from a small corpus of domain specific text can be easily supplemented with statistics obtained from a much larger corpus of "general" text. In §1.3, I provide a roadmap and overview of contributions of this thesis.

1.1 Potential Applications of Frames

If we could specify frames accurately and with sufficient coverage, then what (scientific) questions could we answer? On the other hand, what systems could we build? An information extraction system, when presented with input documents, generally extracts and distills the core information being communicated—i.e., the "who," "what," "when," "where," "why," and "how." The general expectation encoded by frames could provide some default knowledge (answers) for such a system.

While computer scientists may recognize frames (or schemas) as the territory of Marvin Minsky, Roger Schank, and Charles Fillmore (described in more detail in chapters 3 and 7), they have a history within other disciplines, such as psychology and sociology too. Bartlett (1933) used the term "schema" to help explain how people remember things and events and Anderson (1977) developed "schema theory" for an educational context, in order to study how students learn. Relatedly, Goffman (1974) posited the idea of "conceptual frames" as a way to help understand social and interpersonal relationships. Inspired by Minsky (1974) i.a., Rumelhart (1980) argues that if schemas are meant to represent knowledge, then they form an essential part in understanding cognition.

Therefore, a (large) repository of accurate and comprehensive frames could help in the above endeavors: on the one hand, a frame repository may help elucidate previously unseen connections for domain experts (e.g., education experts, sociologists

or behavioral psychologists).[2] These hypothesized connections could then be tested in subsequent studies. On the other hand, a collection of frames could be used to computationally model, and potentially help make sense of, the ways people behave in certain contexts.

While accurate and comprehensive frames could generate scientific interest, they could also inspire new systems and applications to be built. First, just as a frame repository might be able to help pose scientific questions, could a frame repository similarly help an end user complete tasks more accurately or faster? For example, could frames, especially ones that tie the general knowledge they encode to language-specific syntactic constructs, help users translate documents from one language to another? Or perhaps frames could be used to personalize an information extraction tool; two different analysts for the same task can care about different aspects of the underlying events (Sundheim, 1992).

1.2 How Events Are Reported

Knowledge and event acquisition efforts glean common, background information from large, general corpora (Lin and Pantel, 2001; Van Durme and Schubert, 2008). However, the data used for knowledge acquisition is generally not the same as what is used for annotation by humans: e.g., general newswire may help provide coverage and statistical strength for learning broad patterns, but when text is presented to

[2]Such repositories are being explored for medical and health domains (Poon et al., 2014).

Three people have been fatally shot, and five people, incluidng a mayor, were seriously wounded as a result of a Shining Path attack today.

(a) The beginning paragraph of a sample information extraction document (Sundheim, 1992).

The death toll in the Los Angeles riots rose to 50 today. The Los Angeles County Coroner's Office is continuing to try to identify those killed, many of whom had no identification.

(b) The beginning paragraph of a newswire article (Sandhaus, 2008).

The 150-year-old weeping Camperdown elm in Prospect Park in Brooklyn stands a mere 12 feet high, but its intricate pattern of branches etches a 25-foot circle.
"What a glorious silhouette it is against the winter sky," said Robert Makla, a founder of the Friends of Prospect Park, which since 1966 has insured the tree's survival. "It's duplicated by no other tree or abstract work of art."
Weeping trees like this elm - plants whose limbs grow down, not up - are spectacular sculptures in winter and graceful accents in summer.

(c) The beginning three paragraphs of a newswire article (Sandhaus, 2008).

Figure 1.1: A comparison of the conciseness of articles for different genres of reporting. In Figure 1.1a I show the first paragraph of an information extraction document from the MUC 3/4 corpus (Sundheim, 1992), while in 1.1b and 1.1c I show the lead paragraphs (respectively, first paragraph and first three paragraphs) of two different *New York Times* articles (Sandhaus, 2008). Newswire can be as concise and to-the-point (Figure 1.1b) as articles for information extraction (Figure 1.1a); we can think of these as being easily reportable articles. However, newswire can also be very difficult to identify as encompassing a single, reportable story (Figure 1.1c).

human annotators for an information extraction task, it will often be very targeted, domain specific and of different linguistic quality (c.f. §§ 3.3 and 3.3.2). This can be seen in Figure 1.1, which juxtaposes the starting portion of an information extraction document (Figure 1.1a) with the starting portions of two different newswire documents (1.1b and 1.1c). Notice that both Figure 1.1a and Figure 1.1b are concise and clearly report on a particular event—we can think of these as more like abstracts or summaries of the event: they describe the core information. In contrast, Figure 1.1c

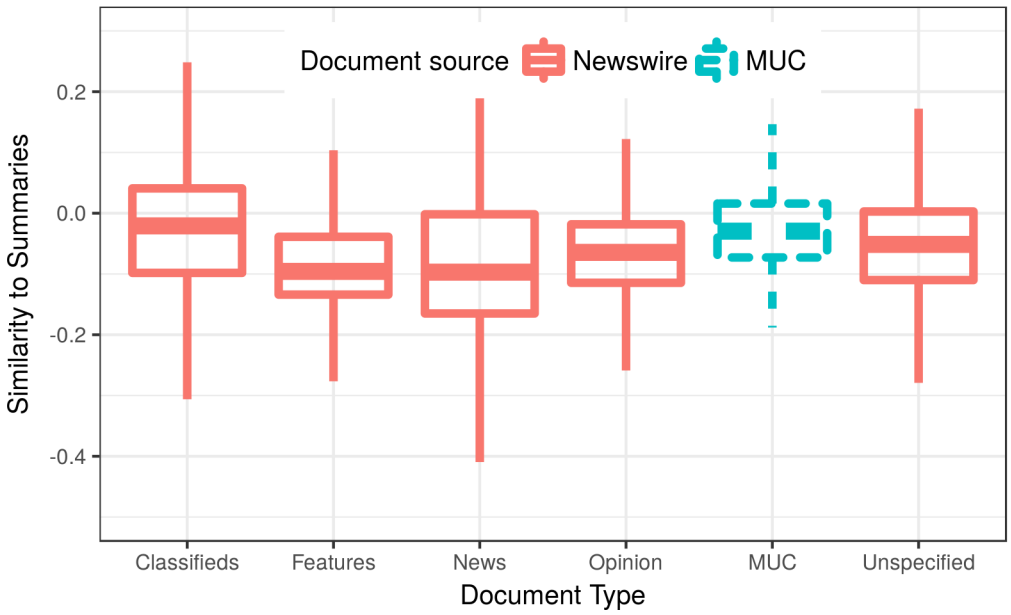


Figure 1.2: Summarized distribution of the overall average bias that certain types of newswire documents vs. MUC reports demonstrate in their verb usage. Bias is measured via the verb biases computed by Nye and Nenkova (2015); positive scores indicate a greater use in summary-/abstract-type documents, while negative scores indicate a greater use in full-length documents. Newswire types are directly from Sandhaus (2008); Ferraro et al. (2014); see chapter 4.

is arguably less reportable—the main point is less obvious, and it is not clear if the core information has been conveyed, even three paragraphs in.

Although it may seem eminently evident that, yes, task-oriented documents *are* different from newswire, I would like to highlight two specific ways in which they are. The general text I consider are *New York Times* articles, and the task-specific ones I consider are from the MUC-4 information extraction task (Sundheim, 1992), which focus on military-style reports of terrorism.

Verb Use

Under the assumption that verbs are a sufficient proxy for representing events, the first relevant way newswire and MUC are different is in what verbs, and the types of those verbs, they use. The *New York Times*, in meta-analysis pieces, provides advice on how to write newswire: Brown and Schluten (2012) writes that "sentences can even act as miniature narratives," while Hiltner (2017) notes certain sentences "for their clarity, their rhythm, their beauty and their enchantment." While some simple statistics may be hard pressed to quantify these qualitative descriptions, we can still examine some of these hinted at differences.[3]

Nye and Nenkova (2015) studied how verbs are used in articles and summaries of those articles. Using binomial tests to model how frequently verbs (specifically, their lemmas) appeared in articles and summaries, they derived summary biases for each verb lemma: a higher bias indicates the verb is more indicative of summaries. Many of the core, event-carrying MUC verbs—those that could singularly indicate what the MUC document is about, such as "kill," "murder," "kidnap," and "fight"—are biased toward summaries rather than full newswire. Meanwhile, verbs of reporting ("say," "tell"), belief ("believe", "think," "suppose") and (metaphorical) communication ("argue," "explain," "label," "blast") skew away from summaries and toward full articles.

[3]There are vast fields of study around narrative structure and how different types of text are written (Halliday and Hasan, 1976; Lehnert, 1981; Trabasso and Sperry, 1985; Graesser et al., 1997, 2002, i.a.). This analysis is intended as a scalable way to illuminate some differences between MUC and newswire; it is not meant as a replacement for deeper, more involved analysis.

In Figure 1.2 I use these biases to quantify how verbs are used differently in MUC (blue, dashed lines) and *New York Times* articles (Sandhaus, 2008, red, solid lines). The task of identifying verbs in these documents is described in greater detail in chapter 4. The NYT corpus also provides labels for a lot of the documents. While these labels can become very fine grained, they indicate if an article is an op-ed, a "hard" news piece, or a lengthier feature, among other possibilities. I separate out NYT documents by these labels to get a better sense of just how certain types of articles skew.

I compute a document's overall bias by simply averaging the biases of all verbs in the document. Though naïve, this method does yield noticeable differences: MUC documents have overall less variability and skew more toward summaries than newswire does. In particular, notice that while "news" types of newswire skews away from being like summaries, it has a very high variability.

Entities and Participants in Events

The second difference concerns the number of entities and how they interact with each other and different verbs.[4] Simply put, newswire documents have more entities than MUC documents. While this may not be in-and-of-itself surprising, the nature of the discrepancy is interesting.

Figure 1.3 reflects the cumulative density of the number of entities a document is

[4]An "entity" here roughly means a referential or Skolemizable (instantiable, c.f. §3.1) span of text. In practice, many noun phrase spans help form valid entities.

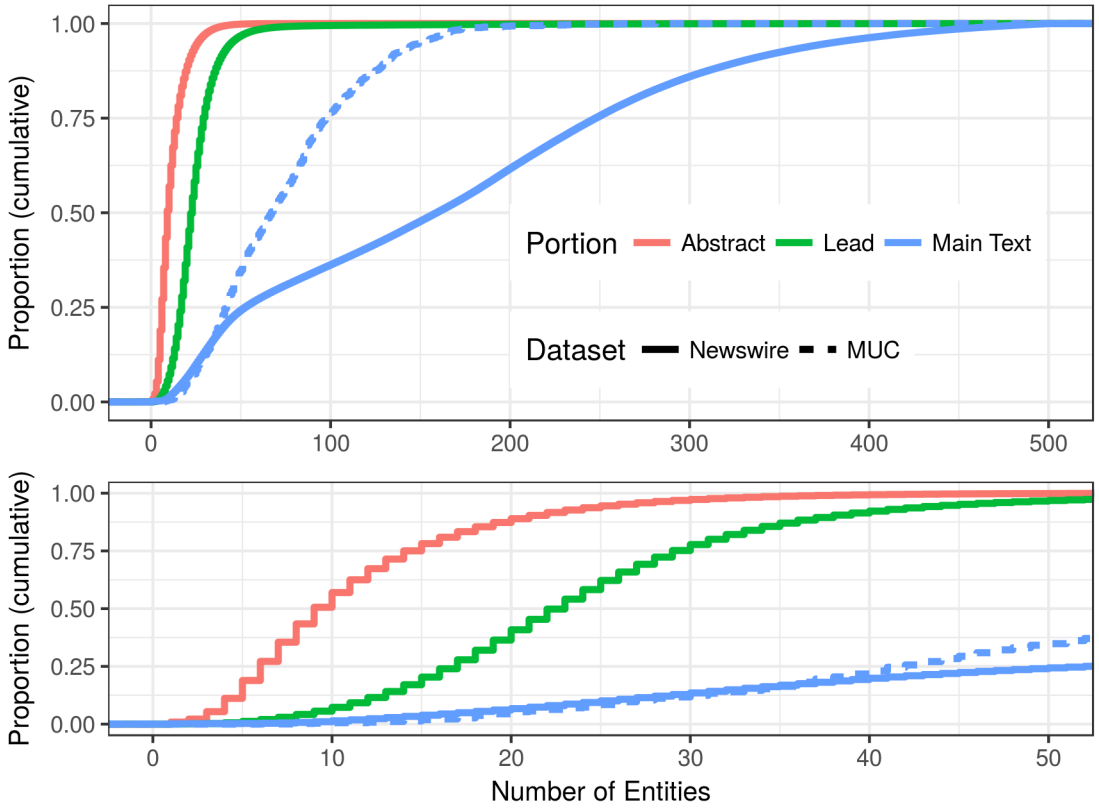


Figure 1.3: The cumulative number of entities per document, broken across source (newswire or MUC) and, for newswire, type. The "lead" is often the first two or three paragraphs of the full document (the "main text"), while the abstract is often a summary written separately from the text.

likely to have. As above, the task of identifying entities within these documents is described in chapter 4. First consider the number of entities in a complete newswire (NYT, (Sandhaus, 2008)) document (solid blue) vs. a MUC document (dashed blue): for small numbers of entities ($\leq$ 40), the two document sources behave similarly. However, notice that the newswire distribution has a larger tail: it is more likely to more than double the number of entities.

In addition to containing labels, the Sandhaus corpus also highlights some paragraphs as either an abstract/summary, or as the lead paragraphs. Using these distinctions, we can further analyze how newswire entities arise. The lead paragraphs are often part of the full text, while the abstracts are not.

1.3 Roadmap and Contributions

In the previous section, I demonstrated some of the issues that can arise when trying to leverage "general" knowledge to improve a representation (or understanding) of knowledge in a more "specific" or targeted area. These challenges, among others, will arise throughout this thesis.

In **Chapters 2 and 3** I begin with background summaries of the machine learning and linguistic literature, respectively, that is most relevant to this thesis. In the former, I cover basic probability, statistics, and four different ways of performing inference: expectation maximization, maximum a posteriori estimation, sampling, and variational inference. In the latter, I focus on theoretical and applied representations of events, including linguistic theories, computational-based theories, resources, and applications.

In **Chapter 4** I examine the Concretely Annotated Corpora resource (Ferraro et al., 2014, CAC), a large corpus of more than 15 million documents that have been automatically processed and annotated with different NLP tools. These annotations

are merged together into the same underlying, computer-readable schema called Concrete. I describe a confluence of three main factors that contribute to the benefits of Concrete, but also some of the difficulties involved in developing it and related resources. I explain this schema, and then analyze the event-relevant annotations in the CAC. While I played a significant role in the development and engineering of CAC and Concrete, I present it more as background that will help inform the rest of the thesis. However, the analyses in the latter portion are novel. I also hope that the explanations of the schema and data representation are useful to anyone who may wish to use CAC or Concrete.

In **Chapter 5** I begin the exploration into unsupervised form induction. I use the above CAC annotations, and in particular the multiple, overlapping frame annotations, in order to improve word embedding representations. Due to the nature of frames, I argue that the evaluation and analysis should be approached from an attribute-based perspective. I present a straight-forward generalization from the word embeddings community that easily accomodates many and varied frames, roles, and words analyzed by such. On three different comparative datasets—one from the NLP community which I helped create (Reisinger et al., 2015), and the other two from the cognitive science community (McRae et al., 2005; Vinson and Vigliocco, 2008)—I show that frames do yield lexical embeddings that encode stronger semantic expectations than do standard embeddings methods.

In **Chapter 6** I turn to analyzing sentences through memoized, (possibly) lexi-

calized syntactic frames. I present an EM-based algorithm that uses user-provided constraints to learn these larger frames; it outperforms other algorithms that provide similar styles of syntactic analysis when the amount of backoff smoothing is comparable, and overall it achieves competitive performance against multiple strong baselines. I also present a method for easily obtaining those constraints; while not explained in detail in this thesis, I have shown elsewhere that this constraint generation method can detect (un)grammaticality better than other strong syntactic and lexical methods (Ferraro et al., 2012a).

In **Chapters 7 and 8** I study structured document representations. Both present Bayesian models; chapter 7 presents a more theoretically-minded model that studies document understanding and modeling through hierarchical frames. This is the first unified model of probabilistic frames, encompassing syntactic, semantic, thematic and narrative elements. This unified, *generative* model outperforms a strong information extraction baseline on document modeling and overall semantic coherence. In chapter 8 I extend a frame-based model to allow for features or labels to be provided. The model builds off of recent advances from the neural networks community; and, when necessary, it can hypothesize values for missing features. I present two scalable (streaming) inference algorithms and study their effect on document understanding, lexical semantics, downstream classification.

Finally, I summarize and discuss future directions in Chapter 9.

Chapter 2

Background: Relevant Machine Learning

Developing automated systems of higher-level and "human-like" inference and reasoning have been core problems since the 1956 Dartmouth Conference ("Summer Research Project") on Artificial Intelligence (McCarthy et al., 2006). Since then there have two main approaches to developing systems encompassing such inference and reasoning: a logical, or symbolic, approach, and a statistically-informed approach.[1]

[1]Symbolic approaches generally build off of subsets of first-order logic and are categorically expressive (e.g., rule-based expert systems); statistically-informed approaches, such as Bayesian networks or neural networks, are mathematical models defined by noisy and incomplete observations that tend to sacrifice concision and expressiveness. What statistical approaches lose in expressiveness they often make up in robustness, as symbolic approaches generally are deterministic, heavily influenced or created by humans, and brittle (Russell and Norvig, 2010).

Of course, there is not a strict dichotomy between the two approaches. Deduction-based logical languages, such as Prolog (Colmerauer and Roussel, 1996) and its derivatives (Gallaire et al., 1984; Fuhr, 1995; Apt and Wallace, 2006, i.a.), can allow for certain weighted inference. Various kinds of constraints and prior knowledge can be encoded in these systems, which allow for prior knowledge; these constraints can help direct inference, mitigate exponential costs, and reinforce the logical aspect of inference.

In this chapter I will cover some of the prerequisite mathematical, statistical and machine learning background. This will include an overview of probabilistic graphical models, in particular directed Bayesian models, and a number of different learning and inference algorithms. Though this chapter's contents will be used throughout the rest of the thesis, a general familarity with the content will be useful in the following chapter (3), when I compare both symbolic or knowledge-aware approaches to representing meaning with statistically-informed approaches.

2.1 Probability and Basic Statistics

A typical machine learning recipe involves learning a collection of parameters $\Theta = \{\theta_j\}$ given N input items x_i, $1 \leq i \leq N$. We statistically model these data using the n-ary joint distribution $f^{(N)}$ as

$$x_1, \ldots, x_N \sim f^{(N)}(\cdot\ ;\ \Theta).$$

In a Bayesian setting, as in most (but not all) of this thesis, we imbue the parameters Θ with their own **prior** distribution g and hyperparameters α, arriving at the generative

story

$$\Theta \sim g(\cdot \mid \alpha)$$
$$x_1, \ldots, x_N \mid \Theta \sim f^{(N)}(\cdot \mid \Theta).$$

We often assume mutual (conditional) independence among the items x_i, i.e., that the joint distribution $f^{(N)}(x_1, \ldots, x_N)$ is equal to the product of N identical marginals $\prod_i f(x_i)$. This lets us write the story more simply as[2]

$$\Theta \sim g(\cdot \mid \alpha)$$
$$x_i \mid \Theta \sim f(\cdot \mid \Theta), \quad 1 \leq i \leq N.$$

The primary goal in modeling is to **learn** values for the latent parameters Θ (or the distribution g and α) that optimize a data-dependent objective function $J_\Theta(\{x_i\})$. In a Bayesian setting, this results in needing to perform posterior inference and update the model's beliefs about g:

$$g(\Theta \mid \{x_i\}, \alpha) \propto f(\{x_i\} \mid \Theta) g(\Theta \mid \alpha).$$

As is often the case (and as will be throughout this thesis), this objective function is

[2]For consistency, I will commonly write $f(\cdot \mid \Theta)$ rather than $f(\cdot; \Theta)$—i.e., I will adopt the Bayesian notation even if technically there is no prior distribution.

some form of log-likelihood, be it the joint log-likelihood

$$J_\Theta(\{x_i\}) = \sum_i \log f(x_i, \Theta),$$

or the marginal log-likelihood

$$J_\Theta(\{x_i\}) = \sum_i \log f(x_i) = \int_\Theta d\Theta \sum_i \log f(x_i \mid \Theta) g(\Theta \mid \alpha).$$

Thus while any *appropriate* distributions g and f can technically be used, we restrict ourselves to those that let us accomplish our inference goal with (relative) ease. This thesis makes heavy use of *exponential family* models—a certain class of computationally tractable distributions with a variety of nice properties.

2.1.1 Exponential Family Form

We say f is a member of an exponential family if it can be written in the form

$$f\left(x_i \mid \Theta\right) = h\left(x_i\right) \exp\left(\eta\left(\Theta\right) \cdot \chi\left(x_i\right) - A\left(\eta\left(\Theta\right)\right)\right), \tag{2.1}$$

where $\eta\left(\cdot\right)$ is a vector of parameters, $\chi\left(X\right)$ is a vector of sufficient statistics, the support function $h(X)$ restricts the distribution to a prespecified support, and $A(\eta\left(\cdot\right))$ is the log partition function. That is, $\exp A(\eta\left(\cdot\right))$ is what forces f, for all valid inputs, to be a proper distribution, while h specifies what is valid input. Notice that the log

partition function is independent of any observed data, while the support (measure) function h cannot be reparametrized with changes to Θ.

Exponential families exhibit a number of useful properties that significantly simplify computation and inference. The first is that the expected value of the sufficient statistic is equal to the gradient of the log partition function (Bickel and Doksum, 2006):

$$\mathbb{E}_f\left[\chi\left(X\right)\right] = \nabla_{\eta(\Theta)} A\left(\eta\left(\Theta\right)\right). \tag{2.2}$$

A second useful property is that the Hessian of the log partition function is the Fisher information matrix:

$$\nabla_\eta(\Theta)^2 A(\eta(\Theta)) = \mathbb{E}_f\left[\nabla_\Theta \log f(X \mid \Theta) \nabla_\Theta \log f(X \mid \Theta)^\intercal \mid \Theta\right].$$

A third property is that of *conjugacy*: a distribution q is the **conjugate prior** of a distribution p if, for $\theta \sim q(\cdot)$ and $x \mid \theta \sim p(\cdot \mid \theta)$, the posterior distribution of θ, $q(\theta \mid x)$, is of the same family as the prior distribution. Every exponential family has *some* conjugate prior (Bickel and Doksum, 2006).[3]

I list three common, exponential family distributions that feature heavily in this thesis in Table 2.1. However, both Dirichlet and Categorical distributions are central to examples in this chapter. I cover them explicitly below.

[3]This can be seen by augmenting the prior's sufficient statistics with the likelihood's negated log partition function, and by correspondingly augmenting the prior's natural parameter vector with one additional coordinate.

	Dirichlet	**Categorical**	**Univariate Gaussian**
Support	$\boldsymbol{\theta} \in \Delta^{K-1}$	$\mathbf{x} \in \{0,1\}^{\|e\|}$	$x \in \mathbb{R}$
Standard Parameters	$\boldsymbol{\alpha} \in \mathbb{R}_+^K$	$\boldsymbol{\theta} \in \Delta^{K-1}$	$\mu, \sigma \in \mathbb{R}$
Mass/Density	$\frac{\Gamma(\sum \alpha_i)}{\prod \Gamma(\alpha_i)} \prod x_i^{\alpha_i - 1}$	$\prod p_i^{1[x=i]}$	$\frac{\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}{\sigma\sqrt{2\pi}}$
Natural Parameters	$\boldsymbol{\alpha} - \mathbf{1}$	$\log \frac{\boldsymbol{\theta}}{\theta_K}$	$\begin{pmatrix} \frac{\mu}{\sigma^2} \\ \frac{-1}{2\sigma^2} \end{pmatrix}$
Sufficient Statistics	$\log \mathbf{x}$	$\left(\ 1[x_j = i] \ \right)_i$	$\begin{pmatrix} x \\ x^2 \end{pmatrix}$
Log Partition	$\sum_k \log \Gamma(\alpha_i) - \log \Gamma\left(\sum_i \alpha_i\right)$	$-\log\left(1 - \sum_{i<K} p_i\right)$	$\frac{\mu}{2\sigma^2} + \log \sigma$

Table 2.1: Common probability distributions relevant to this thesis.

Categorical/Multinomial

Given a discrete support $(S_i)_{i=1}^K$, if $X|\theta \sim \text{Cat}(\theta)$, then assuming that θ is a point on the $K-1$ simplex,

$$p(X;\theta) = \theta_X = \prod_{i=1}^{K} \theta_i^{1[X=k]} \tag{2.3}$$

$$= h(X) \exp\left\{ \begin{pmatrix} \log \frac{\theta_1}{\theta_K} \\ \log \frac{\theta_2}{\theta_K} \\ \cdots \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1[X = S_1] \\ 1[X = S_2] \\ \cdots \\ 1[X = S_K] \end{pmatrix} - \log\left(\sum_i \exp \theta_i\right) \right\} \tag{2.4}$$

where $1[q] = 1$ iff q is true (and 0 otherwise). Here, $h(X)$ ensures that X references a valid item of the support, while $\eta(\theta) = \log \frac{\theta}{\theta_K}$ and the sufficient statistic $\chi(X)$ is a

one-hot vector of length K (a vector with $K-1$ zeros and one 1).[4]

Dirichlet Distribution

Dirichlet distributions are continuous distributions over the probability simplex Δ^{K-1}: they are parametrized by positive vectors of size K and draws $\theta|\beta \sim \mathrm{Dir}(\beta)$ are K-length distributions. That is, each coordinate $\theta_i \geq 0$ and $\sum_i \theta_i = 1$. The Dirichlet density is given by

$$p(\theta;\beta) = \frac{\Gamma\left(\sum_{i=1}^{K}\beta_i\right)}{\prod_{i=1}^{K}\Gamma(\beta_i)}\prod_{i=1}^{K}\theta_i^{\beta_i-1} \tag{2.5}$$

$$= h(\theta)\exp\left\{\begin{pmatrix}\beta_1-1\\ \dots \\ \beta_K-1\end{pmatrix}\cdot\begin{pmatrix}\log\theta_1\\ \dots \\ \log\theta_K\end{pmatrix} + \left(\sum_{i=1}^{K}\log\Gamma(\beta_i) - \log\Gamma(\sum_{i=1}^{K}\beta_i)\right)\right\} \tag{2.6}$$

$$= h(\theta)\exp\{\eta(\beta)\cdot\chi(\theta) - A(\eta(\beta))\}, \tag{2.7}$$

where $\Gamma(\cdot)$ is the Gamma function, a generalization of the factorial function to real (complex) numbers. Here $h(\theta)$ is the base measure: it is 1 if and only if θ is a point on the $K-1$ simplex Δ^{K-1}.

[4]This properly defines an exponential family Categorical distribution. An alternative, and rather common, form simply sets $\eta(\theta) = \log\theta$, with $A(\eta(\theta)) = 0$. Technically this second formulation forms a *curved* exponential family. Despite the moniker, a curved exponential family can lose properties of a "true" exponential family, such as $\mathbb{E}_f[\chi(X)] = \nabla_{\eta(\theta)}A(\eta(\theta))$. The issue is that while there are K parameters, we only need $K-1$ to fully describe the distribution (Bickel and Doksum, 2006).

In exponential family form, it is easy to calculate a quantity such as entropy:

$$\begin{aligned} & H(p \sim \operatorname{Dir}(\cdot \| \beta)) \\ & = \mathbb{E}_p\left[\eta(\beta)\right] \cdot \mathbb{E}_p\left[\chi(\theta)\right] - \mathbb{E}_p\left[A(\eta(\beta))\right] \\ & = \eta(\beta) \cdot \nabla_{\eta(\beta)} A(\eta(\beta)) - A(\eta(\beta)) \end{aligned} \tag{2.8}$$

The hyperparameters β control the shape of the distribution. Smaller components result in lower entropy (draws tend to be closer to vertices of the simplex) while larger components result in higher entropy.

Finally, it is easy to show that the Dirichlet is the conjugate prior of the Categorical. With $\theta \sim \operatorname{Dir}(\beta)$ and N conditionally independent $X_i \sim \operatorname{Cat}(\theta)$

$$\begin{aligned} q(\theta \mid \{X_i\}) \propto \; & q(\theta) \prod_i p(X_i \mid \theta) \\ \propto \; & h_{\mathrm{Dir}}(\theta) \exp(\eta_{\mathrm{Dir}}(\beta)^\intercal \log \theta - A_{\mathrm{Dir}}(\eta_{\mathrm{Dir}}(\beta)) + \\ & (\sum_i \chi(x_i))^\intercal \log \frac{\theta}{\theta_K} - N A_{\mathrm{Cat}}(\eta_{\mathrm{Cat}}(\theta))) \\ = \; & \operatorname{Dir}\left(\cdot \mid \; \eta_{\mathrm{Dir}}(\beta) + \sum_i \chi(x_i)\right). \end{aligned}$$

Dirichlet-Multinomial Compound

Given the hierarchical model $\theta \sim \operatorname{Dir}(\beta) \in \Delta^{(K-1)}$ and $\mathbf{z}$, a collection of conditionally independent K-dimensional discrete/Categorical variables distributed according to $\operatorname{Cat}(\theta)$, what can we say about the prior predictive distribution, $p_\beta(\mathbf{z})$, where we

marginalize out θ?

Just as the Dirichlet being the conjugate prior to the Categorical let us analytically derive the posterior for θ, so does this conjugacy let us analytically derive the prior predictive (also called compound) distribution. Let $c(k)$ be the number of zs with value k; then the joint probability of $\mathbf{z}$ is given by the Dirichlet-Multinomial compound distribution $\mathrm{DMC}(\mathbf{z}|\beta)$:

$$\begin{aligned} p_\beta(\mathbf{z}) &= \int_\theta p(\mathbf{z} \mid \theta) p_\beta(\theta) d\theta \\ &= \frac{\Gamma\left(\sum_k \beta_k\right)}{\prod_k \Gamma(\beta_k)} \int_\theta \prod_k \theta_k^{c(k)} \prod_k \theta_k^{\beta_k - 1} d\theta \\ &= \frac{\Gamma\left(\sum_k \beta_k\right)}{\Gamma\left(\sum_k (c(k) + \beta_k)\right)} \prod_k \frac{\Gamma(c(k) + \beta_k)}{\Gamma(\beta_k)} \qquad (2.9) \\ &= \mathrm{DMC}(\mathbf{z}|\beta). \qquad (2.10) \end{aligned}$$

This compound can be generalized to a mixture scenario, where each z_i is stochastically generated by any of M different distributions. Specifically, given a collection of M Dirichlet samples $\theta_m \overset{i.i.d}{\sim} \mathrm{Dir}(\beta)$ and discrete indicator variables y_i, if $z_i \mid y_i, \theta \overset{i.i.d}{\sim} \mathrm{Cat}(\theta_{y_i})$, then we can consider the collection $[\mathbf{z}]_{\mathbf{y}=m}$ — only those z_i such that $y_i = m$. With $c(m, k)$ being the number of z_i with value k whose corre-

sponding $y_i = m$,

$$p_\beta(\mathbf{z}; \mathbf{y}) = \prod_{m=1}^{M} \left(\mathrm{DMC}\left([\mathbf{z}]_{\mathbf{y}=m} \,|\beta \right) \right) \tag{2.11}$$

$$= \prod_{m=1}^{M} \left(\frac{\Gamma\left(\sum_k \beta_k\right)}{\Gamma\left(\sum_k \left(c(m,k) + \beta_k\right)\right)} \prod_k \frac{\Gamma\left(c(m,k) + \beta_k\right)}{\Gamma\left(\beta_k\right)} \right). \tag{2.12}$$

Note that, in both cases ((2.10) and (2.12)), integrating out θ has removed the conditional independence of all z_i. Thankfully, though, we only must maintain summary histograms.

2.2 Graphical Models

Natural language processing problems often contain hundreds of thousands each of observed and latent variables, with millions of parameters to learn. Unless operating with very liberal independence assumptions, working with the joint distribution quickly becomes intractable. Graphical models provide a number of options for tractably handling these formulations (Bishop, 2006; Koller and Friedman, 2009).

The main idea behind graphical models is to associate properties about the joint distribution of random variables with graph-theoretic concepts. First, we associate every random variable in $\mathbf{X} = \{X_i\}$ with a node in a graph. Then, given a sufficient (topological) ordering over the variables $\{X_\cdot\}_<$, any joint distribution over $\mathbf{X}$

Generate K topics $\psi_k|\beta \sim \text{Dir}(\beta)$, for $k = 1 \dots K$

Generate topic usage priors for every document d $\theta_d|\alpha \sim \text{Dir}(\alpha)$

Assign each token i in each document d a topic $z_{d,i}|\theta_d \sim \text{Cat}(\theta_d)$

Generate each token i in each document d $w_{d,i}|z_{d,i}, \psi \sim \text{Cat}\left(\psi_{z_{d,i}}\right)$

Figure 2.1: The generative story for Latent Dirichlet Allocation (Blei et al., 2003).

factorizes into a product of local conditional distributions,

$$f(\mathbf{X}) = \prod_i f\left(X_i \mid \{X_j\}_{j<i}\right), \tag{2.13}$$

the goal is to find conditional independence properties among the nodes for X_i and subsets of $\{X_j\}_{j<i}$. If dependencies between any two random variables are indicated via an edge connecting the corresponding nodes in the graph, then by examining the graph we can easily gain insight into the underlying probabilistic model.

One type of graphical model—Bayesian networks—feature heavily in this thesis. In a Bayesian network, a directed edge from X_i to X_j indicates a probabilistic dependence relation; in equation (2.13), the given set $\{X_j\}$ is just the parents of X_i. Simple criteria exist to easily test for conditional independence (Pearl (1988); Bishop (2006)), and the graph topology clearly displays causal processes and relations among random variables.[5]

[5]Markov random fields (MRF) are another main type of graphical model. MRFs are undirected graphs; due to this, rather than talking about parents of nodes, we talk about maximal cliques of nodes. A simple "blocked" condition tests for conditional independence; we rewrite (2.13) as a

Example 2.1: Topic Models

Latent Dirichlet Allocation (Blei et al., 2003, LDA) is a well-known Bayesian network that models unigram word counts in documents as stochastic (ad)mixtures of different "topics"—distributions over the entire vocabulary. Provided each topic ends up forming a proper probability distribution, each can reweight words, thereby reassigning importance. The full generative story for LDA is shown in Figure 2.1: to learn K topics over a V-sized vocabulary, the topics ψ_k are drawn according to a Dirichlet parametrized by vocabulary hyperparameters $\beta \in \mathbb{R}_V^+$. Each document generates its own convex weighting θ_d from a Dirichlet parametrized by $\alpha \in \mathbb{R}_K^+$. Given the topic proportions, each individual word (token) i in a document d is softly assigned to a specific topic, $z_{d,i} \sim \text{Cat}(\theta_d)$; finally, the observed word form $w_{d,i}$ is drawn from that topic, $w_{d,i} \sim \text{Cat}\left(\psi_{z_{d,i}}\right)$.

The per-document topic proportions θ_d and topics ψ_k are elements of the $K-1$ and $V-1$ simplexes Δ^{K-1} and Δ^{V-1}. That is, they are proper discrete distributions over K and V elements, respectively: each coordinate $\theta_{d,t}$ and $\psi_{k,v}$ must be between 0 and 1, and θ_d and ψ_k must each sum to 1. Recall from §2.1 that the Dirichlet hyperparameters β and α control the *a priori* shape of the topics and proportions, respectively. In practice, they can be optimized on development data (Wallach, 2008), injected with domain specific

product of scores over cliques, $f(\mathbf{X}) \propto \prod_C \psi_C(\mathbf{X}_C)$, where each C is a maximal clique, $\mathbf{X}_C$ are those nodes in C and $\psi_C(\cdot) \geq 0$ is a *potential function* that scores $\mathbf{X}_C$. ψ_C does not need to be a probability density/mass function.

information (Paul and Dredze, 2012), or tuned toward specific tasks (Mimno and McCallum, 2008; Ramage et al., 2009).

Having observed each document's words (word counts), the goal is to perform posterior inference $p(\psi, \theta, z|w, \beta, \alpha)$. Posterior inference is intractable for LDA; we must instead use a tractable approximation.

2.3 Inference Techniques

In this section I examine three methods for performing posterior inference: maximum a posteriori, sampling, and variational inference. All three share the goal of arriving at tractable techniques for computing the posterior distribution, $g(\Theta \mid \{x_i\}, \alpha) \propto f(\{x_i\} \mid \Theta) g(\Theta \mid \alpha)$.

2.3.1 Maximum A Posteriori (Maximum Likelihood)

Maximum a posteriori (MAP) formulates inference as an optimization problem that finds the values $\Theta^\star$ that maximize the posterior. The logarithm being monotonic, MAP inference generally optimizes the log of the posterior up to a constant, $\tilde{g}$:

$$\Theta^\star = \arg\max_{\Theta} \log \tilde{g}(\Theta \mid \{x_i\}, \alpha) = \log f(\{x_i\} \mid \Theta) + \log g(\Theta \mid \alpha).$$

Note that MAP estimation specifically searches for a single value of Θ—despite Bayesian statistics allowing us to quantify and describe uncertainty about Θ in principled ways.

Of course, if this uncertainty is uninformative (i.e., we do not have any preference for one value of Θ vs. another), MAP estimation *effectively* becomes regularized maximum likelihood estimation. Maximum likelihood estimation (MLE) seeks $\Theta^\star$ that optimizes the (log) likelihood of the observed data: $\Theta^\star = \arg\max_\Theta \log f(\{x_i\} \mid \Theta)$. Though MLE estimates can be effective in practice for large data, they still can suffer sparsity issues. Regularized MLE is a compromise: it uses the notion of a modulating force on Θ, though without imposing distribution requirements.

The specifics of MAP (MLE) inference depend greatly on the particular problem: what, if any, constraints are there on Θ? Is the log posterior (up to a constant) a convex function, i.e., is $\tilde{g}(a\Theta_1 + (1-a)\Theta_2) \geq a\tilde{g}(\Theta_1) + (1-a)\tilde{g}(\Theta_2)$, for $a \in (0,1)$? If so, then any optimizing $\Theta^\star$ will be no better a solution than any other optimizing values (Boyd and Vandenberghe, 2004). Is the log posterior (sub-)differentiable, such that we can compute and follow the gradient? If so, then we have a well-understood basic recipe for how to perform inference (covered more in §2.4.1).

Example 2.2: A Log-linear Language Model

Consider a basic log-linear language model of the form $p_{\boldsymbol{\theta}}(v \mid h) \propto \exp(\boldsymbol{\theta} \cdot \mathbf{f}(h, v))$. This is a general exponential family model, with K-dimensional sufficient statistics (feature vector) $\mathbf{f}$. To limit overfitting, we can treat each coordinate θ_k as drawn from a univariate Gaussian distribution of

zero mean and variance σ^2. (The analogous non-Bayesian treatment involves subtracting an $\ell - 2$ regularizer on $\boldsymbol{\theta}$,

$$R(\boldsymbol{\theta}) = C \sum_k \theta_k^2, \tag{2.14}$$

where $C = 2\sigma^{-2}$.) Given joint cooccurrence counts $c(h, v)$, MAP inference optimizes

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} \sum_{h,v} c(h, v) \log p_{\boldsymbol{\theta}} (v \mid h) + \sum_k \log \text{Normal}(\theta_k; 0, \sigma^2),$$

while MLE optimizes

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} \sum_{h,v} c(h, v) \log p_{\boldsymbol{\theta}} (v \mid h) + C \sum_k \theta_k^2.$$

In both cases the objective is convex and differentiable with respect to $\boldsymbol{\theta}$, leaving us with the standard and well-known partial derivative (of the MAP objective)

$$\frac{\partial}{\partial \theta_k} = \sum_{h,v} c(h, v) \mathbf{f}(h, v)_k - \sum_{h,v} \sum_w p_{\boldsymbol{\theta}} (w \mid h) \, \mathbf{f}(h, w)_k - \frac{1}{2\sigma^2} \theta_k. \tag{2.15}$$

Though exponential family distributions' densities, parametrized with their *natural* parameters, are convex (Bickel and Doksum, 2006), hierarchical *models*—even

composed of exponential family distributions—with latent variables as considered in this thesis, generally do not from convex objectives. Non-convex objectives yield local, rather than global, optima. MAP/MLE inference for non-convex objectives uses variants of the **expectation maximization** algorithm (Dempster et al., 1977). The EM algorithm is composed of two basic steps. First the E-step uses current parameter estimates to compute the log-likelihood averaged over latent variables. Second the M-step optimizes the just-computed averaged log-likelihood to re-estimate the parameters.

Although in this thesis I do not study HMMs directly, I do perform probabilistic inference over trees. Performing inference in an HMM can be considered a special case of inference in trees (the linear chain is just a right branching tree); inference issues that occur with HMMs occur in trees as well. Therefore, studying HMM inference here provides a high-level introduction to inference in trees.

Example: Hidden Markov Models

Consider a standard, discrete hidden Markov Model (HMM) with K hidden states: at timestep t ($1 \leq t \leq T$), the Categorical observation x_t is drawn from a particular emission model, $\text{Cat}(\phi_{z_t})$, where the selection is determined by a first-order Markov transition distribution, $z_t \sim \text{Cat}\left(\theta_{z_{t-1}}\right)$. All $2K$ θ_k and ϕ_k parameter collections are multinomial parameters of sizes K and the vocabulary, respectively. Though a full derivation is beyond the scope of this chapter, we estimate these parameters with the Baum-Welch algorithm, an

iterative EM approach, from the observed sequence $x_1 \dots x_T$

$$\{\theta_k^\star\}, \{\phi_k^\star\} = \arg\max_{\theta,\phi} \mathbb{E}\left[\log p(x_1 \dots x_T)\right]. \tag{2.16}$$

This optimization involves marginalizing over *just* the sequence of latent states $z_1 \dots z_T$, *holding fixed* the current parameters. In the E-step at iteration i, we use our current estimates $\theta^{(i-1)}$ and $\phi^{(i-1)}$ to compute expected joint state-state counts $c_{k,k'}$ and state-observation counts $c_{k,w}$ from the expected log-likelihood of the observed sequence. For the discrete HMM, the M-step completes the optimization by (conditionally) renormalizing these acquired expected counts to get updated estimates $\theta^{(i)}$ and $\phi^{(i)}$. See Jurafsky and Martin (2008) for additional details.

Notice that the above HMM estimation is a maximum likelihood estimate. Treating the parameters as random variables significantly complicates the expectation in Eq. (2.16). We will consider in the next section how to generally handle this problem. First though, let's look at another example.

Example: Topic Models

MAP inference for Bayesian topic models are afflicted with the same issues as MAP inference for Bayesian HMMs. However, non-Bayesian topic models, or probabilistic latent semantic analysis, readily yield maximum likelihood estimates (Hofmann, 1999). Without the priors complicating the expectation, EM updates can easily be derived; the E-step computes the intermediate topic

assignment posterior

$$p(z_{d,i} \mid \ d, w_{d,i}) \propto p(z_{d,i})p(d \mid \ z_{d,i})p(w_{d,i} \mid \ z_{d,i}),$$

which is used to compute standard discrete expected counts c and optima, e.g.,

$$p(w_{d,i} \mid \ z_{d,i}) \propto \sum_{d} c(d, w)p(z_{d,i} \mid \ d, w_{d,i}).$$

Of course, some researchers *have* provided MAP inference algorithms, but they have relied on either reparametrizing the model or pre-marginalizing the topic assignments (Chien and Wu, 2008; Soufifar et al., 2011; Taddy, 2012; Chen et al., 2015; May et al., 2015).

2.3.2 Variational Inference

In the previous section we considered two applications of point estimation. However, in the latter, although we arrived at easy (and rather intuitive) update formulas, we did so by omitting the Bayesian aspect. What if, due to philosophical or practical concerns, we wanted to maintain the principled way of incorporating prior beliefs?

Variational inference is a general technique for approximating a complex, intractible posterior $p(\Theta \mid \{x_i\})$. In variational methods, we specify an entire *family* of distributions $\mathcal{Q}$, and try to find the $q_\phi \in \mathcal{Q}$ that is "closest" to the true model. We

measure closeness by KL divergence, optimizing the objective

$$\mathcal{L}_{q_\phi}\left(\{x_i\}\right) = -\mathrm{D}_{\mathrm{KL}}\left(q_\phi\left(\Theta\right) \| p\left(\Theta, \{x_i\}\right)\right) = \mathbb{E}_{q_\phi(\Theta)}\left[\log \frac{p\left(\Theta, \{x_i\}\right)}{q_\phi\left(\Theta\right)}\right]. \tag{2.17}$$

Because KL divergence is non-negative, if we exactly match q_ϕ to p, then equation (2.17) is minimized at $\mathcal{L}_{q_\phi} = 0$. We directly change this objective by optimizing Eq. (2.17) with respect to the **variational parameters** ϕ. By computing the gradient of Eq. (2.17) with respect to ϕ, $\nabla_\phi \mathcal{L}_{q_\phi}\left(\{x_i\}\right)$, we can use gradient ascent methods to numerically optimize Eq. (2.17), or we can directly set the gradient equal to 0 and analytically solve it (Blei et al., 2003; Hoffman et al., 2013). It's important to note that when we optimize Eq. (2.17), we are optimizing the variational distribution $q_\phi\left(\Theta\right)$, which acts as a proxy for the true posterior. Variational methods, in general, do not allow us to say anything about p itself; typically, whenever the true posterior is needed, we must use the optimized $q_\phi\left(\Theta\right)$. Variational methods also generally do not have any asymptotic guarantees (though in practice they tend to be fast to run).

Eq. (2.17) is called the ELBO—the **E**vidence **L**ower **Bo**und; it is a proxy objective

for the marginal data log-likelihood:

$$\begin{aligned}
\log p\left(\{x_i\}\right) &= \log \int_{\Theta} p\left(\{x_i\} \mid \Theta\right) p\left(\Theta\right) d\Theta && (2.18)\\
&= \log \int_{\Theta} \underbrace{\frac{q_\phi\left(\Theta\right)}{q_\phi\left(\Theta\right)}}_{=1} p\left(\{x_i\} \mid \Theta\right) p\left(\Theta\right) d\Theta && (2.19)\\
&= \log \mathbb{E}_{q_\phi(\Theta)}\left[\frac{p\left(\{x_i\} \mid \Theta\right) p\left(\Theta\right)}{q_\phi\left(\Theta\right)}\right] && (2.20)\\
&\geq \mathbb{E}_{q_\phi(\Theta)}\left[\log p\left(\Theta, \{x_i\}\right) - \log q_\phi\left(\Theta\right)\right] && (2.21)\\
&= \mathcal{L}_{q_\phi}. && (2.22)
\end{aligned}$$

The last step follows from Jensen's Inequality, a well-known calculus inequality (MacKay, 2003). Given a convex (down) function $f(x)$ and $\alpha \in [0, 1]$, Jensen's inequality states that for any two points x_0 and x_1 in the domain of f, the value of f at the interpolation of those points will never be less than the interpolation of f applied to those two points. Given $\boldsymbol{\alpha} \in \Delta^{K-1}$, Jensen's Inequality generalizes to K points:

$$f(\boldsymbol{\alpha}^\intercal \mathbf{x}) = f(\sum_k \alpha_k x_k) \geq \sum_k \alpha_k f(x_k). \tag{2.23}$$

In Figure 2.2, the blue dot represents the value of the interpolation $f(\alpha x_0 + (1-\alpha)x_1)$, while the green dot represents the interpolation of the function values $\alpha f(x_0) + (1 - \alpha) f(x_1)$.

The difficulties in variational methods occur when computing the expectations. Note that we have not required q to take any particular form; choosing an appropriate

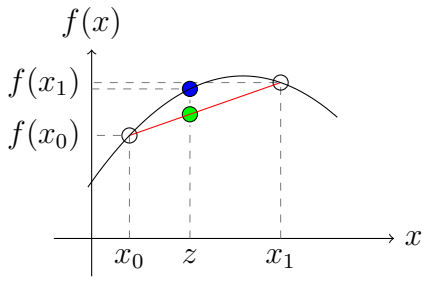


Figure 2.2: An illustration of Jensen's inequality in one dimension. Given the convex (concave) function $f(x)$ and $\alpha \in [0, 1]$, for any two points x_0 and x_1 in the domain of f, we compute $z = \alpha x_0 + (1 - \alpha)x_1$. Represent $f(z)$ with the blue dot and $\alpha f(x_0) + (1 - \alpha)f(x_1)$ with the green dot. Then $f(z) \geq \alpha f(x_0) + (1 - \alpha)f(x_1)$, i.e., the blue dot will never be less than the green dot.

factorization of q over the latent variables in p affects our ability to analytically compute the expectations. To circumvent as many difficulties as possible, researchers have typically relied heavily on two strong assumptions. These assumptions have nevertheless allowed significant progress.

The first assumption is that both the "true" model p and the approximate model q are constructed from appropriate conjugate exponential family distributions. Such pairs include Dirichlet priors for multinomial (Categorical/discrete) variables, Gaussian priors for fixed-covariance Gaussian variables, and inverse-Gamma priors for fixed-mean Gaussian variables. The second assumption is that q is the mean-field approximation: it fully, and independently, factorizes over all latent variables in Θ.

That is, each latent variable is independent[6]:

$$q_\phi(\Theta) = \prod_j q_{\phi_j}(\theta_j).$$

The mean field approximation makes individual expectations much "simpler" to compute. One such expectation is entropy, which now decomposes (and is iterated) according to any internal, distribution structure within components of Θ.

Some of the models in this thesis rely heavily on variational inference in Bayesian networks. Let's examine how it works with topic models.

Example: Topic Models

The standard variational approach, presented by Blei et al. (2003), is to use a mean field approximation $q(\{\theta_d\}, \{\psi_k\}, \{z_{d,i}\})$ that treats all latent variables as independent from one another. This removes the troublesome links we had when trying to derive a MAP EM algorithm.

Each latent parameter and variable is goverened by its own variational parameter. These variational parameters control the variational distribution for the latent parameter; for instance, every topic $\psi_k \in \Delta^{K-1}$ is governed by its own $\lambda_k \in \mathbb{R}^K$, each topic proportion θ_d is governed by its own γ_d, and every assignment $z_{d,i}$ is governed by its own multinomial $\phi_{d,i}$. The variational family has the form

$$\prod_k q(\psi_k|\lambda_k) \prod_d q(\theta_d|\gamma_d) \prod_i q(z_{d,i}|\phi_{d,i}).$$

[6]Recall that each component θ_j could itself be a vector-valued random variable.

To easily compute the entropy $H(q)$ and the expected log joint, each variational distribution q will be in the same exponential family as the corresponding distribution in the full model.

Combining the variational factorization with the definition of the ELBO, and noting in particular that the mean-field approximation allows us to *refine* the distributions with which we take expectations, we get

$$\begin{aligned}
&\sum_k \mathbb{E}_{q(\psi)}\left[\log p(\psi_k|\beta)\right] + \sum_d \mathbb{E}_{q(\theta)}\left[\log p(\theta_d|\alpha)\right] + \\
&\sum_{d,i} \mathbb{E}_{q(\theta)q(z)}\left[\log p(z_{d,i}|\theta_d)\right] + \sum_{d,i} \mathbb{E}_{q(\psi)q(z)}\left[\log p(w_{d,i}|z_{d,i},\psi)\right] - \\
&\sum_k \mathbb{E}_{q(\psi)}\left[\log q(\psi_k|\lambda_k)\right] + \sum_d \mathbb{E}_{q(\theta)}\left[\log q(\theta_d|\gamma_d)\right] + \\
&\sum_{d,i} \mathbb{E}_{q(z)}\left[\log q(z_{d,i}|\phi_{d,i})\right].
\end{aligned}$$

The conditionally conjugate exponential family forms make it easy to differentiate and analytically optimize the ELBO. For example,

$$\begin{aligned}
\mathbb{E}_{q(\psi)}\left[\log p(\psi_k|\beta)\right] &= \mathbb{E}_{q(\psi)}\left[\eta(\beta)^\intercal \log\psi_k - A(\eta(\beta))\right] \\
&= \mathbb{E}_{q(\psi)}\left[\eta(\beta)\right]^\intercal \mathbb{E}_{q(\psi)}\left[\log\psi_k\right] + \text{constant} \\
&= \eta(\beta)^\intercal \mathbb{E}_{q(\psi)}\left[\log\psi_k\right] + \text{constant} \\
&= \eta(\beta)^\intercal \nabla_{\eta(\lambda_k)} A\left(\eta\left(\lambda_k\right)\right) + \text{constant}.
\end{aligned}$$

The last step followed from (2.2): that for exponential families, the expected

value of the sufficient statistic is equal to the gradient of the log partition function. Applying the same type of approach to all summands of the ELBO, we can then differentiate with respect to the variational topics λ_k:

$$\nabla_{\lambda_k} L(q) = \eta(\beta)^\intercal \nabla^2_{\eta(\lambda_k)} A(\eta(\lambda_k)) - \eta(\lambda_k)^\intercal \nabla^2_{\eta(\lambda_k)} A(\eta(\lambda_k)) + \sum_{d,i} \phi_{d,i,k} t(w_{d,i})^\intercal \nabla^2_{\eta(\lambda_k)} A(\eta(\lambda_k)).$$

Factoring and setting the gradient to 0 implies

$$\lambda_k = \beta + \sum_{d,i} \phi_{d,i,k} t(w_{d,i}),$$

i.e., that the posterior variational topics are the expected number of times each vocabulary item is assigned to that topic, modulated by the hyperparameters β. Similar calculations hold for the other variational parameters, taking special care that the variational assignment parameters $\phi_{d,i}$ must be multinomial parameters, i.e., distributions—Lagrange multipliers easily handle this constraint.

For the special case when the variational distribution q is of the same form as p (i.e., when $p(z, \theta|x) \in \mathcal{Q}$), then computing expectations under q *is* inference in our model.

While variational inference is particularly applicable in conditionally conjugate models, there have been a number of efforts to variational inference in conditionally

non-conjugate models. These approaches range from using additional approxmations (Jaakkola and Jordan, 1997; Wang and Blei, 2013), reparametrizations (Kingma and Welling, 2014), and hybrid approaches using both sampling and optimization (Ranganath et al., 2014; Rezende et al., 2014). Second-order and cumulant approximations have also been found useful, both within variational inference and more general Bayesian inference (Barber and de van Laar, 1999; Smith and Eisner, 2006; Wang and Blei, 2013).

2.3.3 Markov Chain Monte Carlo

In the posterior inference we've been examining, a key difficulty is dealing with latent variables and couplings among those variables. Variational inference, and MAP EM, attempt to marginalize out the latent variables to get a tractable estimation algorithm. *Knowing* what those variables are would (generally) simplify matters; given values for all latent variables, the full joint distribution can easily be computed, so (up to a constant) the posterior can also be computed.

Markov Chain Monte Carlo (MCMC) techniques run with this idea of fully specifying latent variables. Of course, the true values cannot actually be known, so instead inference via sampling involves sampling values of the variables from a user-specified transition function. MCMC algorithms also tend to be easy (or easier) to derive and implement than MAP EM or variational algorithms. Under appropriate conditions,[7]

[7]Informally, there are two primary criteria (see Motwani and Raghavan (2010) for more formal definitions):

MCMC *does* come with asymptotic guarantees of eventual convergence. Moreover, this convergence is without regard to the initial variable settings. Unfortunately, MCMC can be slower to converge: these guarantees provide few practical guarantees.

Though there are many sampling techniques, like Metropolis-Hastings (Metropolis et al., 1953; Hastings, 1970; Motwani and Raghavan, 2010), Hamiltonian (Hybrid) Monte Carlo (Duane et al., 1987; Betancourt, 2017), and slice sampling (Neal, 2003), the one used in this thesis is Gibbs sampling (Geman and Geman, 1984), which samples variables' values from their conditional posterior distribution. In particular, I use a variant called collapsed Gibbs sampling, which analytically marginalizes out certain latent parameters prior to sampling (Liu, 1994; Griffiths and Steyvers, 2004).

2.3.3.1 Gibbs Sampling

Given our collection of latent parameters $\Theta = \{\theta_j\}_j$ with some preset values, Gibbs sampling iteratively samples new values θ_i from the posterior of θ_i, conditioned on the values of all other variables $\Theta \backslash \{\theta_i\}$. While sampling from the full conditional may sound daunting, we use the conditional independence properties of the probabilistic model to (hopefully) simplify the conditional. In particular, for directed models, each variable only needs to know the values of the variables in its Markov blanket $\pi(\theta_i)$: its parents, its children, and its childrens' parents (Koller and Friedman, 2009). Thus,

Irreducibility In the limit, it must be possible to get from one configuration of the variables to any other.
Aperiodicity In the limit, particular configurations of the variables must be able to occur at any time.

in iteration t, we sample

$$\theta_i^{(t)} \sim p(\cdot \mid \pi(\theta_i)).$$

The Markov blanket $\pi(\theta_i)$ may include variables that have already been updated in iteration t as well as those not yet resampled (i.e., with "old" values). Because Gibbs sampling requires us to sample from the full conditional, conditional conjugacy is important.

As mentioned, collapsed Gibbs sampling is Gibbs sampling after analytically marginalizing select variables $\dot{\Theta} \subset \Theta$ out. The parameter set becomes $\tilde{\Theta} = \Theta - \dot{\Theta}$. We sample

$$\tilde{\theta}_i^{(t)} \sim \int p(\cdot \mid \pi(\theta_i'), \dot{\Theta}) dp(\dot{\Theta}).$$

This marginalization directly affects the Markov blanket, coupling variables that may not have been coupled in the original model. This can necessitate additional modeling restrictions, as now, depending on what variables are collapsed out, greater care may be needed for (collapsed) conditional conjugacy. Empirically though, with intelligent collapsing, the increased complexity and coupling involved in collapsed Gibbs sampling results in faster converging samplers (Griffiths and Steyvers, 2004).

Example: Topic Models

Griffiths and Steyvers (2004) presented a collapsed Gibbs sampler for latent Dirichlet allocation in which they integrated out all topic proportions θ_d and topics ϕ_k. Their sampler thus relies on the (gated) Dirichlet-multinomial

compound distribution, Eq. (2.12), which I reproduce here:

$$p_\beta(\mathbf{z};\mathbf{y}) = \prod_{m=1}^{M}\left(\frac{\Gamma\left(\sum_k \beta_k\right)}{\Gamma\left(\sum_k \left(c(m,k)+\beta_k\right)\right)}\prod_k \frac{\Gamma\left(c(m,k)+\beta_k\right)}{\Gamma\left(\beta_k\right)}\right).$$

With the topics and topic proportions integrated out, the only variables to sample are the topic assignments $z_{d,i}$. Originally, the assignments were conditionally independent of one another, given the proportions θ_d; now that independence is gone. Even worse, by collapsing the topics, each assignment depends on assignments in other documents. That is, given all words $\mathbf{w}$, we will sample

$$z_{d,i} \sim p_{\alpha,\beta}(\cdot \mid \mathbf{z}\backslash\{z_{d,i}\};\mathbf{w}) = \frac{p_{\alpha,\beta}(\mathbf{z}\ ;\mathbf{w})}{p_{\alpha,\beta}(\ \mathbf{z}\backslash\{z_{d,i}\};\mathbf{w})}. \tag{2.24}$$

As discussed earlier though, this dependence manifests only through (gated) summary histograms.

Using the fact that Γ function is a generalization of factorial, we can use the property

$$\Gamma(x+1) = x\Gamma(x)$$

to show that Eq. (2.24) can be computed as

$$p_{\alpha,\beta}(z_{d,i}=k \mid \mathbf{z}\backslash\{z_{d,i}\};\mathbf{w}) \propto \left(c(d,k)+\alpha_k+1\right)\frac{c(k,w_{d,i})+\beta_{w_{d,i}}}{\sum_v c(k,v)+\beta_v}.$$

2.4 Gradient-Based Learning Algorithms for Optimizable Objectives

Throughout §2.3, I made a number of references to both optimizing and differentiating an objective. In this section, I survey a few fundamental gradient-based optimization routines. Fundamentally, they are all based on the notion of gradient ascent (§2.4.1): that is, iteratively making small steps (in the parameters) in the direction of the largest change of the objective function. The following sections first elaborate and define gradient ascent, and then expand on it as relates to scalability, speed of convergence, and adapting the algorithm to better optimizing probability distributions.

2.4.1 Gradient Ascent

Gradient ascent is a fundamental technique for optimizing differentiable functions. Given a function $J : \mathbb{R}^K \to \mathbb{R}$, gradient ascent solves the problem

$$\max_{\Theta \in \mathbb{R}^K} J(\Theta)$$

by refining an initial hypothesis $\Theta^{(0)}$ according to

$$\Theta^{(t)} = \Theta^{(t-1)} + \rho_t \nabla_\Theta J(\Theta) \mid_{\Theta^{(t-1)}}, \tag{2.25}$$

for $\rho_t \geq 0$. That is, gradient ascent forms a sequence of intermediate points where points are reoriented, both in direction and magnitude, by (a multiple of) the gradient at that point.

It is easy to see that Eq. (2.25) optimizes a first-order Taylor approximation to J, centered around $\Theta^{(t-1)}$:

$$J(\Theta) \approx J(\Theta^{(t-1)}) + \left(\Theta - \Theta^{(t-1)}\right)^\intercal \nabla_\Theta J(\Theta) \mid_{\Theta^{(t-1)}} + o(\|\Theta - \Theta^{(t-1)}\|).$$

Note that the residual second-order term, which uses the Euclidean (ℓ-2) norm effectively places a Euclidean constraint on the new parameters: the closeness, which should be small, of the new and old parameters must be measured according to $\sqrt{\sum_i (\Theta_i - \Theta_i^{(t-1)})^2}$.

For example, if we perform MAP inference with the log-linear model of Example 2.2, then we can simply evaluate Eq. (2.15) at the (full) current point $\Theta^{(t-1)}$ and shift each component of $\Theta^{(t-1)}$ by (a multiple of) its corresponding partial derivative. This is the standard way to optimize log-linear models.

On the other hand, gradient ascent is not a panacea—consider LDA, our example 2.1. In §2.3.2, we saw that the LDA ELBO gradient can be solved analytically. Empirically, this results in good convergence and subsequent learned models. Although we could apply gradient ascent with the ELBO's gradient, I have found in my own experiments that this can lead to numerical instability and very poor convergence.

2.4.2 Stochastic Gradient Ascent

In both of the above examples, computing the gradients involved acquiring statistics across the *entire* dataset. For large datasets, this can be taxing: (1) simply iterating over some datasets, such as described in chapter 4, can take hours; (2) the required computations for the entire dataset may not fit in memory; or (3) a lot of computation must be done before any progress is made, even if the computation used poor parameter estimates.

Stochastic gradient ascent (SGA) performs intermediate (potentially partial) updates based on a small sample, down to a single instance, of the available data (Robbins and Monro, 1951). In the case where we use a single instance per SGA update, we draw this element d uniformly at random from the dataset. Using this sampled data point, we compute the *stochastic gradient* based on d, $\nabla_\Theta J_d(\Theta)$. The update rule is

$$\Theta^{(t)} = \Theta^{(t-1)} + \rho_t \nabla_\Theta J_d(\Theta) \mid_{\Theta^{(t-1)}} .$$

This gradient is a noisy, but unbiased, estimate of the full gradient. In particular, there is no guarantee that the stochastic gradient will reflect the true direction (and magnitude) of steepest ascent. This highlights a tradeoff in SGA: sampling fewer datapoints at a time will (generally) decrease the computational costs, though the resulting computations and updates will be subject to greater variability; sampling more datapoints can help stabilize computations. We can sample a small number of

datapoints, called mini-batches, rather than just one element at a time, to address this tradeoff.

2.4.3 Tuning the Step Size

The (stochastic) gradient ascent update of Eq. (2.25) includes ρ_t, a way to rescale the gradient at each iteration. This rescaling (step size) can have an outsized impact on the efficacy of gradient optimization: a step size that is too large can cause the algorithm to diverge or oscillate, while one that is too small can take too long to converge.

Though we could attempt to pick the *optimal* step size, this problem is often difficult to solve, as it requires optimizing, with respect to ρ, $J(\Theta + \rho \nabla_\Theta J(\Theta))$. Note that this requires re-evaluating the gradient, possibly multiple times, just to pick a step size. It often suffices to settle for a good enough solution. One such solution—linesearch, or backtracking linesearch—iteratively tries increasing or decreasing values of ρ to determine a step size that gives sufficient improvements. In this setting, note that the gradient only needs to be computed once. Various conditions, like the Armijo-Wolfe conditions (Armijo, 1966; Wolfe, 1969, 1971), help formalize what "sufficient improvement" means.[8] Backtracking linesearch can be robust and finds use in more complex gradient-based algorithms, like L-BFGS (Byrd et al., 1995).

Looking specifically at stochastic optimization, Robbins and Monro (1951) provide

[8]These standard conditions (1) compare the current and proposed values of J to a modified first-order Taylor approximation of J; and (2) verify that the gradient has been sufficiently reduced.

two criteria that the sequence of step sizes should follow: (1) the sum of all step sizes should diverge ($\sum_t \rho_t = \infty$), but (2) the sum of all squared step sizes should converge ($\sum_t \rho_t^2 < \infty$). While a simple schedule like $\rho_t = \frac{\gamma}{t}$, for constant γ, satisfies the criteria, Hoffman et al. (2013) empirically demonstrate an effective alternative: given some *delay* $\tau \geq 0$ and *forgetting rate* $\frac{1}{2} < \kappa \leq 1$, set

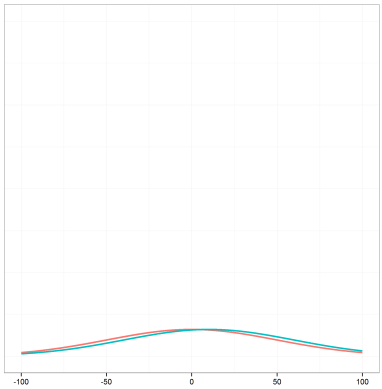
$$\rho_t = \frac{1}{(t + 1 + \tau)^\kappa}. \tag{2.26}$$

In this thesis I follow Hoffman et al. and use Eq. (2.26).

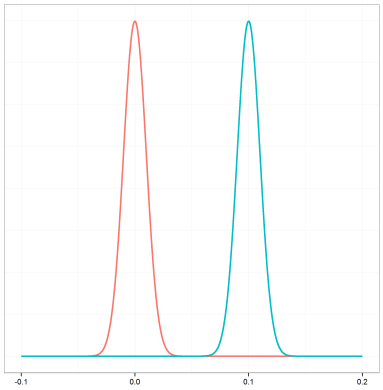
A third adaptive schedule that I also use in this thesis is AdaGrad (Duchi et al., 2011). In AdaGrad, the step size is actually a vector of step sizes, one for each component of the gradient. Each step size component $\rho_{t,i}$ takes into account the ith partial derivatives from all prior iterations. Letting $g_i^{(j)}$ be the partial derivative $\frac{\partial J}{\partial \Theta_i}$ at iteration j, the AdaGrad step size is

$$\rho_{t,i} = \frac{\delta}{\epsilon + \sqrt{\sum_{j \leq t} \left(g_i^{(j)}\right)^2}}. \tag{2.27}$$

The parameters δ and ϵ give additional user control, and in the latter case help stabilize the algorithm.



(a) Two distributions having a relatively high Euclidean distance (10), but low symmetrized KL divergence (1×10^{-6}). The distributions significantly overlap one another ($\mu_1 = 0, \sigma_1 = 10K$, $\mu_2 = 10, \sigma_2 = 10K$).



(b) Two distributions having a relatively low Euclidean distance (0.1), but high symmetrized KL divergence (100). The distributions do not significantly overlap one another ($\mu_1 = 0, \sigma_1 = 0.01$, $\mu_2 = 0.1, \sigma_2 = 0.01$).

Figure 2.3: Two sets of one-dimensional Gaussian distributions, under standard parametrizations (rather than with their natural parameters), show that Euclidean distance does not correlate with probability distribution similarity. These examples are due to Hoffman et al. (2013).

2.4.4 Optimizing Probability Spaces

Although we may specify probability distributions, in particular exponential families, according to some vector of parameters Θ, recall from §2.1.1 that these parameters often do not directly control the end distribution: rather, the natural parameters do. Though we can transform our specified parameters into natural ones via $\eta(\Theta)$, this function may not be linear. Therefore, changes in our Euclidean parameters Θ may not be proportionately reflected by changes in $\eta(\Theta)$ or the character of the distribution. These concerns will arise in chapter 8.

This lack of proportionate change can be seen in Figure 2.3, originally from Hoffman et al. (2013), which demonstrates an anticorrelation between standard parameters and the end distribution they parameterize. Specifically, Figure 2.3 considers four univariate Gaussian distributions: in Figure 2.3a, the distributions are parameterized by $\Theta_1 = (0, 10,000)$ and $\Theta_2(10, 10,000)$. The distributions display significant overlap, which we can measure through their symmetrized KL divergence, $\frac{1}{2}(\mathrm{D_{KL}}(\Theta_1 \| \Theta_2) + \mathrm{D_{KL}}(\Theta_2 \| \Theta_1))$. The Euclidean distance between Θ_1 and Θ_2 is relatively high (at 10), but their distribution distance is low (1×10^{-6}). Figure 2.3b demonstrates the opposite: the distributions are parameterized by $\Theta_1 = (0, 0.01)$ and $\Theta_2(0.1, 0.01)$. The distributions behave very differently, with a high symmetrized KL of 100, but a low Euclidean distance (0.01).

This suggests that the standard, Euclidean gradient may not best reflect how to better fit distributions. Recall that the update Eq. (2.25) included a Euclidean distance constraint. Amari (1982) and Amari (1998) propose that the update should reflect the underlying parametrization, or geometry, of the distribution and probability space. Amari proposed the *natural gradient* as a way to more accurately reflect the coupling between probability parameters and their distributions.

Specifically, Amari showed that the Euclidean constraints in the gradient ascent update Eq. (2.25) can be respecified in terms of a distance measure G that is specific to the target (probability) space. He then demonstrated that not only can the natural gradient be derived from the Euclidean gradient by premultiplying by a function of the

metric for the underlying space, but also that for exponential families, this measure is the Fisher information. Thus, he showed we can compute the natural gradient $\tilde{\nabla}_\Theta J(\Theta)$ from the Euclidean gradient via

$$\tilde{\nabla}_\Theta J(\Theta) = \mathcal{I}^{-1}(\Theta)\nabla_\Theta J(\Theta), \tag{2.28}$$

where $\mathcal{I}$ is the Fisher information defined by the model and parameters Θ.

Amari (1998); Sato (2001); Honkela et al. (2010); Hoffman et al. (2013, i.a.) have all explored using the natural gradient in variational inference for certain kinds of exponential family models; the natural gradient has shown to outperform Euclidean-based gradient optimization. I will use natural gradients in chapter 8, where I develop scalable, semi-supervised models of event and document representations.

Example: Topic Models

Recall from §2.3.2 that under mean-field variational inference, the gradient of the, e.g., topics has the form

$$\nabla_{\lambda_k} L(q) = \eta(\beta)^\intercal \nabla^2_{\eta(\lambda_k)} A(\eta(\lambda_k)) - \eta(\lambda_k)^\intercal \nabla^2_{\eta(\lambda_k)} A(\eta(\lambda_k)) + \sum_{d,i} \phi_{d,i,k} t(w_{d,i})^\intercal \nabla^2_{\eta(\lambda_k)} A(\eta(\lambda_k)).$$

We could analytically find the root of this equation, essentially by factoring out $\nabla^2_{\eta(\lambda_k)} A(\eta(\lambda_k))$, the Hessian of the topic (Dirichlet) log partition function. Recall from §2.1.1 that for exponential families, the Hessian of the log partition

is the Fisher information:

$$\nabla^2_{\eta(\lambda_k)} A(\eta(\lambda_k)) = \mathcal{I}(\eta(\lambda_k)).$$

Using the definition of the natural gradient Eq. (2.28) and the definition of the Dirichlet's natural parameters $\eta(\cdot)$, we compute the topic's natural gradient to be

$$\begin{aligned}
\tilde{\nabla}_{\lambda_k} L(q) = \; & \mathcal{I}^{-1}(\eta(\lambda_k)) \left[\eta(\beta)^\intercal \nabla^2_{\eta(\lambda_k)} A(\eta(\lambda_k)) - \eta(\lambda_k)^\intercal \nabla^2_{\eta(\lambda_k)} A(\eta(\lambda_k)) + \right. \\
& \left. \sum_{d,i} \phi_{d,i,k} t(w_{d,i})^\intercal \nabla^2_{\eta(\lambda_k)} A(\eta(\lambda_k)) \right] \\
= \; & \eta(\beta) - \eta(\lambda_k) + \sum_{d,i} \phi_{d,i,k} t(w_{d,i}) \\
& \beta - \lambda_k + \sum_{d,i} \phi_{d,i,k} t(w_{d,i}).
\end{aligned}$$

The natural gradient provided a principled method for achieving the same overall gradient update. Empirically, this gradient can be used much more easily within gradient ascent frameworks. See Hoffman et al. (2013) for additional details.

Chapter 3

Background: Structured Representations of Meaning

In this chapter I provide an overview on a number of ways to approach defining, learning and using event-based structured representations of meaning:

1. a symbolic and logic-based perspective, aimed toward *precisely* capturing event meaning from both theoretical and computable perspectives;

2. a resource-based perspective, aimed toward easily *annotating* event meaning; and

3. a classification-based perspective, aimed toward creating systems that demonstrate event meaning via *prediction*.

I have enumerated these as distinct items and will cover them in subsequent sec-

tions, but they are not mutually exclusive: one informs another. In particular, the resource- and classification-based perspectives act symbiotically, and annotation is often inspired by, if not grounded in, the theoretical or symbolic.[1]

3.1 Symbolic Representations: Precision and Computability

3.1.1 Event Logics

3.1.1.1 Davidsonian and neo-Davidsonian Events

Prior to the seminal work of Davidson (1967), the primary accepted logical form analysis of action sentences made impractical and unsatisfying assumptions about the lexicon. For instance, as (3.1) shows, there are many different ways that a core event can be modified (Kenny, 1963):

(3.1) $\overbrace{\{\text{John buttered the toast}\}}^{\text{core event}}$ $\underbrace{[\text{in the kitchen}]\ [\text{with a knife}]\ [\text{at midnight}]}_{\text{event modifiers}}$.

Analyzing the various scenarios in (3.1) requires one of the following compromises: (1) separate predicates of differing arity; (2) overly descriptive predicates; or (3) sen-

[1] Though this symbolic/numeric distinction aims to help guide the reader, there have been particular approaches relating to events and natural language that blur the line. Researchers have attempted to directly learn event knowledge, general knowledge or inference rules from noisy sources (Schubert, 2002). One such approach, inductive logic programming (Muggleton and de Raedt, 1994), has seen application in ontology induction (Kazakov, 1999), syntactic parsing with semantic (thematic role) constraints (Zelle and Mooney, 1994), and information extraction (Nijssen and Kok, 2003; Carlson et al., 2010).

Compromise	**pre-Davidsonian**
separate predicates of differing arity	buttered(x, y) buttered(x, y, z)
increasingly more descriptive predicates	buttered(x, y) buttered-in-kitchen(x, y)
default arguments to multi-arity predicates	buttered$(x, y, \texttt{in} = \text{KITCHEN}, \ldots)$ buttered$(x, y, \texttt{in} = z, \ldots)$

Table 3.1: A comparison of pre-Davidsonian approaches for handling modified base events, like "John buttered the toast in the kitchen."

sible, implicit default arguments to a predicate (with unspecified arity). See Table 3.1 for non-Davidsonian illustrations of how these compromises could be realized.

There are a number of issues with these compromises. They expand a verb's meaning (**denotation**) to be responsible for that verb's syntactic **valencies**, or the number, combinations, and types of arguments a verb may have. From the practical perspective, expanding the core denotations vastly expands the lexicon. Another issue is that, like entities, we can refer back to previously mentioned events. For instance, we can follow (3.1) with the elaboration "It was something he did when drunk," referring to the entire buttering episode with *it*. The compromises do not provide a method to easily capture this phenomenon.

Davidson (1967) argued that a proper analysis of action sentences should (a) separate event descriptors from the event predicate, and (b) then tie all descriptors back together with an event variable (individual). He thought a better way to represent the core event of 3.1 is by $\exists e.\ \textit{buttering}(\text{John}, \text{toast}, e)$. Further event descriptors are con-

joined, e.g., *With*(a knife, e). The variable e allows anaphora as well as nominalized (also called deverbal) events. A full Davidsonian account of (3.1) could be

(3.2) $\exists e.$ *buttering*(John, toast, e) $\wedge$ *With*(a knife, e) $\wedge$ *At-Time*(midnight, e) $\wedge$ *In-Location*(the kitchen, e).

While Davidson's approach fundamentally shifted how linguists thought of event meaning, it was not a panacea (Castañeda, 1967). Davidson's approach still requires some valence information to be part of the verb denotation. Why is the core predicate on (3.2) a 3-place predicate? And while the syntactic positions of "John" and "toast" typically correspond to an event's "core" arguments, what if the sentence instead read, "The toast was buttered in the kitchen with a knife at midnight?"

Many researchers proposed similar changes to Davidson's theory: extract the core arguments from the verbal predicate and represent their relation to the event with separate role predicates (Castañeda, 1967; Carlson, 1984; Parsons, 1990, i.a.). While the exact labels for these new roles, or any roles in general, was up for debate. A **neo-Davidsonian** representation of (3.1) could be

(3.3) $\exists e.$ *buttering*(e) $\wedge$ *Agent*(John, e) $\wedge$ *Patient*(toast, e) $\wedge$ *With*(a knife, e) $\wedge$ *At-Time*(midnight, e) $\wedge$ *In-Location*(the kitchen, e).

As we will see in §3.1.2, this draws on a notion similar to, and in some cases derived from, Charles Fillmore's case grammar (Fillmore, 1967).

3.1.1.2 Logics with Doubt

> The aim is to have a semantic account that does not go through any sort of first-order 'logical form', but operates off of the syntactic rules of English.
> — Barwise (1981)

Event meanings were often studied with verbs that

1. yield a 'crisp,' completed action or outcome, excluding verbs such as stative verbs and verbs of communication and reporting; and

2. do not lend themselves to doubt, such as counterfactuals and verbs of belief and attempt.

Counterfactual statements, such as those presupposing existential instantiation ("John saw *a ghost*"), and counterfactual predications ("John wanted to butter the toast (but didn't)") pose issues for Davidsonian or neo-Davidsonian approaches. The *wanting* may be real, the *buttering* may not be. In this section, I provide an overview of two extensions or alternatives: one due to Hobbs (1985), and one to Schubert (2000). Both of these can be considered modified versions of Barwise and Perry's situation theory (Barwise and Perry, 1981). All three theories actively try to keep the logical form representation as close as possible to a natural language representation (leveraging light syntactic representations as needed).

The core idea in situation theory is that we communicate by describing *situations*—a catch-all for events, states, actions, eventualities, and beliefs. Situations support notions of completeness and minimality, and issues that arise (implicitly)

in real human language understanding. Example issues include handling ambiguous quantifier scoping, interpreting statements of belief or reporting (where one or more participants have incomplete knowledge about the embedded clause), and resolving logically-difficult implicit domain restrictions and entity/anaphora reference.

Let's compare (3.1), "represented" by a situation variable s_1, and (3.4), "represented" by a situation variable s_2,

(3.4) John made breakfast.

To delve more into what it means for a variable to "represent" a described situation, Barwise and Perry use a notion of *minimality*. A sentence or proposition is minimally supported by a situation if that situation meets, but does not exceed (describe more than) the proposition. For example, if s_2 minimally supports ("represents") (3.4), then it may also support (3.1). In contrast, s_1 minimally supports ("represents") (3.1). This notion of minimality is lacking in general (neo-)Davidsonian accounts, though there have been approaches to bridge that gap (Kratzer, 2016).

Hobbsian Logical Forms

Hobbs (1985) extends the Davidsonian position to all types of predications p, including stative, propositional, and counter-factual predications. The core of the proposal centers around schematic axioms that rewrite logical forms in order to explicitly represent the (lack of) some event or event prerequisite (not) being met.

Like Davidson's eventive predicates, Hobbs represents verbal denotations as pred-

icates that take an additional argument. This argument represents a characterization of the predicate. However, Hobbs makes a distinction between the theoretical denotation and the actualized denotation, e.g., a failed buttering has different implications than a completed buttering. Hobbs represents theoretical denotations with *primed predicates*, like *buttered'*, and actualized denotations with *unprimed predicates*, like *buttered*. The axiom he proposes relating primed p' and unprimed p predicates utilizes a unary EXIST predicate in a notational rewrite:

$$p\left(\{x_i\}_{i=1}^{n}\right) \equiv \exists e.\text{EXIST}(e) \wedge p'\left(e, \{x_i\}_{i=1}^{n}\right). \tag{3.5}$$

This EXIST predicate returns true if and only if its argument refers to some actual "thing" in the "real" universe. Hobbs's theory applies EXIST to *any* object, be it a "real" entity, actualized events, or non-real entities and events.

The primed/unprimed axiom does not solve all problems though, especially regarding "identity verification:" when do two named objects refer to the same underlying object (i.e., entity coreference resolution). Hobbs does not consider identity verification to be a major issue for most real world systems (18-20); this is unfortunate, as there remain significant challenges in coreference resolution (see, e.g., Lu and Ng (2017), and chapter 7 of this thesis). Hobbs's alternative suggestion to rely on metonymic interpretations runs into limitations on current state-of-the-art metonymy interpretation (Ferraro, 2011). See §3.4.1 for an extended discussion.

Hobbs extends this theory to provide as "simple" an approach to causality as possible (Hobbs, 2005). Important to the framework are *causal complexes*—the minimum set of eventualities that must hold for some effect to hold—and modals, such as *would*. Roughly, Hobbs is saying that if an eventuality c causing y can be captured by x, then x also captures the modulating effect of c on y. Though not a major component of the theory, Hobbs very briefly talks about causation and probabilities; he defines the probability of a causal complex causing an eventuality, given that a superset of the causal complex can cause e, is the joint probability of all superset variables not in the causal complex being true. However, this all is an afterthought in Hobbs's system; he does not propose how to obtain any of these probabilities.

Episodic Logic

Hwang and Schubert (1993), and subsequently Schubert (2000), argue that different types of eventualities behave differently, and any system that talks about eventualities must reflect those differences. In particular, he notes that propositions, but not events, can be stated or proven, whereas events, but not propositions, can have participants, or be commenced. He argues that Hobbs's approach is deficient in handling these nuances.

Episodic logic links *episodic* variables with Davidson-inspired formulas Φ via two operators, $\star$ and $\star\star$ (Schubert, 2000; Schubert and Hwang, 2000). Briefly, we say ϵ *fully* characterizes Φ when $[\Phi \star \star\epsilon]$, and *partially* describes Φ when $[\Phi \star \epsilon]$. The

operators allows complex sentences, with associated modality, causality, etc. to be associated with a single episodic reference. In contrast, Hobbs's flat notation can only associate eventuality references with eventuality modifiers in a chained fashion (Hobbs, 1985, pg. 8, ex. 4).

As Schubert notes, we can try to equate $\star\star$ with Hobbs's prime notation, but there exists a subtle, yet important, difference: the prime notation is simply defined as a notation on *existing predicates*, while $\star\star$ is a well-defined, systematic operator on *formulas*. Of course, $\star$ is also an operator in the same sense that $\star\star$ is; Hobbs's logic does not provide an analogous candidate.

Full episodic logic requires a lot. It is beyond first order logic, so binary, let alone any weighted, inference is intractable. Both $\star\star$ and $\star$ require quantifier scoped logical forms, which can be exponential in the number of quantifiers. In contrast, Hobbs opts for a flat structure that is as close as possible to the sentential form; this could make it easier to (create a system to) produce logical forms. While direct manipulation of logical forms may mean easier, but less sound (and complete) inference. We have seen this simplicity in both Hobbs (1985) and Hobbs (2005). Regardless, both Hobbs's and Schubert's systems and theories require external knowledge—as logical axioms, meaning represenations, or other forms, such as frames—in order for any inference to actually take place. See §3.4.2 for an extended discussion on episodic logic's expressiveness.

Case	Meaning
Agentive	The instigator of an action
Instrumental	An inanimate object (or force) involved in an action
Dative	The object being affected by an action
Factive	The result of an action
Locative	Where an action takes place
Objective	Nouns who participate in an action, as specified by verb denotation

Table 3.2: The six deep cases from Fillmore (1967), with summary descriptions.

3.1.2 The Case for Fillmore

Prior to Fillmore (1967)'s seminal "The Case for Case" ("C4C"), many human constructed grammars emphasized morpho-semantic accounts—the morphology of a language explaining meaning—using lightweight, "rule of thumb" syntactic rules as a bridge, of sorts. In C4C, Fillmore advocates for a deep connection between the syntax and the semantics of a language, arguing that, at a minimum, six, deep nominal "cases" explain both observed syntactic valencies and corresponding semantics. These cases, drawing inspiration from Latin cases, are shown in Table 3.2. Fillmore analyzed sentences as a verb and one or more noun phrases, where each noun phrase has one of these cases. These cases encode semantic and pragmatic properties of the arguments; together, the cases for a verb effectively represent its selectional preferences. These notions of deep case inspired and refined the development of theta (thematic) roles, and the neo-Davidsonian representation of (3.3).

3.1.2.1 Frame Semantics

Frame semantics (Fillmore, 1976, 1982) explain how we use language together with idealized "cognitive frames"—or those "structures" we use to encode everyday experiences—in order to understand language and our world (Minsky, 1974; Fillmore and Baker, 2009). Frames are data structures that are **triggered** by sense-disambiguated words, called **lexical units**. Extending his notion of deep case, they identify and categorize those words and concepts that particpate in actions of the lexical units; they also refer to one another in order to build up meaning. The interconnectedness gets at a core idea of frame semantics: we can only understand the meanings of words, concepts, and actions by understanding the meanings of their associated words, concepts, participants, and actions. For example, to understand the atypicality of (3.1), we must not only understand what the objects (toast, kitchen and knife) are, nor solely understand what is involved in buttering; rather, we must know *when* butterings and (presumed) eating of toast are likely to occur. That is, we must have background, social knowledge that allows us to bridge gaps within the observed language.

Frames are perhaps best known through the machine readable resource of FrameNet, which I will cover in §3.2.1. Additional theoretic and modeling discussions of frames will be covered in more depth in chapter 7.

3.1.2.2 Construction Grammar

Construction grammar is a syntax-based approach for combining lexical, syntactic, and semantic rules and expectations (Fillmore et al., 1988). Working with elements called "constructions," which are often just pairs of syntactic and semantic patterns, construction grammar theory posits that certain phrasal meanings depend on the syntactic configurations of individual words. Syntactically, construction grammar accounts for idiomatic uses of phrases and commonly occurring elements, such as function words (often taken to be non-content bearing and discarded in NLP). Whereas case grammar focused on assigning semantic labels to words and spans, construction grammar combines semantic expectations into the syntactic structure.

3.1.3 Discourse Representation Theory

Discourse representation theory (Kamp, 1981; Heim, 1982, DRT) is an incremental approach for semantic processing. Its primary aim is to describe a formal approach for representing event meaning and anaphora (entity coreference) accurately. It is defined recursively in terms of *discourse representation structures* (DRSs), where each DRS has access to a set of "discourse referents," i.e., the entities that appear, and the various facts and knowledge, typically represented as predicate relations defined on the discourse referents, that have been introduced into the discourse. Because pronoun resolution can be ambiguous, new DRSs may have discourse facts that are

partially or fully unbound. As new information is introduced, DRT defines procedures by which new DRSs are merged into the existing structure and discourse referents are merged (anaphoras are resolved).[2]

Lascarides and Asher (1993) present a method for performing this merge in a way that respects discourse temporal interpretation.[3] They argue that linguistic knowledge alone cannot solve the inference problem: rather, non-monotonic world knowledge reasoning must occur. Lascarides and Asher devise a methodology for employing discourse relations and presupposed, empirical background knowledge as constraints that allow chaining DRSs.[4]

Lascarides and Asher present a set of five necessary, but not necessarily sufficient, discourse relations needed to perform coherent reasoning in DRT. While an analysis of temporal ordering must consider causality, not every discourse relation must: a nuanced relation like NARRATION can be used to describe how one situation may be a consequence of another, even though there may not be a direct causal, or stated temporal, link between them.

While Lascarides and Asher do not talk about probabilities, per se, they do consider tendencies in rules. That is, if you know that some meanings or interpretations

[2]DRT and the iterative merging processes are often presented pictorially with boxes; thus, DRT is often thought of as the "box theory."

[3]Although I do not consider aspects of temporal extraction in this thesis, the discourse relations and ideas Lascarides and Asher consider *are* relevant, particularly to chapters 7 and 8.

[4]They visualize discourse representation pairs (DRPs) as a graph, where DRS nodes can be either open or closed: intuitively, open nodes draw upon Gricean maxims, signifying that something 'relevant' to discourse understanding has not yet been said. An open DRS must either have been just added or needing some further explanation. Though from a computational standpoint they build the graphs left-to-right and depth-first, they claim there are not always unambiguous ways to resolve openness, particularly in larger graphs.

occur more frequently than others, you can obtain bounded estimates of the probabilities of those meanings. They argue that knowing bounds on the probabilities (even if the actual values are unknown) can help resolve discourse ambiguities.

3.2 Annotating Event Knowledge

In this section, I consider three different ways that researchers have annotated events: as structured, predicate argument representations (§3.2.1); as semi-structured spans linking multiple sentences (§3.2.2); and as featurized representations (§3.2.3).

Events are typically thought of as being evoked by verbs; as a result, many event ontologies's annotations are defined on verbs. However, there are also **deverbal** events, i.e., those that are evoked or represented by non-verbs. Most often, the event is represented through a nominalization; for instance, rather than evoking a CONFESSION event with a verbal predicate "confessed,"

(3.6) The man **confessed**. He was sent to jail.

we can instead evoke the same event with the verb's nominalization,

(3.7) The man's **confession** sent him to jail.

While some of the resources covered below, notably FrameNet, annotate deverbal events, there are annotation efforts, specifically NOMLEX (Macleod et al., 1998) and NomBank (Meyers et al., 2004), that focus on these types of events. WordNet (Fellbaum, 1998), through its hierarchical ontology, also encodes deverbal event

information. Due to issues of scope, I will not cover explicit deverbal annotations below.

Moreover, there are a number of annotation efforts that, due to scope, I cannot cover. These include EventCorefBank (Bejan and Harabagiu, 2010) and its extension, ECB+ (Cybulska and Vossen, 2014), which does for events what entity coreference does for mentions: it groups together descriptions of the same event; plethora annotation efforts for targeted information extraction (Over and Yen, 2004; Walker et al., 2006; Giannakopoulos et al., 2017; Strassel et al., 2017, i.a.); and multidisciplinary efforts (Heise, 1989; Griffin, 1993; Kim, 2010, i.a.).

3.2.1 Predicate Argument Annotation

Generally, predicate argument annotations are understood through graphs and trees, such as syntactic parsing: words or spans are connected to one another in a directed fashion, and the connections may or may not be labeled with names like "subject" and "direct object." Here though I generalize the meaning of predicate argument annotation to that of representing meaning through generalized slot filling: a particular item, generally a word, evokes a certain number of slots that must, can, or cannot be filled by other items or concepts. In the parsing example, the slots are the various grammatical relations, with the entire structure defining the ordering (or positional) information of the relations, i.e., which slots (grammatical relations) appear to the left of the trigger, and which appear to the right. This view allows us to

COMMUNICATION: A COMMUNICATOR conveys a MESSAGE to an ADDRESSEE the TOPIC and MEDIUM of the communication also may be expressed....

Role Filler	Role	Role Meaning and Gloss
The paper	COMMUNICATOR	The sentient entity that uses language in the written or spoken modality to convey a MESSAGE to the ADDRESSEE.
—	MEDIUM	The physical or abstract setting in which the MESSAGE is conveyed.
the truth	MESSAGE	MESSAGE is a proposition or set of propositions that the COMMUNICATOR wants the ADDRESSEE to believe or take for granted.
—	TOPIC	The TOPIC is the subject matter to which the MESSAGE pertains. It is normally expressed as a PP Complement headed by "about", but in some cases it can appear as a direct object.

(a) An excerpt of the FrameNet frame COMMUNICATION.

Role Filler	Role	Role Meaning and Gloss
—	$\texttt{Arg}_0$	*none specified*
the truth	$\texttt{Arg}_1$	The object casting a reflection.
The paper	$\texttt{Arg}_2$	The surface casting the reflection; the image being reflected.

(b) PropBank frame `reflect-v-1`.

Figure 3.1: FrameNet and PropBank frames for the verb "reflect." The FrameNet labeling is based on the expanded lexicon of (Pavlick et al., 2015).

consider FrameNet, PropBank, VerbNet and other verb valency databases together conceptually.

FrameNet

The Berkeley FrameNet Project (Baker et al., 1998; Ruppenhofer et al., 2006) is an on-going endeavor to put Fillmore's frame semantic theory into practice by performing an exhaustive exemplar annotation effort. A FrameNet frame consists of

a set of lexical units (generally part of speech tagged words, but sometimes multiword expressions and idioms) that *trigger* said frame. FrameNet implicitly assumes distinct word senses: if the same lexical unit appears as a trigger for different frames, then those frames are assumed to represent different senses of the lexical unit.

Each frame has multiple collections of roles (termed *frame elements*) to fill: some are "core," and represent a notion that is critical to fully understanding the frame; others are not, e.g., "peripheral" or "extra-thematic," that supplement the meaning. Note that a core role does not need to be explicitly represented in text. For instance, in Figure 3.1a, all four listed roles (frame elements) for the `Communication` frame are core; the frame also has a number of peripheral (unlisted) roles like DURATION and PLACE that provide auxiliary information about the event.

Frames in FrameNet are arranged in an ontology, with asymmetric relations defined between two frames. Among these relationships, frames can inherit (be inherited by) one another; indicate that "use" of one by another; represent linguistic alternations through inchoative and causative relationships; and composition of frames to form larger events. For instance, `Communication` can be refined into a `Gesture` frame, uses an `Information` frame, and can be used by a `Candidness` frame. These inter-frame relations tend to be underused, though they have helped some unsupervised induction tasks (Bejan, 2008).

FrameNet is intended to be a high precision resource: according to Ruppenhofer et al. (2006), its *structural* annotations are exhaustive *within* a frame. However, its

recall can be poor: not every triggering lexical unit is listed as a valid trigger. This is an issue that multiple efforts have attempted to address (Rastogi and Van Durme, 2014; Pavlick et al., 2015).

Despite these limitations, FrameNet has been shown to be both influential and useful. In addition to spurring the development of frame semantic parsers (Baker et al., 2007; Bejan, 2009; Das et al., 2010), it has also helped to form the NLP task of semantic role labeling (Gildea and Jurafsky, 2002; Litkowski, 2004). Many efforts have shown it to be useful in downstream tasks (Chen et al., 2014; Agarwal et al., 2014; Narayanan, 2014; Rastogi et al., 2015; Peng and Roth, 2016; Ferraro and Van Durme, 2016, i.a.); see Petruck and de Melo (2014) for additional instances.

PropBank

PropBank (Palmer et al., 2005) annotates semantic roles atop the Penn TreeBank, a collection of 60K manually-created (constituent) parse trees (Marcus et al., 1993). Palmer et al. (2005) aimed to annotate each verb within the Penn TreeBank with a frame-like structure. They called they structures *framesets*. Like FrameNet frames, PropBank framesets list ways in which they can be invoked, and specify a set of frameset-specific roles. In total, they annotated roughly 3,300 verb types with 4,500 framesets.

Unlike FrameNet's descriptive role labels, PropBank's are coarse $\texttt{Arg}_i$ labels: instead of a MESSAGE (Figure 3.1a), PropBank uses $\texttt{Arg}_1$ (Figure 3.1b). Though the

same label forms are used across framesets, they are distinct: the $\texttt{Arg}_0$ for one frameset (technically) places no requirements or restrictions on the $\texttt{Arg}_0$ of another. However, there is a tendency for different $\texttt{Arg}_i$s to represent the same neo-Davidsonian semantic role. PropBank has six primary $\texttt{Arg}_i$ roles, though they can have suffix modifiers, indicating manner, cause, discourse and the like. Because PropBank sits atop constituency trees, Palmer et al. annotated entire linguistic phrases as role fillers; however, they cannot cross sentences.

Like FrameNet, PropBank has been very influential. It has inspired a number of different shared tasks (Carreras and Màrquez, 2004; Carreras and Màrquez, 2005; Surdeanu et al., 2008) and subsequent parsers. It has also been incorporated into composite resources—those that either layer annotations atop one another or provide a translation from one annotation schema to another (Loper et al., 2007; Weischedel et al., 2013).

Verb Valencies and Selectional Restrictions

While we can use "reflect" in a simple transitive construction ("the paper reflected the truth"), we can also use it with sentential constructions ("the paper reflected how the operation happened") and intransitively ("the truth reflected the truth to its readers"). Other verbs, e.g., "acknowledge," and "show," have the same syntactic allowances. Beth Levin provided a very comprehensive enumeration and clustering of different syntactic alternations (Levin, 1993).

VerbNet (Schuler, 2005) implements Levin's classes in a machine readable form. Beyond enumerating exemplars, the clusters of verbs and their syntactic alternations, VerbNet provides its own semantic frame analyses with neo-Davidsonian semantic roles for more than five thousand verbs in each of its synactic frames. For example, the subject and object of "reflect"'s transitive syntactic frame are labeled, respectively, the (semantic) agent and a non-sentential topic. These are then used in the neo-Davidsonian form, with a prepended "?" indicating an implicit role filler:

$$\exists e : \text{transfer\_info}(\text{during}(E), \text{Agent}, ?\text{Recipient}, \text{Topic}) \ \wedge \ \text{cause}(\text{Agent}, E).$$

Note in transitive constructions, the recipient is implicit.

Because the semantics are defined for each syntactic frame, the two are, by construction, linked together: the semantic declarations, like the syntactic alternations, apply to each verb within the syntactic frame of that cluster. VerbNet allows roles to place certain selectional restrictions on what can fill them: for instance, AGENTs and RECIPIENTs of "reflect" should both either be animate or represent organizations.

There have been a number of attempts to validate, augment or supplement, in particular, these semantics. VerbCorner (Hartstone et al., 2013; Hartshorne et al., 2014) validates the semantic annotations in VerbNet, ensuring that the listed properties (selectional restrictions of arguments) are those that are logically entailed of the argument, e.g., if the "animacy" properties of reflect's AGENTs and RECIPIENTs

must be true. Bonial et al. (2011) outlined a hierarchy over VerbNet roles based on the selectional restrictions and properties of those roles. Meanwhile, the on-going Pattern Dictionary of English Verbs (Hanks, 2013) lists both the valency of verbs as well as semantic role restrictions according to a backend ontology (Pustejovsky et al., 2004). These are in contrast to Reisinger et al. (2015), who, as part of a larger agenda, identify and validate VerbNet properties that are *likely* to be true.

Composite Resources

There are composite annotations that join multiple resources together, or use and extend existing resources to form new ones, also exist. Some operate at the type level, describing generalities in language and semantics. For example, the SemLink project (Loper et al., 2007) performs type-level mappings among VerbNet, FrameNet, PropBank, and WordNet. It is an on-going effort, with internal and external contributions (Reisinger et al., 2015).

Other resources operate at the instance level, demonstrating how the different resources can actually be applied to different and new types of text. For instance, Abstract Meaning Representation (Banarescu et al., 2013) use (and extend) PropBank frames and roles to represent edge labels in a semantic graph representation. Unfortunately, this section cannot be exhaustive; see Abend and Rappoport (2017) for a survey of recent semantic annotation efforts.

3.2.2 Discourse over Multiple Sentences

Prasad et al. (2008) provide an empirical account of discourse and causality. Their effort focuses on annotating both explicit (3.8) and implicit (3.9) discourse connections among the one million words from the WSJ portion of the Penn Treebank:

(3.8) {U.S. Trust ... has faced intensifying competition}$_{\text{Arg}_1}$ .... **As a result**, {U.S. Trust's earnings have been hurt}$_{\text{Arg}_2}$.

(3.9) ... {Some have raised their cash positions to record levels}$_{\text{Arg}_1}$. [**Implicit = because**] {They help buffer a fund}$_{\text{Arg}_2}$...

For each example relation (bold), there are two relational arguments, labeled Arg_1 and Arg_2. These annotations sit atop the original syntactic parse trees; with some additional effort, they could also be aligned with shallow semantic annotations, such as those given by PropBank. Although both of these examples are cross-sentential, intrasentential discourse relations were also annotated. To maintain interannotator agreement, they provided lexical realizations of implicit relations (c.f., 3.9). They call their annotations the Penn Discourse Treebank (PDTB).

The annotations are intended to be theory neutral, and indeed, there are no DRS annotations, or ontologically ambiguous eventuality variables floating around (but see §3.4.4 for more on what this means). Whereas Lascarides and Asher (1993) and Hobbs (2005) provide formal mechanisms to talk about, i.a, causality, the PDTB effort is more about explaining via observable surface forms. Prasad et al. (2008) selected

	The	**paper**	**reflected**	**the**	**truth**	**.**
		Proto-Agent			Proto-Patient	
AWARENESS		very likely			very unlikely (NA)	
CHANGE_OF_LOCATION		very unlikely (NA)			very unlikely (NA)	
CHANGE_OF_STATE		unsure			very unlikely	
CHANGES_POSSESSION		very unlikely			very unlikely (NA)	
CREATED		very unlikely			very unlikely	
DESTROYED		very unlikely			very unlikely	
EXISTED_AFTER		very likely			very likely	
EXISTED_BEFORE		very likely			very likely	
EXISTED_DURING		very likely			very likely	
EXISTS_AS_PHYSICAL		very unlikely (NA)			very unlikely (NA)	
INSTIGATION		very likely			very unlikely (NA)	
LOCATION_OF_EVENT		very unlikely (NA)			very unlikely (NA)	
MAKES_PHYSICAL_CONTACT		very unlikely (NA)			very unlikely (NA)	
MANIPULATED_BY_ANOTHER		very unlikely (NA)			very likely	
PREDICATE_CHANGED_ARGUMENT		unlikely			very unlikely	
SENTIENT		very unlikely (NA)			very unlikely (NA)	
STATIONARY		very unlikely (NA)			very unlikely (NA)	
VOLITION		very likely			very unlikely (NA)	

Figure 3.2: A full semantic proto-roles (SPR) annotation, as provided by White et al. (2016).

the explicit relations according to grammatical categories such as subordinating and coordinating conjunctions, and discourse adverbials. Implicit relations were lexically encoded. Many of the examples Lascarides and Asher (1993) and Hobbs (2005) consider fall into the "implicit" category.

3.2.3 Featurized Representation and Expectations: Semantic Proto Roles

One criticism of frame semantics is that the frames and concepts are both defined as discrete items: a particular frame has certain roles, but that role's *label* ends up

carrying the bulk of the meaning. Dowty (1991) argued that we label an entity an AGENT, for example, by comparing that entity, and how it participates in the action, to our notion of how a prototypical "agent", or a PROTO-AGENT, would act. For instance, an AGENT typically will have awareness, act volitionally, and may enact change, whereas a PATIENT will generally be affected by a change. Dowty's thematic proto-role theory proposed to replace discrete semantic roles with these proto-roles; they can actually be represented as collections (clusterings) of properties that are true of that entity's participation in the action.

Semantic proto-role (SPR) theory (Reisinger et al., 2015; White et al., 2016) was motivated by Dowty (1991)'s thematic proto-role theory. Whereas Dowty proposed replacing roles with judgments about properties and characteristics that *are* true, SPR proposes replacing roles with judgments as to properties and characteristics that are *likely* to be true. In Figure 3.2 I show a full SPR analysis of the two arguments of "reflect," where the property likelihood judgments are human annotated judgments from White et al. (2016). While SPR will be discussed more in chapter 5, I would like to draw attention to a couple of key elements and distinctions of SPR.

Notice that the "paper" is very likely to be "aware" during the reflection situation. On the other hand, notice that while the "paper" is very unlikely to be sentient, it is also judged that it does not make sense to ask if it even is sentient (the "NA", standing for **not applicable**). Together, these two property likelihood judgments heavily suggest that the interpretation of the "paper" is metonymic, with the term "paper"

standing in for the editors and journalists. This is important because it suggests some annotators may be performing a type of semantic promotion or elaboration.

The issue of the applicability of a property is a difficult one (Reisinger et al., 2015). Namely, if we ask about a certain property with respect to a concept, what presuppositions are we making? Consider the sentence

(3.10) Chris ate a pastry.

While it would be reasonable to say the pastry was very likely to have existed before "participating" in the eating and it was very *unlikely* to have existed after, what can we say about the pastry's volition? Arguably the pastry did not consent to being eaten, which might suggest a "very unlikely" rating. However, does it even make sense to talk about a pastry being volitional?

3.3 Event Meanings Through Tasks

It is generally a rule that where there is a resource, there is a task about predicting items in it. This is the case for the resources described in §3.2. For instance, FrameNet and PropBank have each allowed for the creation of a number of different frame semantic parsers (Baker et al., 2007; Bejan, 2009; Das et al., 2010; Titov and Khoddam, 2015; FitzGerald et al., 2015; Wolfe et al., 2016). Though a new task, researchers have proposed methods for performing semantic proto-role labeling (Teichert et al., 2017).

While these are very useful, especially as they allow complex analyses on novel and varied corpora (c.f., chapter 4), the tasks that I will focus on summarizing here revolve around analyzing language use in a more holistic manner. Specifically, I consider semantic language modeling and information extraction tasks. I will return to the former for a more in-depth exploration in chapter 7.

3.3.1 Semantic Language Modeling

Chambers and Jurafsky (2008), subsequently extended by Chambers and Jurafsky (2009), helped renew interest in narrative scripts (event chains). To evaluate his PMI-based event scripts, Chambers and Jurafsky (2008) proposed the task of narrative cloze—given a collection (ordered or not) of verbs, hold out one of the verbs and predict it, given the remaining verbs. Though narrative cloze was presented as a prediction task, I helped show that narrative cloze can be productively thought of as language modeling (Rudinger et al., 2015). Interpreting narrative cloze as language modeling, the subsequent efforts that narrative cloze inspired can be thought of as helping to advance semantic language modeling in a variety of ways: the benefit of syntactic information (Chambers and Jurafsky, 2009), the incorporation of additional context and more robust parameter estimation (Jans et al., 2012; Pichotta and Mooney, 2014), neural methods (Granroth-Wilding and Clark, 2016; Modi, 2016), and the modeling of longer narratives (Mostafazadeh et al., 2016).

A number of approaches based on clustering and topic modeling have been pro-

posed to better model diverse descriptions of events. While some have used basic topic models but with sophisticated semantic observations, such as based on FrameNet or other predicate-argument formulations (Bejan, 2008; Van Durme and Gildea, 2009; Kasch, 2012), others have proposed derivative models Materna (2012); Gottipati et al. (2013); Bamman et al. (2014); Frermann et al. (2014); Ferraro and Van Durme (2016). Others still have presented neural methods (Peng and Roth, 2016; Pichotta and Mooney, 2016; Granroth-Wilding and Clark, 2016; Iyyer et al., 2016).

Some have approached semantic language modeling through computational plot analysis (Lehnert, 1981). For example, Goyal et al. (2010) and Goyal et al. (2013) present AESOP, which analyzes a class of narratives' plot, including identifying each character and inferring connections between the plot and a character's "state of mind." They provide resource-rich methods, utilizing ontologies like FrameNet, to identify certain kinds of verbs from 34 manually annotated tales. Meanwhile Chaturvedi (2016) models relationships within narratives with a variety of (neural) sequence models.

3.3.2 Information Extraction

The event identification and extraction tasks from the Message Understanding Conferences are the canonical complex event extraction task within NLP. The most popular of these is the MUC-4 template induction task (Sundheim, 1992), which is based on (1) the notion of a template, such as BOMBING, and (2) selecting some

{*Three people*} have been killed ... as a result of a {*Shining Path*} {*attack*} today against a community in Junin...

(a) A sample document from MUC-4.

slot	**text filler**
TYPE	*attack*
PERP	*Shining Path*
# KILLED	*Three people*

(b) A MUC-4 ATTACK template filling the slots directly from the text.

slot	**reified filler**	**text filler**
TYPE	`armed action-12`	*attack*
PERP	`attacker-1`	*Shining Path*
# KILLED	`attackee-43`	*Three people*

(c) A MUC-6 ATTACK template filling its slots with reified filler objects, whose provenance is bits of textual evidence.

Figure 3.3: Contrasting MUC-4 vs. MUC-6 on a sample MUC document.

number of slots or roles to fill, such as PERPETRATOR. The 1700 MUC-4 documents, on which most recent prior work has focused (see below), are concise newswire-style reports labeled primarily with arson, bombing, kidnapping, and attack (e.g, murder) templates.

While allowing a basic description of complex events, the size, domain and original intended use limit its effectiveness from both practical and theoretical standpoints.The lesser-used MUC-6 data suffer from similar problems, though as discussed above the MUC-6 formulation is slightly richer.

While still considering a template to be a collection of roles to fill, MUC-6 (Sundheim, 1996) presented a richer event description than did MUC-4. Crucially, the

MUC-6 representation reified (some of) the slot fillers, thereby requiring a deeper, hierarchical interpretation of templates. For instance, a MUC-4 slot pointed directly into the text (e.g., Figure 3.3b), while a MUC-6 slot pointed to an entity object (Figure 3.3c). This entity object was generated from textual clues.

A number of generative, Bayesian models have been proposed for the MUC task, or derivative tasks. Bamman et al. (2013) and Chambers (2013) both adopt a topic modeling approach, viewing documents as bags of entities and their mentions. Documents in these models are admixtures over templates and slots, assigned at the entity level. Cheung et al. (2013) meanwhile view a document as a Bayesian Markov model. In all of these cases, syntactic dependency information drives the modeling effort. These models will be considered in greater depth in chapter 7.

In contrast to the above works, and to this thesis, Nguyen et al. (2015) present a purely entity-driven generative model for event induction *for MUC*. While this thesis, where applicable, and the above models consider entities and their event assignments specific to a particular document, Nguyen et al. simply model the entities, and do not model inter-document differences. That is, rather than model their corpus as a set of documents, which are admixtures of template- and slot-assignments, Nguyen et al. model all entities, across (and without regard to) document boundaries, as an admixture of slot-assignments. They define an event template implicitly, through post-hoc slot assignments that optimize the downstream MUC F-score.

Other efforts have focused on both generative and discriminative models for less-

than-supervised template induction. Minkov and Zettlemoyer (2012) presented a joint model for unsupervised learning and extraction of relational schemas, while Haghighi and Klein (2010) presented a semi-supervised entity-centric model.

Sha et al. (2016) frames unsupervised template-based information extraction as an integer, non-linear program. They apply normalized cut, a method developed for image segmentation, in order to solve this constrained optimization problem; this method finds clusters that tend to be internally homogenous and externally distinct. In a very similar vein as Chambers and Jurafsky (2011) and Chambers (2013), they employ sentence constraints, to encourage consistency across assignments within the same sentence.

For supervised template induction, multistep approaches are quite standard, as are methods incorporating document level information. Unlike unlabeled probabilistic approaches, much of the previous effort has gone into identifying *trigger* words for relevant discourse phrases and relations. Specifically, Maslennikov and Chua (2007) incorporated rhetorical structure theory into pipelined classification, while separating a sentence into primary and secondary vital information spans and identifying anchor words or phrases that trigger discourse relation among these spans. Patwardhan and Riloff (2009)'s conditional model jointly identified event-carrying sentences and role fillers from those sentences using semantic class information and keying off of syntactic patterns. Chen et al. (2011a)'s feature driven generative model declaratively specifies broad discourse constraints, as well as identifying trigger words.

Bootstrapping approaches to template extraction and learning have met with success. Huang and Riloff (2013) found it useful to incrementally identify and build up agent and purpose and an agent and purpose extraction phase. Both phases pattern-match dependency parses of probable event sentences using information gleaned in the previous iteration. A parametric generative model captures the notion that even though a news story may be about one main event, multiple sub-events may also be described; they therefore allow every "important" word in a document to be generated from either one of many global or local unigram language models. All "non-important" words are generated from a single background model. Finally, Liao and Grishman (2010) and Reichart and Barzilay (2012) both use supervised graphical model to extract multiple templates from a single document according to global (document-level) and local (sentence-level) constraints. Note that the discourse portions of these efforts can be viewed (loosely) as a coarse approximation to the elaboration, narration, etc. framework developed by Lascarides and Asher (1993), as discussed in §§ 3.1.1 and 3.4.

Other applications exist for structured event semantics and meaning representations beyond straight slot-fill information extraction. Irwin et al. (2011) attempt to incorporate narrative schema information into a cluster-ranked coref system, based on Rahman and Ng (2009), that classifies each potential mention iteratively. They attempt to go beyond a subset of standard shallow semantic features, such as binary demonstrative indicators and named entity class, via clustering of Chambers and Ju-

rafsky (2009)'s size-12 narrative schemas. Specifically, when considering whether to add a mention m to an existing entity e, they first group all schemas that $m' \in e$ participate in; then, they indicate whether a mention partipates in the first, second or third most common schema (for that entity).

Diao and Jiang (2013) present a user-based topic model that jointly models both "event" tweets (those concerning/of interest to many users that change over time) and "topical" tweets (those of a personal nature and of interest to a very small set of users). The authors assume every user has a tendency toward personal or event tweets, in addition to differing propensities for a fixed number of cross-user personal topics. Tweets are grouped into epochs, such that in every epoch, events — drawn from a recurrent CRP — are either novel or are taken from the previous epoch. However, to force few events to survive beyond multiple epoch, they include duration regularization via Bernoulli pseudo-observations: they "observe" a (constant, binary) random variable, whose stochasticity is coerced from the sampling parameter, which is dependent on both epoch and user-specific latent bias variables. Posterior inference is done by block Gibbs sampling for discrete variables and gradient ascent for continuous variables. Using 650K tweets over a three month span from 150K Singaporean tweeters, they find joint modeling yields better event and personal coherence over a postprocessing method.

Do et al. (2011) present an ILP for identifying causation among textual events by optimizing over latent connections between event predicates and discourse connec-

tives. To appropriately weight their latent variables, they first use a combination of information retrieval-inspired similarity measures to quantify associations among event predicates (both verbal and nominal) and their arguments, obtained from 760,000 automatically parsed Gigaword documents. Second, they further make use of the Penn Discourse Treebank and a discourse parser to perform discourse connective analysis. Evaluating on a newly-made dataset, they found their ILP setup was able to increase both recall and precision over simpler PMI-based systems.

Huang et al. (2016a) present a pipelined approach to event induction that uses ontological knowledge to learn type-aware clusterings of predicates and their likely arguments. They identify possible candidate event triggers as licensed by OntoNotes's word sense disambiguation and FrameNet's lexical unit dictionary; they represent these triggers with sense-disambiguated word vectors that themselves have been learned using OntoNotes and WordNet. A candidate trigger's arguments are extracted according to some observed structure centered around that trigger: either an AMR, FrameNet, or Stanford dependency parse. They define an autoencoder that recursively composes embeddings of event triggers, their arguments, and relations between triggers and arguments according to this observed structure. Based on their representations, they then iteratively cluster arguments for each trigger and use these clusters to refine each trigger's relation's argument preferences. They label each trigger cluster with the name of the trigger closest to that cluster's centroid, and perform manual mappings for relations.

Schlachter et al. (2017) consider the problem of learning event structure from novel, location-based unstructured text input. They apply Chambers and Jurafsky (2011)'s PMI clustering to documents focused on two main event types: social protests and providing aid to those in need. Due to the focused nature of the study, they apply the clustering to a small corpus—roughly 6,000 in-domain documents. Through domain expert evaluation, they find they learn event templates that are generally coherent templates. These findings are (implicit) evidence for the difficulties in scaling up event extractors large-scale and from general text.

3.4 Extended Comparisons of Event Representations

This section contains a number of in-depth discussions and comparisons of the different event theories and representations covered in this chapter. The material here is meant to be supplementary: it is not a prerequisite for any future chapters in this thesis and can safely be skipped.

3.4.1 Hobbs on Eventuality Individuation and Verification

There are still issues regarding eventuality individuation and identity verification in belief propositions. First, if Bill is fifteen years old, we might say

(3.11) "Bill is almost a man"

$$\text{ALMOST}(E) \wedge man'(E, B)$$

but surely we would still like to make the inference, from E, that "Bill is a human." Unfortunately, given ALMOST's opacity requirements, eventualities are finely individuated; it is not clear how to realize this inference (Hobbs, 1985, pg. 10).

There are many situations in which people use ambiguous references. This can often happen in reporting contexts, when people report on other eventualities. In these cases, people routinely use different surface forms to refer to the same individuals, e.g., "the White House" rather than the name of the White House press secretary, and the same surface form to refer to significantly different entities (e.g., THE WHITE HOUSE as "mansion" vs. "front security gates" vs. "spokesperson").

3.4.2 Expressiveness of Episodic Logic

One question to ask is if we even need $\star$ and $\star\star$ in Schubert (2000)'s episodic logic. Schubert argues partial descriptions are not able to capture causation, but full descriptions may be too strong to be used as meaning postulates, axioms required

for inference. Having one operator, but not the other, may prevent valid logical entailment (Schubert and Hwang, 2000, pg. 4).

Partial descriptions allow general information to be encoded, while full descriptions jump-start the inference process. While Schubert's logic is beyond first order, the most significant barriers are axiom and meaning postulate population. Specifically, the $\star\star$ operator allows a distinction between specific actions/events and generic kinds of actions/events. This means that there is a mechanism for talking about the typicality for various types of situations. This does not exist under Hobbs, at least not as presented in Hobbs (1985). One could possibly add a GENERIC operator, similar to ALMOST, but it's not clear what complications that would introduce.

From a practical perspective, there are issues regarding data availability, as well as identifying and teasing out $\star$ sentences from $\star\star$ sentences; $\star/\star\star$ distinctions may be too strong when true entailment is not needed (e.g., RTE). This partial/full distinction may be crucial to proper judgments dependent on deeper reasoning, such as COPA (Roemmele et al., 2011)).

I explained earlier how $\star\star$ and $\star$ allow a single episodic variable to encompass complex collections of actions and events. Another way to see this difference is via the modifier *almost*. Recall Hobbs defined ALMOST on eventuality variables, which was the only connection between the modifier and the modified. Under Schubert, the modifier operates directly on the modified predicate, with a descriptive situation variable scoped over the entire nested formula. This allows intuitive scoping and LF

structure.

This nested structure also appears in statements about belief as well. Rather than adopt Hobbs's flat structure, Schubert's produces an analysis with proper scoping of "belief" situation variables.

3.4.3 Temporal Predicates in HLF and EL

Operationally, $\star\star$ and $\star$ are defined as a scoping and argument transformation on FOL formulas. In the most basic case, they dictate that *fluent* predicates, which roughly correspond with temporal (both telic and atelic) predicates whose first argument η is situational/episodic, get mapped to a predicate with η removed. Applied at a larger scale, FOL formulas with some properly scoped situational variable e

$$\exists x, y.worship(e, x, y),$$

become *situation abstracts*

$$\exists x, y.worship(x, y).$$

Now, e is operationally bound beyond the formula. As Schubert describes, this abstracts the (high-level) situational description from the lower-level predicates, weakening the innermost constraints for predicate satisfication. This can be seen as a form of back-off, or possibly parameter sharing, which may ease statistical knowledge acquistion, given proper LFs.

How does the fluent to situation abstract transformation differ from Hobbs's primed notation? First, while Hobbs's transformation was bidirectional, Schubert's is not (at least not obviously so): transforming a fluent predicate to a situation abstract does not imply any "actualness" about the eventuality, which would be encoded within a higher-scoped restrictor. This issue of higher-scoping restrictors complicates transforming a situation abstract back to a fluent predicate (or first order formula). Fluent predicates and formulas also allow formulation of outward/inward persistence, which we can think of as the generalization of upward/downward entailments to temporal sub/supersumption.

3.4.4 Discourse and Inference

Hobbs, Lascarides and Asher all require an ability to perform defeasible inference. One of the consequences of Prasad et al. (2008)'s theory neutral approach in the Penn Discourse Treebank concerns defeasible inference. Theoretically, if we could get the appropriate logical forms *and* the defeasible rules, then we could use whatever appropriate inference mechanism we have to get the larger semantic meaning. So, it seems that by not placing their effort with one particular theory camp, the PDTB is rather flexible.

But the above assumes a lot. The one that should give greatest pause is achieving broad coverage defeasible inference rules. As both Hobbs and Lascarides and Asher showed, these defeasible rules interact with the discourse relations. It may be difficult

to compensate for these rules in any practical setting.

Prasad et al. (2008) annotate *senses* of the discourse relations too. While it was motivated analogously as WSD is motivated, they created a sense hierarchy three levels deep. From a practical perspective, they found this kept interannotator agreement/reliability higher, and it may allow sense inference to adapt to either certain observed data. However, this sense hierarchy is important for this paper: recall that Lascarides and Asher (1993) focused on five discourse relations, with a couple being similar, and others being "dual." Meanwhile, Hobbs assumed there would be some discourse relation, but did not go into details. The question remains, how well do the hierarchies match up?

Does the sense hierarchy give a better breakdown? Qualitatively, by annotator agreement scores, yes, it was better. The hierarchy was linguistically informed in its making: they defined each of the levels with a particular purpose. The highest level aimed to capture a main class type. The next level is a refinement, and the third is meant to define the added-semantic-value of each argument.

Chapter 4

Concretely Annotated Corpora

NLP systems often rely upon the output of existing systems; for instance, many of the efforts discussed in §3.3, such as Irwin et al. (2011), Reichart and Barzilay (2012), and Huang and Riloff (2013) use syntactic and entity coreference annotations as assumed input for event-based information extraction. Obtaining these annotations is often considered to be part of the initial preprocessing—i.e., an uninteresting technicality. However, this preprocessing often limits reproducability, as it tends to be a silent first step, done as needed and not shared with the community. Preprocessing systems that produce "deeper" annotations tend to be more compute intensive: as datasets become larger, and deeper systems are used, this initial preprocessing can directly affect labs with limited resources. Finally, and more from a technical and user perspective, these prior efforts are developed independently, in different (programming) languages, and reading and writing different file formats. Systems may define

what "tokens"—often a key, initial step—differently, leading to difficult-to-align annotations. Due to even something non-NLP related such as formatting idiosyncrasies, something basic like "get all words from Wikipedia" can be tricky to get right.[1]

In this chapter, I examine a solution to the above that I helped propose in Ferraro et al. (2014). This solution, termed Concrete, maps common NLP annotations into a structured and documented schema while providing programmatic polyglot access. I also explore large "Concretely" annotated corpora—i.e., corpora that have been annotated with a number of different tools and serialized in the Concrete schema. These corpora are instrumental for the remainder of this thesis.[2]

At its core, Concrete is a data schema that is meant to facilitate the development of human language technology research and tools. Researchers have consistently relied on data; one might ask what makes Concrete germane to today's interests. That is, why was there not a Concrete (or Concrete-like approach) achieved until now? Schemas and tools arise to address particular problems and goals. As covered in §4.3, Concrete is not the first data schema, and it will probably not be the last.

I argue that what makes Concrete (and related resources) achievable today is a confluence of a number of factors. First, the availability of a sufficient number of tools of a sufficient quality: these tools are noisy, and for the foreseeable future, they

[1] If one is looking to replicate another's output, there can be other issues. For instance, how deterministic are those preprocessing tools? As systems are improved, updates may be released: how do those updates affect downstream tasks?

[2] The first portion of this chapter, §4.1, summarizes Ferraro et al. (2014). The second portion, §4.2, provides novel analyses.

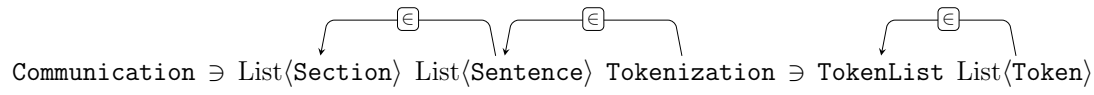
will remain noisy. So the question is less of how accurate are the tools, and more are the tools accurate enough? While more accurate tools will most likely be desired, the current generation of (non neural) tools are, as evidenced by this thesis, accurate enough. Second, the availability of a lot of high-powered (many CPU, large memory) commodity computers makes large scale processing feasible to complete without a large supercomputer. That is, not only do we have a sufficient number of tools to run, but we also have the resources available to actually make use of all of those tools. Third, and perhaps most importantly, there is an ability and willingness to analyze and process language more holistically than there has been over the past two decades. Concrete is designed to facilitate that holistic analysis. This includes

1. the multitheory annotation ability, allowing multiple annotations from the same kind of tool;

2. the inclusion of annotation types for both speech and text data;

3. being agnostic to the programming language tools are developed in; and

4. programmer- and user-aware definitions, safe-guards, and utility libraries, such as the programmatically-defined schema itself, a type system defined once at "compile-time", and libraries that ensure proper serialization and deserialization.



(a) The basic hierarchy-preserving CONCRETE nested structures.

`TokenTagging`
`Parse`
`DependencyParse`

(b) Some label-based, token-level CONCRETE objects.

`CommunicationTagging`
`EntityMention`
`Entity`
`SituationMention`
`Situation`

(c) Some semantic- and discourse-level CONCRETE objects.

Figure 4.1: Some of the defined CONCRETE types, showing structure-describing objects (Figure 4.1a), token-level labellings (Figure 4.1b), and semantic objects (Figure 4.1c.)

4.1 Concrete

CONCRETE, proposed by Ferraro et al. (2014), is a typed schema for multimodal linguistic annotations; Ferraro et al. (2014) also released automatically obtained CONCRETE-annotations on millions of structured documents. The schema allows for multi-level annotations, including token-based ones like part-of-speech, and named entity recognition; tree- and graph-based ones like syntactic and semantic parsing; document-level annotations like entity coreference and event detection; and corpus-level annotations like entity linking.

From a user's perspective, CONCRETE provides direct, programmatic access to the data for a number of programming languages (Java, Python, and C++ among them), allowing users to *do* something with their data rather than learn (and debug)

potentially arcane markup formats. CONCRETE forms the data backbone for more than twenty active and published projects.

4.1.1 Some Basic Types

In Figure 4.1 I outline some basic CONCRETE types; many of these can be cross-referenced with one another through unique universal identifiers (`UUID`); from a programming perspective, these ids act as pointers to follow (or foreign keys to use in joins). In Figure 4.1a there are structure-defining objects: these allow users to ingest hierarchical and structured inputs, such as multi-section Wikipedia articles, and retain the structure.[3] A `Communication` refers to a document; just as a document has paragraphs, or areas of interest, so does a `Communication` have a list of `Section`s. Each `Section` has a list of `Sentence`s, which are light-weight wrappers around `Tokenization`s.[4] A tokenization is defined in terms of a `TokenList`, which grounds out in the actual `Token`s.[5]

In Figure 4.1b I list some tokenization-/word-level objects while in Figure 4.1c I list some semantic/document-level objects; these are all particularly relevant to this thesis. In contrast to the structure-defining objects of Figure 4.1a, these an-

[3]Some users may not need or want this additional structure. In this case, destroying is easier than creating: it is easy to iterate over the nested structures, effectively pretending it does not exist, on the fly. It is much harder to impute missing structures.

[4]Though all annotations are defined with respect to a `Tokenization`, separating `Section`s from `Tokenization`s allow the tasks of sentence segmentation and tokenization to (optionally) remain separate.

[5]To support automatic speech recognition, machine translation and text normalization, a `Tokenization` can *alternatively* be defined by a `TokenLattice`, representing a (non-trivial) weighted finite-state machine.

notation objects can appear multiple times within a `Tokenization` (Figure 4.1b) or `Communication` (Figure 4.1c). This reflects the fact that different systems can be trained to produce different labels and label types. For example, `TokenTagging`s are sequences of token-level tag labels, as would be used for part-of-speech tagging or named entity recognition, while `CommunicationTagging`s are collections of document-level labels, as would be used for document classification.

The final relevant semantic-level objects are `Entity`- and `Situation`-based ones. An `EntityMention` is a span (generally of `Token`s) within text that refers to *something* (nominal) with a presupposed existence while an `Entity` is a collection of `EntityMention`s that all corefer to that *something*. In Figure 4.2, each of the two blue "Clinton"s is an `EntityMention`, while both mentions together form an `Entity`.

Drawing on terminology from chapter 3, Concrete defines the same types of structures for "events" or "situations:" a `SituationMention` refers to some *situation* while a `Situation` is a collection of coreferring `SituationMention`s. `SituationMention`s, like `EntityMention`s, can ground out directly in `Token`s. Unlike `EntityMention`s, though, `SituationMention`s can ground out in `EntityMention`s or recursively in *other* `SituationMention`s. I discuss mapping common NLP event/semantic tasks to Concrete in §4.1.2.

Concrete tries to be *additive*: when possible, it does not remove information or metadata. This philosophy can result in conflict between analytics and the (non-curated) data one wishes to run them on: analytics often place requirements on their

Clinton and Congress agreed on a plan. He said Clinton would try the same tactic again.

Figure 4.2: Contrasting entity mentions vs. entities. Each of the blue "Clinton"s is an entity mention: it is an instance of a text span referring to *some* being. Assuming both "Clinton"s refer to the *same* being, then taken together they form an entity. Meanwhile, the red "He" is both an entity mention (non-specified anaphor) and, if this excerpt stands alone, a singleton entity.

input but historically these requirements can be destructive to the input text. A key *practical* challenge is to enable this additive philosophy in a user-friendly way. In Concrete, the structure-describing and annotation objects are based off of the (possibly destructive) tokenization, but they can ground out in the *original* text through code-point offsets. These code-point offsets allow two different tokenizations to be "merged" if necessary. For example, a system must retokenize input but the evaluation is defined with respect to another; grounding the annotations out in the original easily allows for that merge to happen.

4.1.2 Mapping Semantics to Concrete

Given how central events are to this thesis, in this section I describe how to map some common NLP events into Concrete. As described above, the primary entry method to describe "events" is through `SituationMention`s. I highlight aspects of how `SituationMention`s are actually defined in Figure 4.3.

Consider labeling the "agreed" predicate of Figure 4.2: a reasonable FrameNet

`SituationMention`

Field Name	Type	Description
`argumentList`	List⟨`MentionArgument`⟩	A required list of arguments, defined as `MentionArgument`s (see below). This required list can be (explicitly) empty.
`situationKind`	string	A label describing the general kind of situation. These labels are schema-specific. For FrameNet, this would be the frame name.
`tokens`	`TokenRefSequence`	A robust method to ground the situation *trigger* in a specific `Tokenization` and `Token`s.
`text`	string	An easy way to refer to the entire situation instance (the full, or representative, text).

`MentionArgument`

Field Name	Type	Description
`role`	string	A (schema-specific) label describing the role. For FrameNet, this would be the frame element.
`entityMentionId`	`UUID`	A UUID pointing to a particular `EntityMention` that this argument grounds out in. Either this or `situationMentionId` should be set.
`situationMentionId`	`UUID`	A UUID pointing to a particular `SituationMention` that this argument grounds out in. Either this or `entityMentionId` should be set.
`tokens`	`TokenRefSequence`	A catch-all method to ground the argument in a specific `Tokenization` and `Token`s, when grounding in other `SituationMention`s or `EntityMention`s is neither appropriate nor practical.

Figure 4.3: An overview of how `SituationMention`s are defined. For space, I have not included the definition of `TokenRefSequence`. Please see the documentation: `http://hltcoe.github.io/concrete/schema/`.

analysis would trigger the `Make_Agreement_on_Action` frame, while a reasonable PropBank analysis may label it with an `agree-v-1` roleset. Two fundamental parts of describing a situation are to describe its participants and any (human-readable) label or summary. The former is handled by a required list of `MentionArgument`s, described in more detail below; to accommodate zero-arity or self-filling events, as in FrameNet, this list can be empty (Baker et al., 1998). The latter is handled in a number of different ways. First, a single kind (label) for the situation can be stored as a string in `situationKind`: this could be the FrameNet name `Make_Agreement_on_Action` or the PropBank roleset `agree-v-1`. Second, we can describe the situation in an indexable, machine-usable way through the tokens that actively trigger it; in this case, unique pointers and indices to the single token "agreed."[6] Third, we can describe the *observable* event unambiguously through the `text` field.[7]

`MentionArgument`s describe the participants. In our example Figure 4.2, consider the "Clinton" and "Congress" subjects (distributed across the conjunction). The string field `role` provides a human-readable label for each participant, such as `Party1`/`Party2` for a FrameNet `Make_Agreement_on_Action` or simple (roleset-specific) PropBank `Arg0`/`Arg2` labels. CONCRETE lets us ground each of "Clinton" and "Congress" in multiple ways: as `EntityMention`s or as unique pointers to specific tokens. Note that both ways of doing so are indexable—a prime difference is how a system (or downstream system) needs to interpret a situation's participants.

[6] A `TokenRefSequence` is simply an object for uniquely identifying (spans of) tokens.

[7] This may be the same as the `tokens` field, or it may not be. If it is, then filling this field is a potential courtesy to downstream users. If not, then filling this field lessens ambiguity.

While the third method of grounding participants—as `SituationMention`s—would be difficult to apply to either of the subjects, note that it *could* be applied to the third argument: "a plan." In this case, "a plan" could itself trigger a frame, which would be described as a `SituationMention`. This three-fold method for representing participants allows CONCRETE to encompass many forms of event semantics, both established (like PropBank) and under active development (like the "situation frames" project (Strassel et al., 2017)).

4.2 Annotating Large Corpora

By integrating CONCRETE into well-known and novel NLP tools, Ferraro et al. (2014) created data annotation pipelines that allow millions of documents to be annotated with multiple kinds of annotations.[8] Those pipelines were run on three large NLP corpora: English Gigaword Fifth Edition (Parker et al., 2011), the Annotated New York Times Corpus (Sandhaus, 2008), and the February 2016 dump of English Wikipedia. Together, these three annotated corpora comprise the **Concretely Annotated Corpora** (CAC), an annotated collection of 15,609,083 documents. I provide basic statistics for CAC in Table 4.1.

CAC contains the output of Stanford's CORENLP system (Manning et al., 2014, v3.5.2), the SEMAFOR semantic parsing system (Das et al., 2010, 2014, v2.1), and the

[8] Additional pipelines have been developed too, particularly for multilingual workflows (Peng et al., 2015).

	Gigaword	**Annotated NYT**	**English Wikipedia**	**Total**
Documents	8,739,092	1,810,347	5,059,644	15,609,083
Sentences	196,979,012	70,367,495	154,437,835	421,784,342
Tokens	4,301,121,089	1,401,857,789	2,333,564,265	8,036,543,143
Vocabulary	6,583,281	2,927,830	15,550,696	977,038
Vocabulary (≥ 2)	3,199,503	1,750,890	4,966,271	656,731
Vocabulary (≥ 100)	225,393	119,813	263,636	91,093
Semantic Frames	2,582,976,444	780,262,295	1,055,172,246	4,418,410,985

Table 4.1: Basic statistics for the Concretely Annotated Corpora (Ferraro et al., 2014). Vocabulary totals are intersective.

FNPARSE semantic parsing system (Wolfe et al., 2016, v1.0.6). Per document, these three suites produce: one part-of-speech tagging, one lemmatization tagging, one named entity recognition tagging, one constituency parse, four dependency parses, two sets of (coreferenced) entity mentions, and three semantic parses. All annotations are with respect to the same tokenization and sentence segmentation. These annotations required well-above 150,000 CPU hours to produce.

In the following section (4.2.1), I overview the data contained in CAC that are particularly relevant to event semantics.

4.2.1 Annotations for Events

Although *all* of the annotations in CAC can be used for learning events, the ones that directly represent events are the three semantic parses. Two of those are FrameNet-based from SEMAFOR (Das et al., 2014) and FNPARSE (Wolfe et al., 2016), and one is PropBank-based, also from FNPARSE. Each extracted semantic *frame* is

	Gigaword	**Annotated NYT**	**English Wikipedia**	**Total**
Semafor	1,443,431,194	447,663,603	649,148,073	2,540,242,870
fnparse/fn	639,613,122	184,782,652	227,096,224	1,051,491,998
fnparse/pb	499,932,128	147,816,040	178,927,949	826,676,117

Table 4.2: Frame parses (SituationMentions) extracted contained in Concretely Annotated Corpora (Ferraro et al., 2014).

stored as a SituationMention under the Concrete schema; the full semantic *parse*, or all of a particular tool's semantic frames, is stored as a SituationMentionSet. Throughout, I may refer to them as such too.

As Table 4.1 shows, there are 4.4 billion extracted semantic frames in CAC, with 2.6 billion in the Gigaword portion, 780 million in the Annotated NYT portion, and 1.1 billion in the Wikipedia portion. Table 4.2 provides a further breakdown of these frames, across the three tools. Roughly 70% of the FrameNet annotations are from Semafor.

Figure 4.4 shows log-scale histograms for the number of frames per sentence for the three sub-corpora. Notice the general pattern—Semafor prefers to produce a smaller number of frames per sentence in high quantity, while fnparse produces more frames per sentence, but in lower quantity. The same holds for the number of roles (arguments) per frame, as shown in Figure 4.5.

Church and Hanks (1990) established a tradition of sorts for identifying trends among words within large corpora: first approximate joint and marginal probabilities $p(x, y)$, $p(x)$ and $p(y)$, and then identify and examine those words x and y that yield

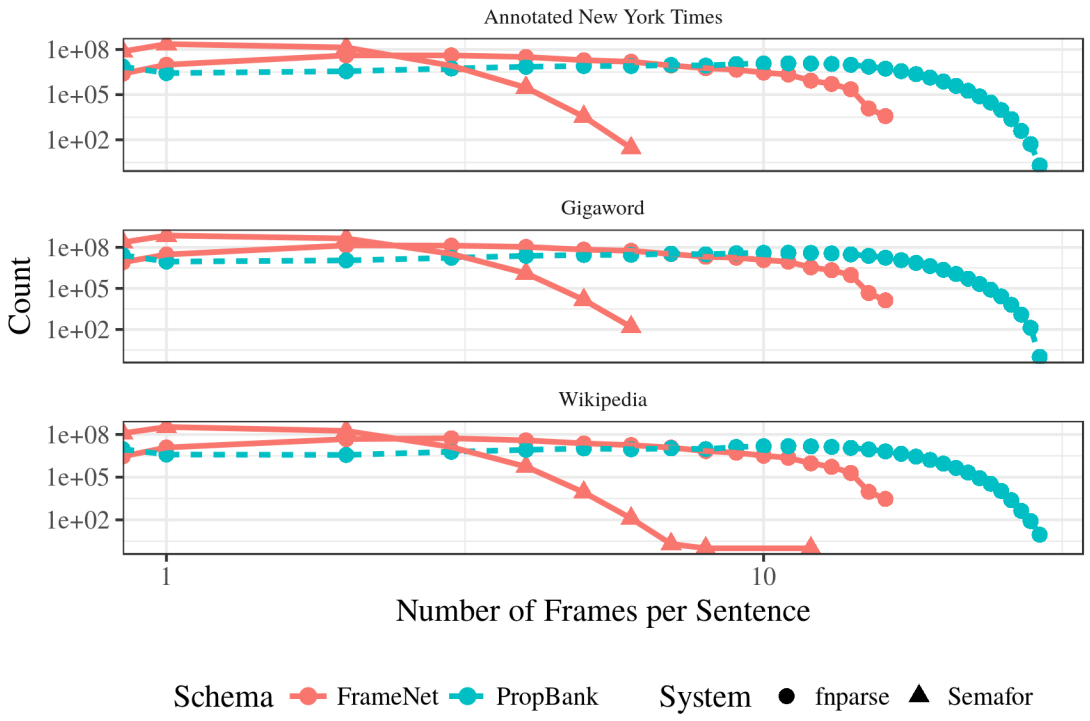


Figure 4.4: Frames per sentence (Ferraro et al., 2014, Concretely Annotated Corpora).

extreme pointwise mutual information (PMI) values:

$$\text{PMI}(x, y) = \log \frac{p(x, y)}{p(x)p(y)}.$$

Roughly, extreme positive values of PMI indicate that x and y are very strongly associated: they occur together much more frequently than chance; on the other hand, extreme negative values indicate x and y are very strongly disassociated. PMI values moderately close to 0 reflect a practical independence of x and y.

obscure	Eclipse	Obscurity	`obscure-v-1`
snap	Breaking off	Cause to fragment	`snap-v-8`
fumble	Bungling	Seeking	`fumble-v-1`
sidestep	Avoiding	Dodging	`sidestep-v-1`
saturate	Being wet	Cause to be wet	`saturate-v-1`
chill	Cause temperature change	Inchoative change of temperature	`chill-v-1`
torment	Cause to experience	Emotion directed	`torment-v-1`
melt	Altered phase	Change of phase	`melt-v-1`

(a) Top PMI for verb triggers.

response	Communication response	Response	`response-n-1`
deal	Be in agreement on action	Make agreement on action	`deal-n-1`
approach	Arriving	Means	`approach-n-2`
speech	Communication	Text	`speech-n-1`
clash	Hostile encounter	Sounds	`clash-n-1`
expression	Encoding	no frame	`expression-n-1`
plan	Project	Purpose	`plan-n-1`
feeling	Feeling	Sensation	`feeling-n-1`

(b) Top PMI for nominal triggers.

cool	Desirability	`no frame`	Temperature
safe	Being at risk	`no frame`	Risky situation
friendly	`no frame`	Sociability	Social interaction evaluation
quiet	Become silent	`no frame`	Sound level
agonizing	Emotion active	`no frame`	Stimulus focus
warm	Ambient temperature	`no frame`	Temperature
unsuccessful	`no frame`	Success or failure	Successful action
cool	Experiencer focus	`no frame`	Temperature

(c) Top PMI for advective triggers.

up	Being up to it	`no frame`	Silencing
back	`no frame`	Remembering experience	Taking sides
later	`no frame`	Relative time	Time vector
worse	Desirability	Morality evaluation	`no frame`
east	Direction	`no frame`	Part orientational
early	`no frame`	Relative time	Temporal subregion
south	Direction	Locative relation	`no frame`
there	Existence	Locative relation	`no frame`

(d) Top PMI for adverb triggers.

Table 4.3: Top PMI values for *Annotated NYT* trigger and differing frame cooccurrence. Frame triggers are italicized (e.g., *obscure*, *expression*, *worse*), FrameNet labels are in small caps (e.g., Eclipse, Encoding, Desirability), and PropBank labels are monospaced (e.g., `obscure-v-1`, `expression-n-1`). Special "no frame" labels indicate that the FrameNet or PropBank systems did not predict a frame for the provided trigger. The frame labels are alphabetized across each row.

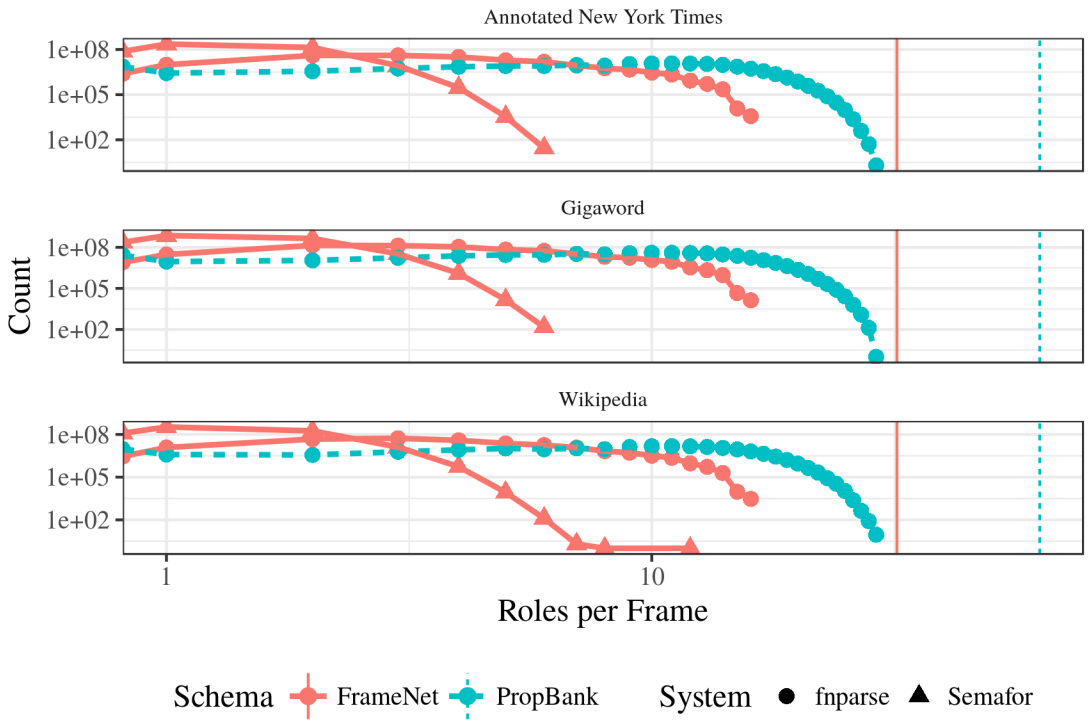


Figure 4.5: Roles per frames (Ferraro et al., 2014, Concretely Annotated Corpora). The vertical lines represent the maximum number of *possible* role labels according to each schema. (According to Wolfe (2017), the exact number of PropBank roles depends on whether one includes referential and continuation roles. The count displayed inthis graph does.)

With the straight-forward generalization of PMI to higher order cooccurrences, the Church and Hanks strategy is an effective, fast method for qualitative examination of frames, despite their automatic and noisy generation. In 4.3a to 4.3d, I show predicates (triggers) and all frames (`PropBank` and both FrameNet) triggered from the *Annotated NYT* portion of CAC that yielded high (positive) PMI where the two FrameNet frames were different. These are stratified according to whether the trigger

was verbal, nominal, adjectival or adverbial, respectively.[9]

Overall, we see that even though differences in utilized frames do appear, there is nevertheless a general consensus among the three frame systems. For instance, examining verbs (Table 4.3a) we see CAUSATIVE OF and INCHOATIVE OF alternations dominate the top PMI (Ruppenhofer et al., 2006). Beyond the linguistic alternations, we also see the nuances of the schemas throughout. This is most noticeable when examining the verbal and nominal triggers: the *deal* row in Table 4.3b has FrameNet frames BE IN AGREEMENT ON ACTION and MAKE AGREEMENT ON ACTION. According to the FrameNet specification, the former has a formal USES relationship with the latter. Meanwhile the nominal *response* row reflects the frame hierarchy of FrameNet: COMMUNICATION RESPONSE inherits from RESPONSE. The adjectival and adverbial frame triggers reflect an unfortunate training data sparsity with respect to PropBank: these kinds of triggers are not labeled, and thus do not appear in downstream annotations. Finally, notice that these frames capture, by construction, antonymy: for example, *safe* triggers `Being at Risk`.

[9]The joint and marginal probabilities were estimated under the following restrictions:

1. The triggers and frames must have occurred at least 500 times, and
2. 0.01 was reserved from the joint distribution for *all* unseen word-frame tuples. In effect, this is a trace amount.

4.3 Related Efforts in Data Serialization

One may argue that CONCRETE is not *really* agnostic to the programming language, in the same way that tab-separated offset annotations (as used in many CoNLL shared tasks, for example) are. In the sense that CONCRETE cannot be used by a developer using a language's base library *only*, such as the Standard C, Java, Python or C++ libraries, and assuming that the standard libraries can incrementally read text from a file, that criticism can be considered valid. After all, CONCRETE is agnostic to the end-user's programming language, provided appropriate serialization utilities have been written in that language. While this may be an initial barrier of entry to future programming languages not-yet written—or to existing, but more esoteric ones—note that there are actively maintained libraries for most of the common, current programming langages.[10] These libraries are open-source, have community support, and provide an **automatic** way of turning the schema into usable code definitions. Therefore, I argue that CONCRETE is *de facto* programming language agnostic, even if some initial effort must be invested up-front to access in a new programming language.

Moreover, in contrast to many of the built-in or standard serializations formats—such as serialized Java objects, Python pickles, or Boost's serialization standards for C++—CONCRETE, and in particular the CAC, uses a single, common binary format that can accessed by any programming language where utilities to serialize Thrift

[10] An up-to-date list of core Thrift libraries can be found at `https://thrift.apache.org/`.

exist. A developer working in Python does not have to write separate functions to read CONCRETE data produced in Java or in C++—they all write to the same format.

Of course, the goal of sharing data produced in one programming language with a tool written in another is not new. There *are* standard data formats (or families of data formats) like XML (eXtensible Markup Language) and JSON (JavaScript Object Notation). These are often well-known among developers and are well-supported by standard programming libraries. However, there are two key differences between XML or JSON and CONCRETE. First, CONCRETE provides a "compile-time" schema: the language-specific definitions for CONCRETE objects are specified automatically from the schema. This places a type of syntactic contract on end-users: improper access to fields should be caught at compile time (or during a *linting* phase for non-native and interpreted languages) of the user's code. However, the meaning of a CONCRETE object is based in its internal (syntactic) representation; syntactic validations and verifications can then result in semantic validations. This is in contrast to JSON, and even XML; though XML offers a schema definition, any improper use of that schema can only be detected at run-time.

A second key difference between CONCRETE and XML or JSON is that CONCRETE is grounded in type-based, programmatic access. Whereas XML often relies on run-time declarative access, with some possible post-filtering, CONCRETE allows the user to directly access elements of an object. For example, accessing the second sentence of the third paragraph of a `Communication comm` could be accomplished as

4.1 (in the style of C++ or Python), while accessing the same in an XML document might be accomplished as 4.2:

(4.1) `comm.sectionList[2].sentenceList[1]`

(4.2) `comm.getElementsByTagName(‘‘section’’)[2]`

`.getElementsByTagName(‘‘sentence’’)[1].`

While these two versions seem very similar, there are two major differences: first, the verification that a list of `Section` being a proper element of a `Communication` is only verified in 4.2 at run-time, while in 4.1 a compiler or linter should likely verify that `Communication` does a defined list of `Section` child. Second, 4.2 is technically only correct because `Section`s are only defined as elements of a list, which is directly contained within a `Communication`. If, for example, one user placed *additional* `Section`s into the "catch-all" `keyValueMap` field of a `Communication`, then any later users who accessed `Section`s via 4.2 would be operating on an incorrect data set. Finally, this is all in contrast to JSON, which can only store a limited set of native types.

Concrete is a method for storing and accessing language annotations. This makes it different from annotation *tools* and *frameworks* like BRAT (Stenetorp et al., 2012). Concrete and BRAT can, in theory, work symbiotically: Concrete-backed data could be fed into a BRAT annotation tool, where new language annotations could be added, or existing ones verified or fixed. These annotations could then be merged or added to the existing Concrete data, which could then be processed by any number of Concrete-based analytics. The conversions between Concrete and

BRAT data could be handled once, thereby (1) limiting the number of different data formats an analytic must support; (2) reducing the potential introduction of bugs, especially around any corner-case limitations of the BRAT format;[11] and (3) allowing new annotations to be added easily, without disrupting other developers' workflows. Finally, Concrete has built-in support for audio annotations, allowing tools to consider data beyond just text.

Language technology engineers and developers may be familiar with Unstructured Information Management Architecture (Apache UIMA Community, 2013, UIMA, originally developed at IBM in the early 2000s) or the General Architecture for Text Engineering (Cunningham et al., 1995; Cunningham, 2014, GATE). Both are based around *building* new tools and analytics; to that end, they both require some stable data representation. However, they are more akin to *services* in the Concrete and Thrift realms, while a core functionality of Concrete is representing and storing data in ways that make it easy for NLP researchers and developers to work together. Note that GATE defines a type-schema in XML, while UIMA defines types directly within a Java hierarchy.

Unlike Concrete, UIMA and GATE have their core definitions and functionality defined in Java. While work-arounds like foreign-function interfaces and interprocess communication allow non-Java analytics to be written and interact with UIMA/-GATE tools, this can represent a high software engineering barrier to entry.

[11] As of publishing, BRAT specified annotations as offset annotations in a tab-separated format. Spans of text containing a tab must be properly handled before annotation.

Concurrent with Ferraro et al. (2014) were a number of research efforts into large-scale NLP analytics (Ide and Grivolla, 2014). One of the most related was a toolsuite and framework called DKPro and DKPro Core (Eckart de Castilho and Gurevych, 2014). Based in UIMA, DKPro Core allows for NLP pipelines to be efficiently created and run. As before, this is more akin to defining and developing an environment for *services* or analytics.

Note that, as a whole, the efforts described by Ide and Grivolla (2014) suggest that is sizable interest and need for developing large-scale NLP and language engineering systems and workflows. The data representation and contract provided by Concrete is one such extendable solution.

4.4 Summary

In this chapter I have described Concrete, a data schema that both stores human language annotations in a type-safe manner and that allows these annotations to be used by many different users, NLP systems, and programming languages. The schema provides for annotations at the token level, such as part of speech tags; at the syntactic/sentence level via structures like dependency parses; at the semantic level, such as with FrameNet and PropBank semantic parses; and at the discourse level, such as with entity and event coreference.

The entire schema is defined in a (programming) language agnostic fashion: rather

than storing, e.g., Java or Python serialized objects, which each have their own binary format, CONCRETE stores its annotations in a single, well-documented binary form. Data are then accessed via utility libraries that can be written for each additional programming language. These libraries already exist for many of the common languages today, such as Java, Python, C++, and JavaScript. However, as new programming languages are used, developers can follow the documentation of the internal format; see `https://thrift.apache.org/` for additional information.

Active development on CONCRETE continues to advance, with more tools and additional data structures and methods being added. These more recent and future developments may be found at `http://hltcoe.github.io/concrete/`.

While I argue that CONCRETE, and the preprocessed data associated with it (the CAC, see §4.2), are themselves beneficial to the NLP community, the data have particular relevance to the remainder of this thesis. In §4.1.2 I demonstrated how semantic NLP representations can be mapped into CONCRETE, and in §4.2.1 I provided an initial, high-level analysis of billions of semantic parses. The CAC will continue to be used extensively in experiments throughout this thesis.

Chapter 5

Frame-Based Attributive Embeddings

Recall from chapter 3 that a common strategy among computational event researchers is to manually annotate some amount of natural language with a schema grounded in theoretical or logical representations, in order to inspire the community at large to create systems that can then automatically perform that annotation on new data. Often, this new data is the fixed test data originally provided by the human annotators. When systems *are* run on truly novel data, the resulting annotations are almost always used as supplementary features within a larger classification system or downstream task; these new, automatically obtained annotations are rarely examined on their own.

I first present an overview of the available data, and then provide multiple intrinsic

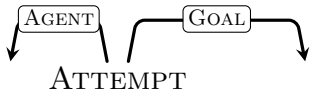


Figure 5.1: A simple frame analysis.

analyses. The first demonstrates the generalizations that can be achieved with these noisy models; the second examines how to use these annotations to better capture computational linguistics annotations; and the third examines how to better model cognitive data. The data (chapter 4) used in this chapter will be used throughout the remainder of this thesis.[1]

5.1 A Method for Continuous Lexical Semantics via Vectors and Frames

In this section, I detail a tensor factorization method for learning word embedding. The aim is to provide a straight-forward approach that can leverage arbitrary, joint observation counts. I will demonstrate this with joint frame and role counts from CAC in §§ 5.3 and 5.4.

Consider "Bill" in Figure 5.1: what is his involvement with the words "would try," and what does this involvement *mean*? As covered in chapter 3, an approach based

[1]This chapter is an extended version of Ferraro et al. (2017). All of §5.4, and many of the qualitative analyses of §5.3, are novel.

in *frame semantics* generalizes word meanings to that of analyzing structured and interconnected labeled "concepts" and abstractions (Minsky, 1974; Fillmore, 1976, 1982). These concepts, or roles, *implicitly* encode expected properties of that word. In a frame semantic analysis of Figure 5.1, the segment "would try" *triggers* the ATTEMPT frame, filling the expected roles AGENT and GOAL with "Bill" and "the same tactic," respectively.

While frame semantics provide a structured form for analyzing words with crisp, categorically-labeled concepts, there are open questions. For instance, the encoded properties and expectations are implicit: what does it *mean* to fill a frame's role? Of course, there are also a number of potential issues: how many of these categorical concepts are appropriate? How robust are frames and concepts to particular domains or end-goals?

Word embeddings present an alternative to the categorical approach. The idea behind word embeddings is to represent meaning as points in a real-valued vector space rather than categorical collections (Deerwester et al., 1990; Mikolov et al., 2013a). These representations derive meaning from the distributional hypothesis: the notion that words are defined by how they "interact" with other words (Harris, 1954; Turney and Pantel, 2010). Typically, then, representations are learned by exploiting the frequency that the word cooccurs with contexts. These contexts are often just the surrounding words within a user-defined window, e.g., those words two to the left and right of a particular target word. When built from large-scale sources, like

Wikipedia or web crawls, word embeddings capture general characteristics of words and allow for robust downstream applications (Kim, 2014; Das et al., 2015, i.a.).

5.1.1 Skip-Gram

Although word embedding methods have been part of the modern computational linguistics literature since Deerwester et al. (1990), Mikolov et al. (2013a)'s `word2vec` methods—skip-gram (SG) and continuous bag of words (CBOW)—significantly re-popularized these methods. I focus on SG, which *predicts* the context i around a word j, with learned representations $\mathbf{c}_i$ and $\mathbf{w}_j$, respectively, as

$$p(\text{context } i \mid \text{word } j) \propto \exp\left(\mathbf{c}_i^\intercal \mathbf{w}_j\right) = \exp\left(\mathbf{1}^\intercal(\mathbf{c}_i \odot \mathbf{w}_j)\right),$$

where $\odot$ is the Hadamard (pointwise) product, and traditionally, the context words i are those words within a small window of j.

Note that predicting a certain context i from word j implies updating not only the parameters associated with i and j, but also the parameters for *all other* contexts. To make this efficient, embeddings are generally trained with an approximation called negative sampling (Mikolov et al., 2013b; Goldberg and Levy, 2014). With negative sampling, you sample a small (2-20, typically) set of other, "incorrect" contexts. Then when you predict i from j, you pretend that i is predicted from this reduced set. This means a very small number of parameters need to be updated at each step.

5.1.2 Skip-Gram as Matrix Factorization

Levy and Goldberg (2014b), and subsequently Keerthi et al. (2015), showed how vectors learned under SG with the negative sampling are, under certain conditions, the factorization of (shifted) positive pointwise mutual information. Cotterell et al. (2017) show that SG is a form of exponential family PCA that factorizes the matrix of word/context cooccurrence counts (rather than shifted positive PMI values). With this interpretation, they provide both a way to generalize SG from matrix to 3-tensor factorization, and a theoretical basis for modeling higher-order SG (or additional context, such as morphological features of words) within a word embeddings framework.

Specifically, Cotterell et al. recast higher-order SG as maximizing the log-likelihood

$$\sum_{ijk} \mathcal{X}_{ijk} \log p(\text{context } i \mid \text{word } j, \text{feature } k) \tag{5.1}$$

$$= \sum_{ijk} \mathcal{X}_{ijk} \log \frac{\exp\left(\mathbf{1}^\intercal(\mathbf{c}_i \odot \mathbf{w}_j \odot \mathbf{a}_k)\right)}{\sum_{i'} \exp\left(\mathbf{1}^\intercal(\mathbf{c}_{i'} \odot \mathbf{w}_j \odot \mathbf{a}_k)\right)}, \tag{5.2}$$

where $\mathcal{X}_{ijk}$ is a cooccurrence count 3-tensor of words j, surrounding contexts i, and features k. Negative sampling can be applied here too.

5.1.3 Skip-Gram as n-Tensor Factorization

When factorizing an n-dimensional tensor to include an arbitrary number of L annotations, I replace *feature* k in Equation (5.1) and $\mathbf{a}_k$ in Equation (5.2) with each annotation type l and vector $\boldsymbol{\alpha}_l$ included. $\mathcal{X}_{i,j,k}$ becomes $\mathcal{X}_{i,j,l_1,\ldots l_L}$, representing the number of times word j appeared in context i with features l_1 through l_L. The objective to maximize is

$$\sum_{i,j,l_1,\ldots,l_L} \mathcal{X}_{i,j,l_1,\ldots,l_L} \log \beta_{i,j,l_1,\ldots,l_L}$$

$$\beta_{i,j,l_1,\ldots,l_L} \propto \exp\left(\mathbf{1}^\intercal(\mathbf{c}_i \odot \mathbf{w}_j \odot \boldsymbol{\alpha}_{l_1} \odot \cdots \odot \boldsymbol{\alpha}_{l_L})\right).$$

The following sections examine this method and some of the types of *enumerative* and *elicitable* semantic knowledge it can capture.

5.2 Evaluating Embeddings

Though there are a number of methods for evaluating learned representations, most rely on correlating some type of human judgment about pairs of words with Euclidean properties of those words' embeddings (e.g., the dot product between two vectors). Often these judgments are couched in some notion of word association or similarity. For example, we can say that the pairs (1) "kill" and "knife," (2) "kill" and "arrest," and (3) "kill" and "assassinate" can be thought of as being highly

"associated" or "similar"—though for different reasons.[2]

Standard word "similarity" datasets first curate a list of word pairs, and then gather (and average) human judgments for how "related" each pair is.[3] Word vectors are evaluated against these judgments by finding Spearman's ρ between the human judgments and embedding dot products.

Word "similarity," though, has a number of limitations. While methodological issues—such as the word pairs themselves, or how humans are asked to judge them—can be a concern, the coarse judgments obscure *why* certain words are similar or related. Even when word pairs are selected or presented to humans based on certain properties, such as how concrete or abstract each word in, these stratifications are not reflected in the final judgment (Hill et al., 2016).

However, Hill et al. (2016) argues that there are deeper issues. Specifically, the concepts of *association* and *similarity* have important psycholinguistic distinctions, but these distinctions are muddled within the computational linguistics community. Among other issues, this confusion results in standard evaluation datasets mischaracterizing what they actually measure.

In the remaining portions of this chapter, I argue that if we are going to be concerned about capturing nuances, either at the empirical or the pyscholinguistic

[2]Generally, associated words have a common domain, use, or some other *semantic* relation. This semantic relation is often reflected by those words commonly occurring together, either in text corpora or in actual (physical) use McRae et al. (2012). Word *similarity*, on the other hand, can be thought of as capturing synonymy.

[3]Either Likert scales or interval scales are used, with the former converted into the latter for evaluation).

level, then different questions need to be asked—both of the research itself, and of any human judges and their responses. If a high-level judgment can be reasoned about through smaller (more atomic) judgments that can be enumerated, then we should record and evaluate against those sub-judgments. As these judgments for each word (word pair) can form a vector in-and-of-themselves, the prior evaluation stategy of evaluating a single response (scalar) vs. a dot product (scalar) does not apply. Instead, we should try to correlate (dimensions of) the two sets of vectors.

QVEC is a method for doing just that (Tsvetkov et al., 2015). QVEC uses canonical correlation analysis to measure the Pearson correlation between $\mathbf{w}$ and the collection of oracle vectors $\mathbf{o}$. These oracle vectors are derived from the human responses to the property decomposition. For QVEC, higher is better: a higher score indicates $\mathbf{w}$ more closely correlates (positively) with $\mathbf{o}$. In the follow sections, I employ QVEC.

5.3 Capturing Semantic Protoroles

One criticism of frame semantics is that the frames and concepts are both defined as discrete items: a particular frame has certain roles, where the *label* ends up carrying the bulk of the meaning. That is, any encoded properties and expectations of that role are implicit. Semantic proto-role (SPR) theory, motivated by Dowty (1991)'s thematic proto-role theory, offers an answer to this. SPR replaces categorical roles

with a cadre of judgments about what is likely true of the entity filling the role.[4] For example, an SPR analysis of Figure 5.1 may talk about how likely it is for Bill to be a willing participant in the ATTEMPT. The answer to this and other simple judgments characterize Bill and his involvement. Since SPR both captures the likelihood of certain properties and characterizes roles as groupings of properties, we can view SPR as representing a type of continuous frame semantics.

In this section, I examine how to capture these SPR-based properties and expectations within word embeddings. I use the tensor factorization method presented in §5.1 in order to learn frame-enriched embeddings from the data described in chapter 4. Overall, I show how to learn word embeddings enriched with multiple, automatically obtained frames from large, disparate corpora; and I demonstrate that these enriched embeddings better capture SPR-based properties.

5.3.1 Extracting Counts

I utilize majority portions of the Concretely Annotated New York Times and Wikipedia corpora from CAC. These have been annotated with three frame semantic parses: one FrameNet from Das et al. (2010), and both FrameNet and PropBank from Wolfe et al. (2016). In total, I use nearly five million frame-annotated documents.

The baseline extraction I consider is a standard sliding window: for each word w_j seen $\geq T$ times, extract all words w_i two to the left and right of w_j. These counts,

[4]Although the number of judgments is unknown, current approaches employ between ten and twenty (Reisinger et al., 2015; White et al., 2016).

forming a matrix, are then used within standard `word2vec`. I also follow Cotterell et al. (2017) and augment the above with the signed number of tokens separating w_i and w_j, e.g., recording that w_i appeared two to the left of w_j; these counts form a 3-tensor.

To turn semantic parses into tensor counts, relevant information from the parses must first be identified. Because SPR characterizes participants of actions, I define and organize the extraction of relevant information around *roles* and *what fills them*. First, I consider all frames that are triggered by the target word w_j (seen $\geq T$ times) and that have at least one role filled by some word in the sentence. This qualification is required, since standard semantic parsing practice allows for frames to be triggered but have no filled roles. However, I *do* allow triggers to be self-filling, i.e., the trigger for a frame also fills a role.

Second, having identified relevant parses and frames, I extract every word w_r that fills all possible triggered frames; each of those frame and role labels; and the distance between filler w_r and trigger w_j. This process yields a 9-tensor $\mathcal{X}$. Specifically, each record (index into $\mathcal{X}$) consists of the trigger, a role filler, the number of words between the trigger and filler, and the relevant frame and roles from the three semantic parsers.

Being automatically obtained, the parses are overlapping and incomplete. I include special ⟨NO_FRAME⟩ and ⟨NO_ROLE⟩ labels as needed so as to completely index $\mathcal{X}$.

5.3.2 Predict Fillers or Roles?

I **always** treat the trigger as the "original" word (e.g., word j, with vector $\mathbf{w}_j$). Since SPR judgments are between predicates and arguments, I train models to **predict the words filling the roles**, and treat all frame and role information as auxiliary features.

On the other hand, because SPR annotations were originally based off of (gold-standard) PropBank annotations, it is reasonable to wonder if predicting PropBank information results in higher SPR-QVEC correlation. Therefore, I also train models to **predict PropBank frames and roles**. In these, I treat the role-filling text and all other (non PropBank) frame information as auxiliary features.

Finally, remember that the FrameNet information comes from two different systems. As observed in Figure 4.4 and Figure 4.5, these two systems produced significantly different annotations. In early development, I found it to beneficial to (1) not distinguish between them, i.e., accept all FrameNet annotations without regard to which system produced it; and (2) not learn correlations between the FrameNet systems, but rather to treat them additively and independently from one another. This last point effectively treated overlapping FrameNet annotations as new and separate ones. Regarding $\mathcal{X}$, this collapsed and aggregated four of the components (FrameNet frames and roles for two systems) into two (cumulative FrameNet frame and role counts). Therefore although $\mathcal{X}$ started as a 9-tensor, I only consider up to 6-tensors: trigger, role filler, token separation, PropBank frame and role, (aggregate) FrameNet

	windowed	**frame**
# target words	232	35.9 (triggers)
# surrounding words	232	531 (role fillers)

(a) *New York Times*

	windowed	**frame**
# target words	404	45.7 (triggers)
# surrounding words	404	2,305 (role fillers)

(b) Wikipedia

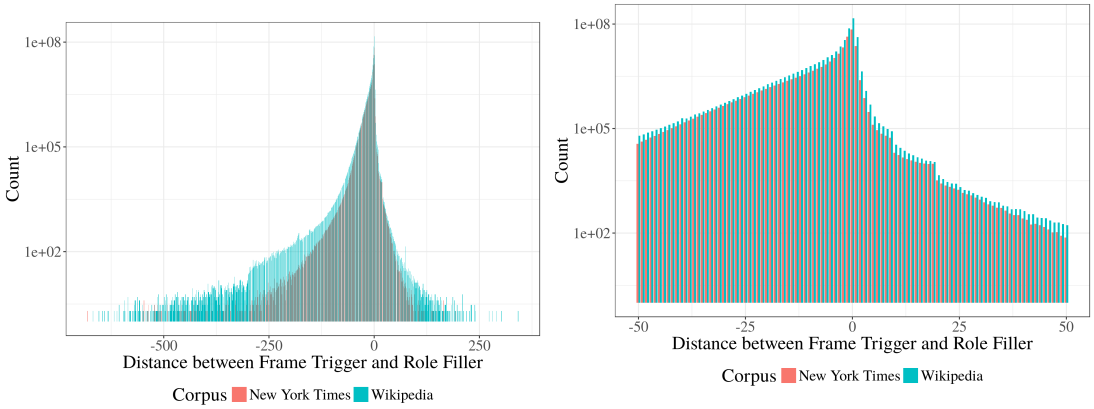
Table 5.1: Vocabulary sizes, in thousands, extracted from Ferraro et al. (2014)'s data with both the standard sliding context window approach (§5.1) and the frame-based approach (§5.3). Upper numbers (Roman) are for newswire; lower numbers (italics) are Wikipedia. For both corpora, 800 total FrameNet frame types and 5100 PropBank frame types are extracted.

frame, and (aggregate) FrameNet role.

5.3.3 Data Discussion

The baseline extraction methods result in roughly symmetric target and surrounding word counts. This is not the case for the frame extraction. The target words must trigger some semantic parse, so the target words are actually target triggers. However, the surrounding context words are those words that fill semantic roles. As shown in Table 5.1, there are an order-of-magnitude fewer triggers than target words, but up to an order-of-magnitude *more* surrounding words.

In Figure 5.2a, I show the log-count histogram of the number of words separating a role's filler from its corresponding trigger. Because the data are automatically



(a) All sentences.

(b) Only sentences with at most 100 words are shown.

Figure 5.2: A histogram of the number of words separating role fillers from their frame triggers. Counts are on a log scale.

processed, rather than being curated, there can be sentence segmentation errors. Figure 5.2b shows those sentences that are at most 100 words long. I found that most correctly segmented sentences are under this length. Notice across both corpora the ability to directly access a long history as well as the similar separation distributions.

5.3.4 Evaluating Semantic Content with SPR

Motivated by Dowty (1991)'s proto-role theory, Reisinger et al. (2015), with a subsequent expansion by White et al. (2016), annotated thousands of predicate-argument pairs (v, a) with (boolean) applicability and (ordinal) likelihoods of well-motivated semantic properties applying to/being true of a.[5] These likelihood judgments, under

[5]This sections uses the training portion of `http://decomp.net/wp-content/uploads/2015/08/UniversalDecompositionalSemantics.tar.gz`.

awareness	change_of_location	change_of_possession	change_of_state
changes_possession	existed_after	existed_before	existed_during
instigation	location_of_event	makes_physical_contact	partitive
sentient	stationary	volition	was_for_benefit

Table 5.2: Available semantic proto-role properties.

the SPR framework, are converted from a five-point Likert scale to a 1–5 interval scale. All of the SPR annotations considered here—Reisinger et al. (2015)'s and White et al. (2016)'s—can be directly linked to gold standard syntactic analyses.

Per SPR predicate v, I define each oracle vector $\mathbf{o}_v$ over all observed joint properties p and syntactic labels s. Each component of an oracle vector $\mathbf{o}_{v,(p,s)}$ is the unity-normalized sum of likelihood judgments over those joint property and syntactic relation responses. That is, given a sentence x with an SPR-labeled predicate v with interval response $y_{x,v,p}$ and (Boolean) applicability response $a_{x,v,p}$, I compute each component as

$$\mathbf{o}_{v,(p,s)} \propto \sum_{\text{SPR sentence } x : v \in x} \begin{cases} y_{x,v,p} & p \text{ is applicable } (a_{x,v,p} \text{ is true}) \\ 0 & p \text{ is not applicable } (a_{x,v,p} \text{ is false}). \end{cases}$$

Recall from §3.2.3 that the applicability responses represent cases when it does not even make sense to ask if a particular property likely holds with respect to a given predicate and object; for example, in example 3.10 (repeated below)

(5.3) Chris ate a pastry

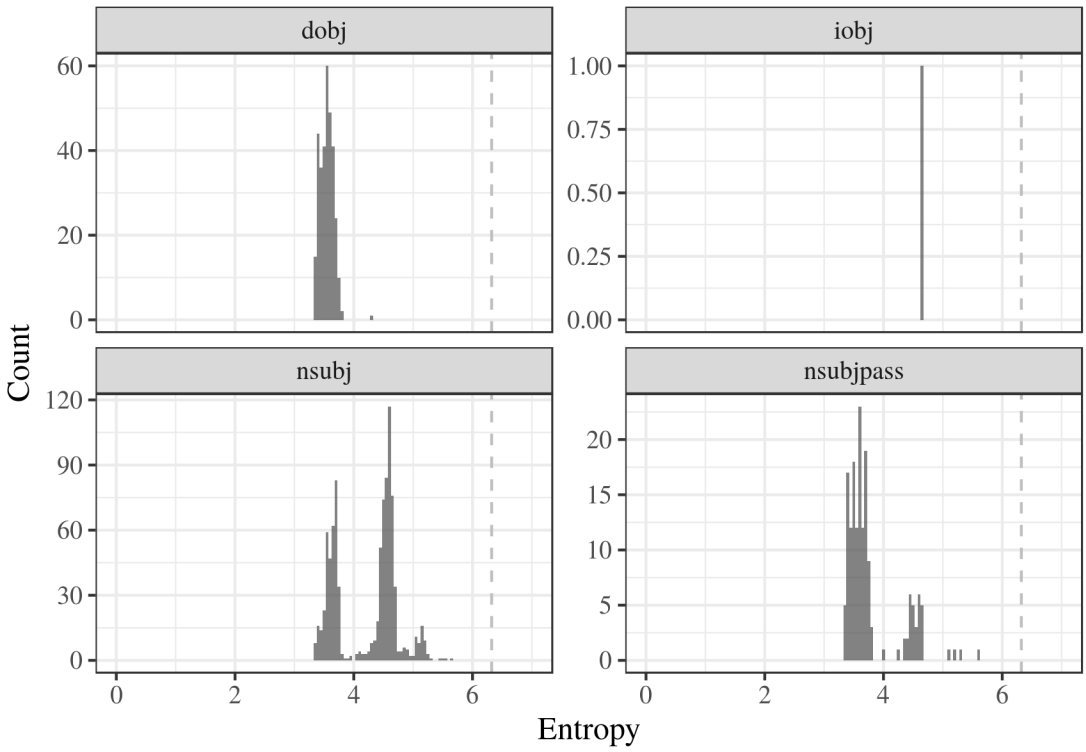


Figure 5.3: The entropy distribution of the oracle SPR-QVEC vectors, grouped according to most frequent syntactic relation.

it would be reasonable to assign an applicability score of "false" to the predicate-object-property combination of "ate-pastry-volition." Notice that in forming the oracle vectors I treat a false applicability response as a 0 response.

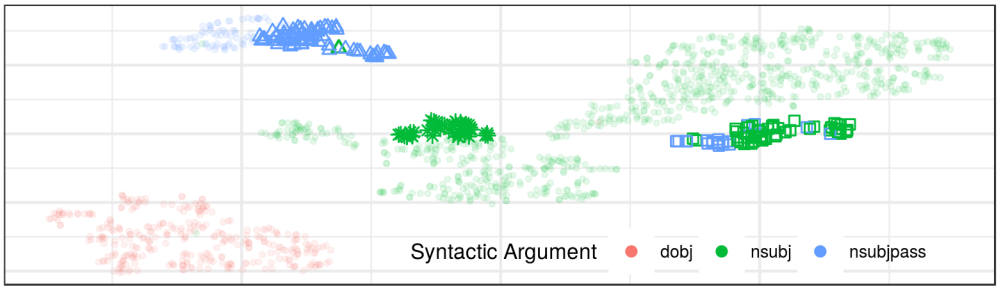
In Table 5.2, I show the combined 20 properties of Reisinger et al. (2015) and White et al. (2016). Together with the four basic grammatical relations *nsubj*, *dobj*, *iobj* and *nsubjpass*, these properties result in 80-dimensional oracle vectors.[6] In Fig-

[6]The full cooccurrence among the properties and relations is relatively sparse. Nearly two thirds of all non-zero oracle components are comprised of just fourteen properties, and only the *nsubj* and *dobj* relations.

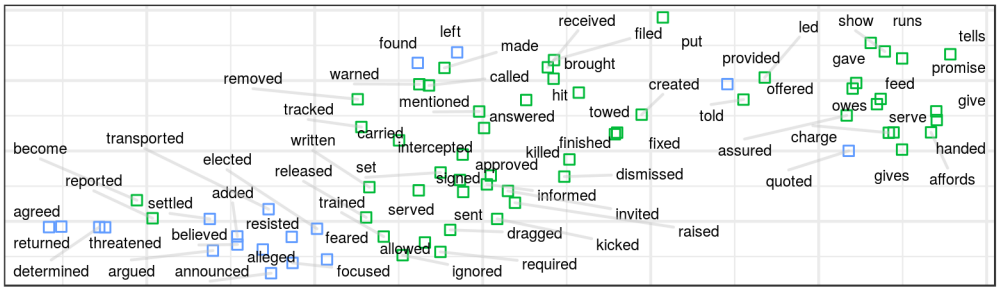
ure 5.3 I show the (empirical) entropies of the oracle vectors $\mathbf{o}_v$, grouped according to v's most frequent syntactic relation s; note that this does not reflect the *overall* syntactic usage distribution—just the most frequent. The dashed line represents a uniform distribution's entropy. Notice that while object preferring verbs (*dobj*, and *iobj*) result in unimodal entropy distributions, subject preferring verbs (*nsubj*, and *nsubjpass*) result in bimodal distributions.

I provide a qualitative t-SNE (van der Maaten and Hinton, 2008) analysis of these oracle SPR vectors in Figure 5.4.[7] As seen in the full plot (Figure 5.4a), there are noticeable syntactic clusterings. Those predicates most frequently observed with active subjects are in green, those observed most often observed with passive subjects are in blue, and those most often observed with direct objects are in red (given the sparsity of *iobj*-preferring predicates, *iobj* is removed from Figure 5.4). I highlight, and in Figure 5.4b through Figure 5.4d zoom in on, three areas within the main figure with squares, triangles and stars, respectively. Figure 5.4b (squares) has a solid representation of reporting and ditransitive predicates, demonstrating a mix of active (green) and passive (blue) subject predicates: notice both the active, demonstrative clustering (e.g., "offered," "owes," and "gave") as well as passive reporting predicates (e.g., "believed," "announced," "added," and "alleged"). In Figure 5.4c (triangles) I present passive subject predicates; note the prevalence of certain violence-laden predicates (e.g., "assassinated," "mauled," "assaulted," and "strangled") all encoding low

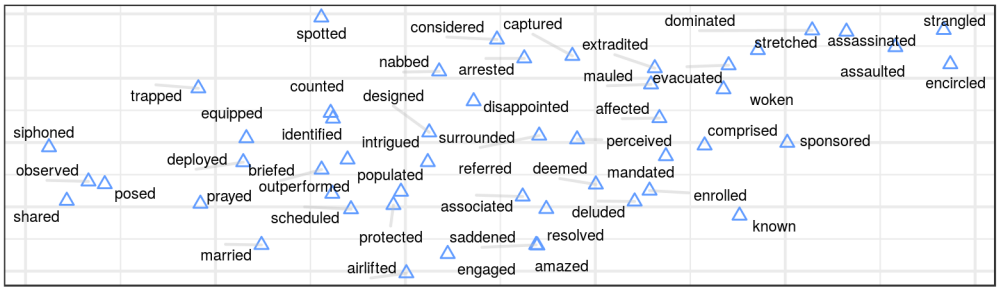
[7] t-SNE is an effective method for visualizing high-dimensional data in two dimensions while maintaining the high-dimensional structure.



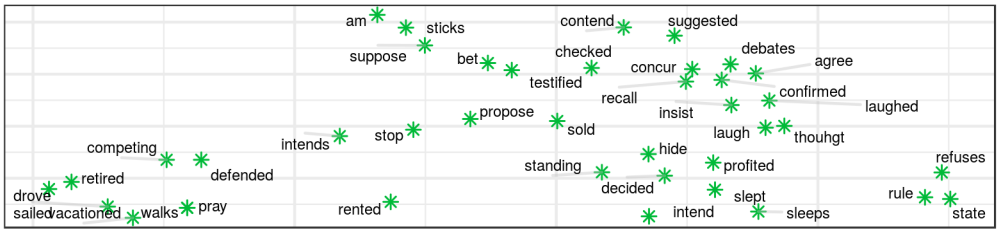
(a) The full T-SNE plot of the oracle vectors. Three zoomed portions have been provided, in 5.4b to 5.4d, as given by squares, triangles and stars, respectively.



(b) Predicates associated with active and passive subjects (the squares from Figure 5.4a).



(c) Predicates associated with passive subjects (the triangles from Figure 5.4a).



(d) Predicates associated with active subjects (the stars from Figure 5.4a).

Figure 5.4: A T-SNE representation of the oracle SPR-QVEC vectors. Each point is an SPR predicate (type). The color of each point indicates the most common syntactic argument; for clarity, *iobj* has been removed given its sparsity.

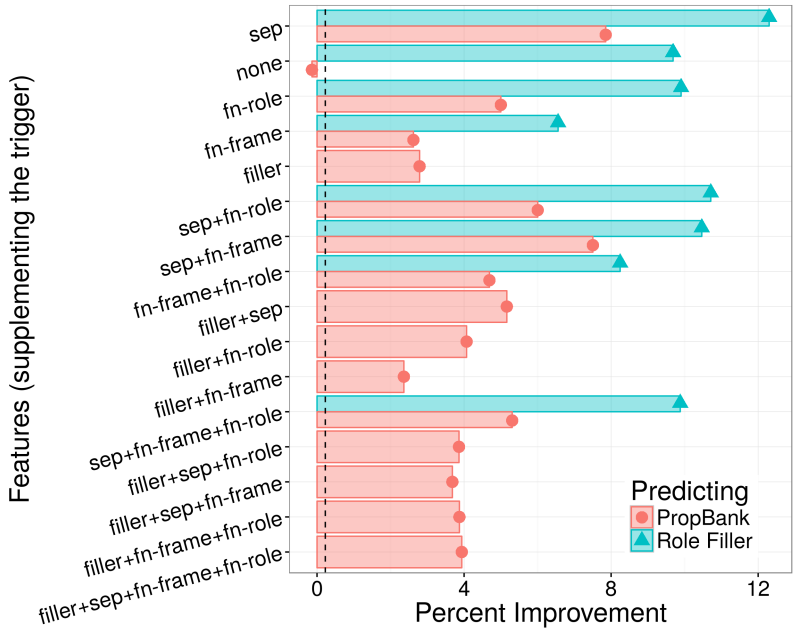
volition but high change of state and sentience of the subject. Finally, in Figure 5.4d I present active subject predicates, such as "sold," "defended" and "confirmed"; these tend to suggest the subject is both volitional and sentient, but does not necessarily change state. For example, in (5.4), Chris is very likely to experience a change of state (going from a state of "not hurt" to "hurt"), but is very unlikely to *want* to participate in this event; meanwhile, in (5.5), Chris is likely both sentient and volitionally participating in the selling act, but Chris does not necessarily change state.

(5.4) *Chris* was *mauled*.

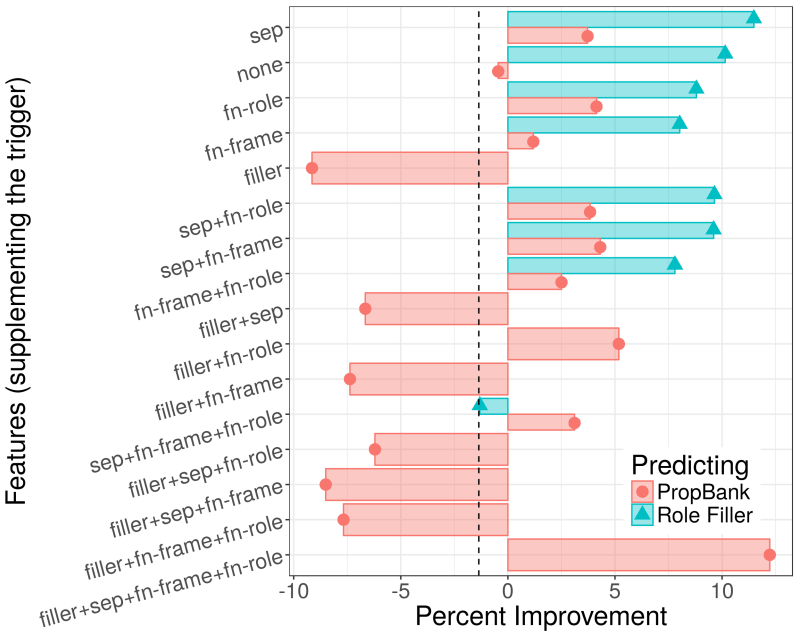
(5.5) *Chris sold* stock.

5.3.5 Results

Figure 5.5 shows the overall percent change for SPR-QVEC from the filler and role prediction models, on newswire (Figure 5.5a) and Wikipedia (Figure 5.5b), across different ablation models. All learned embeddings are 100 dimension vectors; the dimension was not optimized nor was it chosen to be close to the dimensionality of the SPR-QVEC oracle vectors. I indicate additional contextual features being used with a `+`: `sep` uses the token separation distance between the frame and role filler, `fn-frame` uses FrameNet frames, `fn-role` uses FrameNet roles, `filler` uses the tokens filling the frame role, and `none` indicates no additional information is used when predicting. The 0 line represents a plain `word2vec` baseline and the dashed line



(a) Changes in SPR-QVEC for *Annotated NYT*.



(b) Changes in SPR-QVEC for Wikipedia.

Figure 5.5: Effect of frame-extracted tensor counts on SPR-QVEC. Deltas are shown as relative percent changes vs. the `word2vec` baseline. Each row represents an ablation model: `sep` uses the token separation distance between the trigger and filler, `fn-frame` (`fn-role`) uses FrameNet frames (roles), and `filler` uses the tokens filling the frame role. Only PropBank is predicted when `filler` is used.

represents the 3-tensor baseline of Cotterell et al. (2017). Both of these baselines are windowed: they are restricted to a local context and cannot take advantage of frames or any lexical signal that can be derived from frames.

Overall, we notice that we obtain large improvements from models trained on lexical signals that have been *derived* from frame output (`sep` and `none`), even if the model *itself* does not incorporate any frame labels. The embeddings that predict the role filling lexical items (the green triangles) correlate higher with SPR oracles than the embeddings that predict PropBank frames and roles (red circles). Examining Figure 5.5a, we see that both model types outperform both the `word2vec` and Cotterell et al. (2017) baselines in nearly all model configurations and ablations. We see the highest improvement when predicting role fillers given the frame trigger and the number of tokens separating the two (the green triangles in the `sep` rows).

Comparing Figure 5.5a to Figure 5.5b, we see newswire is more amenable to predicting PropBank frames and roles. I posit this is a type of out-of-domain error, as the PropBank parser was trained on newswire. We also find that newswire is overall more amenable to incorporating limited frame-based features, particularly when predicting PropBank using lexical role fillers as part of the contextual features. This is likely due to the significantly increased vocabulary size of the Wikipedia role fillers (c.f., Table 5.1). Note, however, that using all available schema information when predicting PropBank can compensate for the increased vocabulary.

anticipated

Filler | sep

1 foresaw	6 pondered
2 figuring	7 kidded
3 alleviated	8 constituted
4 craved	9 uttering
5 jeopardized	10 forgiven

anticipated

PropBank | sep

1 anticipate	6 intimidated
2 anticipating	7 separating
3 anticipates	8 separates
4 stabbing	9 drag
5 separate	10 guarantee

invented

Filler | sep

1 pioneered	6 tolerated
2 scratch	7 resurrected
3 complemented	8 sweated
4 competed	9 fancies
5 consoled	10 concocted

invented

PropBank | sep

1 invent	6 aspire
2 document	7 documenting
3 documented	8 aspires
4 invents	9 inventing
5 documents	10 swinging

producing

Filler | sep

1 containing	6 storing
2 contains	7 reproduce
3 manufactures	8 store
4 contain	9 exhibiting
5 consume	10 furnish

producing

PropBank | sep

1 produces	6 ridden
2 produce	7 improves
3 produced	8 surround
4 prized	9 surrounds
5 originates	10 originating

Figure 5.6: K-nearest neighbors for three randomly sampled trigger words, from two newswire models.

Learning Similar Triggers

In Figure 5.6 I display the ten nearest neighbors for three randomly sampled trigger words according to two of the highest performing newswire models. They each condition on the trigger and the role filler/trigger separation; these correspond to the `sep` rows of Figure 5.5a. The left column of Figure 5.6 predicts the role filler, while the right column predicts PropBank annotations. We see that while both models learn inflectional relations, this quality is prominent in the model that predicts PropBank information while the model predicting role fillers learns more non-inflectional paraphrases.

Off-the-Shelf Vectors

There are a number of word embeddings that are freely available for download: in a sense, they have become somewhat of a commodity. For completeness, I examined three sets of these:

(1) Google News `word2vec` embeddings (Mikolov et al., 2013a),[8]

(2) GloVe vectors (Pennington et al., 2014),[9] and

(3) multiview LSA vectors (Rastogi et al., 2015, MVLSA).[10]

The Google News embeddings are 300 dimension vectors trained on roughly 100 billion words; the GloVe embeddings consist of a variety of dimensions, ranging from 50 to 300, trained on roughly 6 billion words from a 2014 Wikipedia release and English Gigaword v5 (Parker et al., 2011); the MVLSA embeddings are 300 dimension vectors trained from a preprocessed 2013 Wikipedia release (Al-Rfou et al., 2013), augmented with parallel bitext, morphological, syntactic and a paraphrastic FrameNet expansion (Rastogi and Van Durme, 2014). These *released* vectors are all optimized, in some fashion, toward the similarity tasks discussed in §5.2.

Although each of the three sets of off-the-shelf vectors achieve higher SPR-QVEC performance than the frame-based vectors I trained for Figure 5.5, it is inappropriate to compare them. First, this comparison is of vectors with different dimensionalities.

[8] `https://code.google.com/archive/p/word2vec/`

[9] `http://nlp.stanford.edu/data/glove.6B.zip`

[10] `https://zenodo.org/record/16710/files/combined_embedding_0.emb.ascii.gz` and `https://zenodo.org/record/16710/files/combined_embedding_0.word.ascii.gz`

Off-the-Shelf Baselines				Retrained Baseline	This chapter's models			
					PropBank predictor		Role Filler predictor	
Size	GoogleNews	MVLSA	GloVe	*ANYT*-word2vec	filler	filler+sep +fn-frame+fn-role	—	sep
50	—	—	0.246	0.261	0.255	0.253	0.267	**0.279**
100	—	—	0.255	0.260	0.270	0.270	0.281	**0.285**
200	—	—	0.368	0.376	0.415	**0.417**	0.415	**0.417**
300	0.459	0.462	0.457	0.460	0.520	**0.528**	0.514	0.513

Table 5.3: Comparison of off-the-shelf vectors and select frame-based models of varying vector dimensionality. Each number is the SPR-QVEC score; higher is better. The best performing models, per dimension of learned embeddings, is **bolded**. Comparing against GloVe demonstrates the improved capability of capturing SPR expectations across different embedding dimensions. Comparing against all three off-the-shelf methods, which use different kinds and amounts of auxiliary information, demonstrates the ability of the frame-based tensor factorization presented in this chapter to capture SPR expectations.

I argue that optimizing the metric by changing the number of free parameters is orthogonal to the goal of studying how to capture SPR qualities *through* additional semantic information in as controlled a way as possible.

Second, this section examined a controlled evaluation, which in part involves controlling for the raw training data. This includes the exact data set and documents used, as well as any auxiliary information, such as the complementary views within MVLSA. Such a controlled evaluation is not possible to do with off-the-shelf vectors.

Third, as Rastogi and Van Durme (2014) demonstrate, the hyperparameter values, including subsampling, can be crucial to achieving state-of-the-art performance on a particular task. Such performance was not the goal here.

However, because the released GloVe vectors contain vectors of sizes 50, 100, 200, and 300, we can examine how changing dimensionality can impact the score. Note

that the GloVe training set—a combined corpus of Wikipedia and newswire articles—is moderately close to the training data I used. Looking specifically at the newswire embeddings of this section, the lower dimensional GloVe vectors (50 and 100) both perform worse than any of the four `word2vec` or Cotterell et al. (2017) baselines in Figure 5.5 (thereby being outperformed by a majority of the frame-based newswire models); on the other hand, the higher dimensional vectors (200 and 300) outperform all of the 100 dimensional vectors learned in this section.

The above results can be seen in Table 5.3, which also compares the 300 dimensional Google News and MVLSA embeddings against multiple kinds of vectors learned in this section. Specifically, Table 5.3 compares these off-the-shelf baselines against four different frame-based newswire models: two predicting PropBank and two predicting role fillers. All have access to FrameNet frames and roles; the other contextual features use token separation information and, for PropBank predicting models, lexical role fillers. Overall the selected models represent "middle of the pack" models. Comparing these models to GloVe and a plain `word2vec`-style baseline (the 0 line of Figure 5.5), we first observe that higher dimension vectors almost always produce high absolute SPR-QVEC scores (there is some minor jumbling between 50 and 100). Second, we notice that the *relative* performance gains increase as the dimensionalities increase: while the PropBank predicting models under-perform the `word2vec`-style baseline at 50 dimensions by 3%, at 100 dimensions there is a six point flip, where the PropBank models outperform the baseline by 3%; at 200, the models outperform

the baseline by roughly 11%, and at 300 by roughly 15%. Third, the GloVe vectors underperform all selected models and baselines at all dimensions. Fourth, the other two off-the-shelf vectors perform roughly on par with GloVe, underperforming the frame-based models once dimension is controlled for.

Syntactic Content Evaluation

Even though in this section I am examining word embeddings that capture SPR, it is nevertheless interesting to examine if those vectors that yield improvements translate into other aspects of language. To look at this, I consider one of the oracle sets originally introduced with QVEC: part-of-speech tags (Tsvetkov et al., 2015). In this setting, which I call POS-QVEC, each word's oracle vector is defined over 45 standard Penn Treebank part-of-speech tags (Santorini, 1990, including nine punctation tags).

Because the frame-based models incorporate many long-range dependencies, it is not at all obvious that the frame-based models, as formulated, should outperform either plain `word2vec` or Cotterell et al. (2017) baselines when evaluating parts-of-speech. And overall, many of the frame-based models, trained on either newswire or Wikipedia, do not result in POS-QVEC improvements, but rather actively degrade performance. Unlike with SPR-QVEC, the newswire baselines proved more difficult than the Wikipedia baselines to improve upon.

I found the models that predicted PropBank information to be especially underperforming, while models that predicted the (lexical) role fillers had mixed per-

formance: a number below the baselines, but some above. However, there are two consistent instances where frame-based information *in role filler predicting models* does improve upon both baselines: when recording the number of words between the trigger and role filler (`sep`) and when using FrameNet role information (`fn-role`). These models surpassed one, and often both, baselines. This holds for both newswire and Wikipedia, though the gains in newswire are smaller.

These results should not be too surprising. First, Cotterell et al. (2017) also demonstrated that including token separation information can provide an additional, robust signal for lower level syntactic modeling. Second, given the relatively tight syntax-semantics interface in English, and particularly the successful proxy uses in NLP of syntax for semantics (De Marneffe and Manning, 2008; Rudinger and Van Durme, 2014), it should not be surprising that FrameNet roles can provide signal that benefits syntactic measures—particularly when syntactic forms and patterns directly inform the labeling of roles, as is the case with CAC. Third, it is not surprising that vectors trained, using negative sampling, to predict the correct role filler from a provided trigger outperforms vectors trained to predict PropBank information on any sort of lexically-diverse measure like POS-QVEC.

5.3.6 Related Work

The recent popularity of word embeddings have inspired others to consider leveraging linguistic annotations and resources to learn embeddings. Both Cotterell et al.

(2017) and Levy and Goldberg (2014a) incorporate additional syntactic and morphological information in their word embeddings; this additional information, like the frame-based information used in this chapter, is obtained at the token (rather than type) level.

There has been a variety of work on incorporating type-level information too. One of the off-the-shelf methods from before, Rastogi et al. (2015)'s multiview LSA (MVLSA) approach studies augmenting the learning process with summary statistics from paraphrased FrameNet training data. Yu and Dredze (2014) and Rothe and Schütze (2015) use lexical resource entries, such as WordNet synsets or paraphrases (Ganitkevitch et al., 2013), to improve pre-computed word embeddings. Faruqui et al. (2015) present a belief propagation algorithm to realign, or "retrofit," existing word embeddings using relational information from semantic lexicons. On the applied side, Wang and Yang (2015) used frame embeddings—produced by training `word2vec` on tweet-derived semantic frame (names)—as additional features in downstream prediction. Mousselly-Sergieh and Gurevych (2016) examined the problem of schema alignment: they combined the FrameNet type level information and exemplar data with pre-existing word vectors in order to align FrameNet with WikiData (a user contributed knowledge base).

Other efforts have merged frame semantics with notions of continuous or dimensionality reduced representations of words and documents yield both intrinsic and downstream improvements in NLP systems. Chen et al. (2014) incorporate frame

semantics to improve dialogue systems, Peng and Roth (2016)'s semantic language models leverage semantic frames, and Ferraro and Van Durme (2016), which will be covered in chapter 7, demonstrate how semantic frames help improve script induction.

5.4 Reflecting Human Biases

From the cognitive science perspective, two theories of learning word ("category") meaning—prototype- and exemplar-based—are centered around *featurized* representations (Minda and Smith, 2002).[11] Though the precise methodology differs from study to study, these features are generally elicited and deemed to be "important" or distinguishing in some manner (McRae et al., 2005). As McRae et al. (2005) argue, the featurized representations should be viewed as intermediate and interpretable representations that are useful during elicitation or priming experiments (McRae et al., 1997a; Hare et al., 2009); the use of a particular representation should not by default be taken to be *the* "true" representation of that category. For example, particularly salient features of a "duck" may be that it `lays eggs` and `swims`, while "knives" are `dangerous` but also `found in kitchens`. Lists of (possibly weighted) features for concepts form feature norms, which can be thought of as empirically-based representations of human biases.

[11] Prototype categorization hypothesizes that word meaning is based on how well a novel item's feature representation fits an ideal. Exemplar categorization, like nearest neighbors, hypothesizes that category meaning is based on how well a feature representation matches other representations from that category. Though I used SPR annotations, I do not advocate for one theory over the other (for a discussion of this, see McRae et al., 1997b, Section 2.1, "Computed Prototypes").

$	action	air	anger	animal
bad	body	break	breath	communicate
cook	down	emotion	express	eye
fast	feel	fire	food	foot
force	get	give	go	hand
hit	hot	humans	hurt	intentional
involuntary	leg	light	liquid	loud
make	mouth	move	noise	nose
object	sense	something	sound	tool
up	voice	walk	water	word

(a) Top 50 Vinson and Vigliocco (2008) event norm properties.

a_bird	a_fruit	a_mammal	a_vegetable	a_weapon
an_animal	beh_-_eats	beh_-_flies	beh_-_lays_eggs	beh_-_swims
clothing	different_colours	found_in_kitchens	has_4_legs	has_a_beak
has_a_handle	has_a_tail	has_feathers	has_fur	has_legs
has_seeds	has_wheels	has_wings	hunted_by_people	is_black
is_brown	is_dangerous	is_edible	is_electrical	is_expensive
is_fast	is_green	is_hard	is_heavy	is_large
is_long	is_loud	is_red	is_round	is_small
is_soft	is_white	is_yellow	lives_in_water	made_of_metal
made_of_plastic	made_of_wood	tastes_good	tastes_sweet	used_for_transportation

(b) Top 50 McRae et al. (2005) feature norm properties. The "beh" prefix indicates a behavior.

Table 5.4: Top 50 most common event (5.4a) and concept (5.4b) feature norm properties.

In the previous section, while not appealing to one theory or another, I examined embeddings that better capture semantic protorole properties, which featurize an action and its participants. In this section, I examine the extent to which these same frame annotations and derived embeddings can reflect feature norms, i.e., controlled, empirically-derived human biases.

5.4.1 Experimental Setup

I use the exact same methods, data, and learned vectors from §5.3. I experiment with two different sets of norms: one from Vinson and Vigliocco (2008) involving verb-based events (like "to punch" and "to whisper"), and the other from McRae et al. (2005) involving basic nominal concepts (both living, like "dog," and not, like "chair").

Though I go into details of each norm set below, the basic methodology is the same. The creators first identify concepts, generally represented as a single word, to explore. They generally take care to limit confusion regarding word senses and various semantic phenomena, like holonymy (unless that is what is being studied). Study participants (typically undergraduates) are then asked what the most important, distinguishing or salient feature (property) of those concepts are, often through an ellicitation process. The final feature set is obtained by study creator post-processing, such as removing exceedingly rare features, or reconciling different feature names.

5.4.1.1 Vinson Event Norms

Vinson and Vigliocco (2008) produced and released, as part of a larger feature norm dataset, norms for 216 event-carrying verbs. 280 annotators each annotated between thirty and forty event verbs; the initial feature labels were post-processed and manually verified, with features with fewer than nine annotators and verifiers removed. In total, these verbs have 895 features. I show the top 50 most common

features in Table 5.4a. Although I do not use them, each of the features is coded with a type of meta-feature—visual, perceptual, functional, motoric—describing how those features are experienced by people.[12] Because these norms are for event carrying verbs, which correspond with frame triggers, I use the learned trigger embeddings of §5.3.

5.4.1.2 McRae Nominal Norms

McRae et al. (2005) produced and released a set of feature norms for 541 concrete concepts, such as "cake," "knife," and "sink." The list of concepts was accumulated over decades of work by multiple researchers. Each of these concepts presents as a noun. Each concept was annotated by thirty annotators, and at least five must have included a feature for it to be included. Without any pruning, these concepts have 2,526 features; as with the event norm features, the initial feature labels here were post-processed down. I show the top 50 most common features in Table 5.4b. In Table 5.5 I show twelve randomly sampled concepts (bold rows) against the union of their marked features; I have included the IS_EDIBLE feature to demonstrate a feature that does not fire on any of the sampled concepts.

These norms are for concrete objects; rather than corresponding with frame triggers, they correspond with role fillers. Therefore, I use the learned, contextual role filler embeddings of §5.3. Note that these embeddings are in effect by-products of

[12]These meta-features could be useful for multimodal studies.

	A_WEAPON	IS_BROWN	IS_EDIBLE	IS_HEAVY	IS_LONG	MADE_OF_PLASTIC	MADE_OF_WOOD
bat				✓	✓		✓
board		✓			✓		✓
bow	✓				✓	✓	✓
broom					✓	✓	✓
crowbar	✓			✓	✓		
pipe					✓	✓	✓
rifle	✓			✓	✓		
ruler					✓	✓	✓
sledgehammer				✓	✓		✓
spatula					✓	✓	✓
spear	✓				✓		✓
stick	✓				✓		✓

Table 5.5: Examples of randomly sampled **concepts** such as nouns (rows) and PROPERTIES (columns) from the McRae et al. (2005) feature norms. While all of the sampled concepts are "long," none are "edible," some may be made out of plastic or wood (or both), and some may be used as a weapon.

the frame extraction and embedding process: the extraction described in §5.3.1 is centered around extracting the core semantic parse *type*.

5.4.2 Evaluating Feature Norms

Here I explore mapping both sets of feature norms into oracle vectors. Both sets have more than 1,000 types of features, though they both follow a general power law. Because QVEC is recall-focused, the same vectors can achieve higher scores with larger oracle vectors than with smaller ones (Tsvetkov et al., 2015). I therefore only construct oracle vectors from the 50 most common features per set.

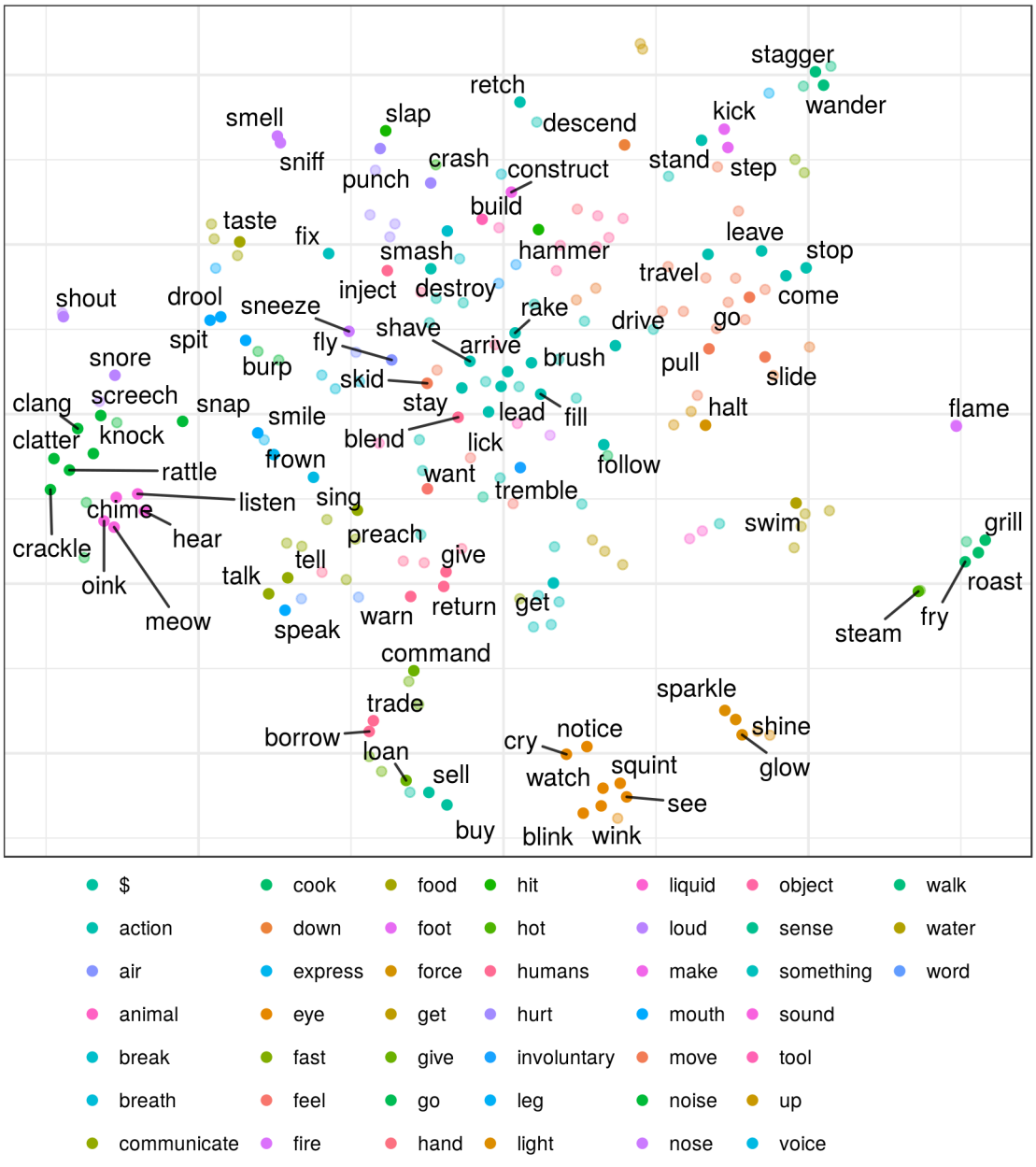


Figure 5.7: A T-SNE representation of the oracle VINSON-QVEC vectors.

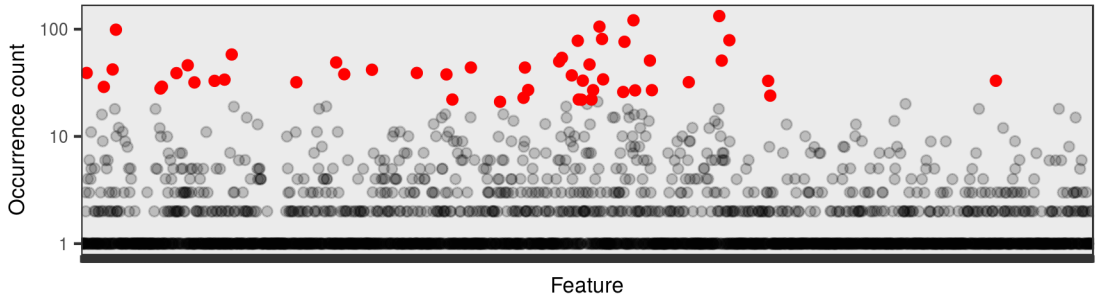
5.4.2.1 Vinson Event Norms

For each event type v annotated by Vinson and Vigliocco (2008), I define each oracle vector $\mathbf{o}_v$ over the 50 most common observed *event* features f. Each component of an oracle vector $\mathbf{o}_{v,f}$ is the unity-normalized number of association judgments over that event $\mathbf{o}_{c,f} \propto N_{v,f}$, where $N_{v,f}$ is the total number of times v was judged to have feature f. I show the top 50 most common features in Table 5.4a.

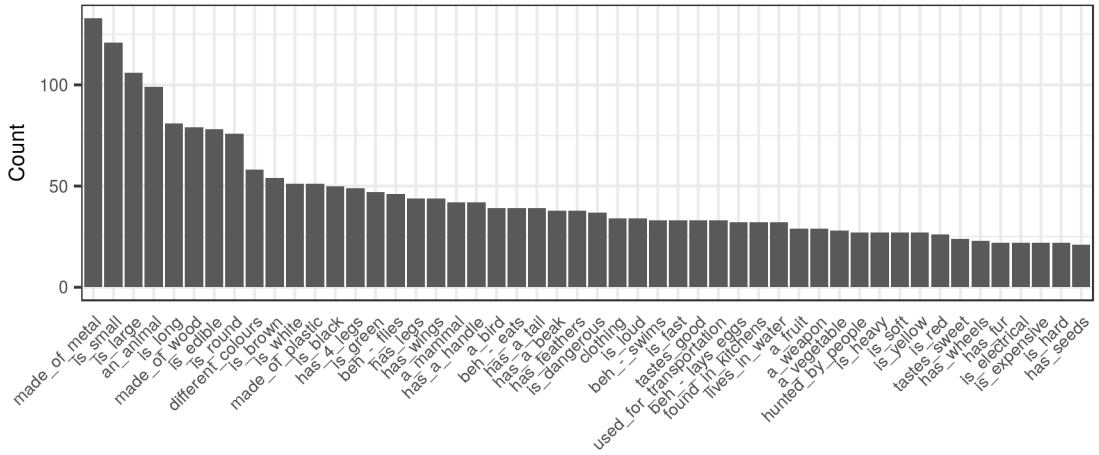
Figure 5.7 provides a visualization of the Vinson event oracles. The color of each point represents the highest weighted feature per concept; in a way, we can think of these highest weighted features as being representative aspects for that event. There are noticeable, intuitive groupings of events: water sports/events, like swimming, diving, and wading, are best represented by `water`; olfactory events "sniff" and "smell" are roughly coincident and characterized by their common means (`nose`); and verbs of observation—"blink," "cry," and "notice" (`eye`)—and luminescent events—"sparkle," "shine," and "glow" (`light`)—exhibit both intra- and inter-group clustering.

5.4.2.2 McRae Nominal Norms

Per concept c annotated by McRae et al. (2005), I define each oracle vector $\mathbf{o}_c$ over all observed features f. Each component of an oracle vector $\mathbf{o}_{c,f}$ is the unity-normalized number of association judgments over that concept. Specifically, if $N_{c,f}$



(a) A scatterplot histogram for all features. The top 50 are shown in red.



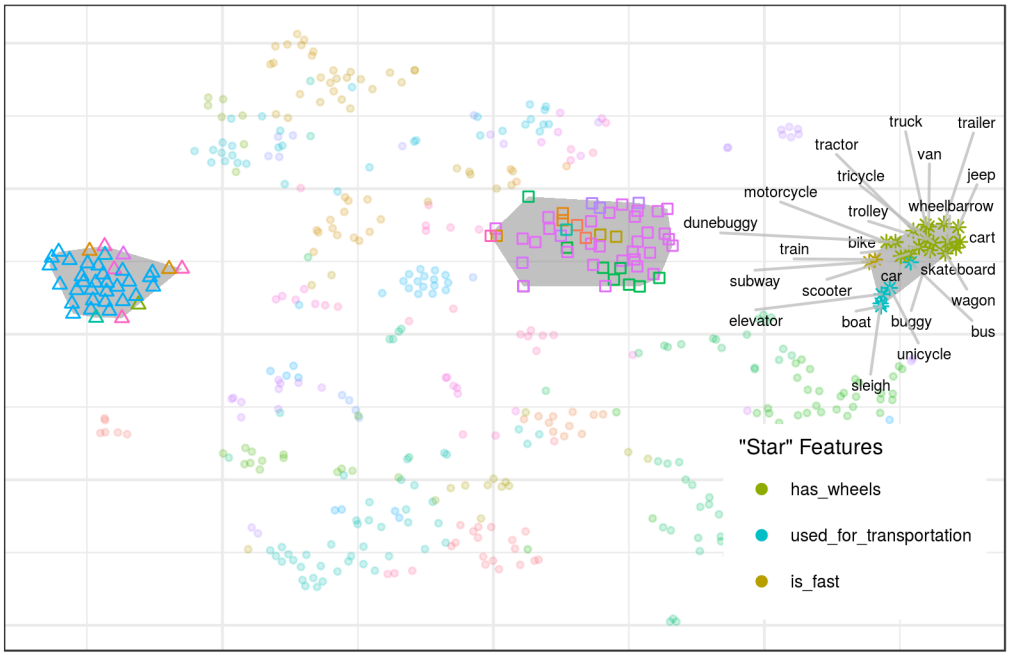
(b) A histogram of the top 50 features.

Figure 5.8: Scatterplot histograms of the oracle McRae-qvec vectors. In Figure 5.8a, all features are shown; in Figure 5.8b, only the top 50 features (those in Table 5.4b are shown.

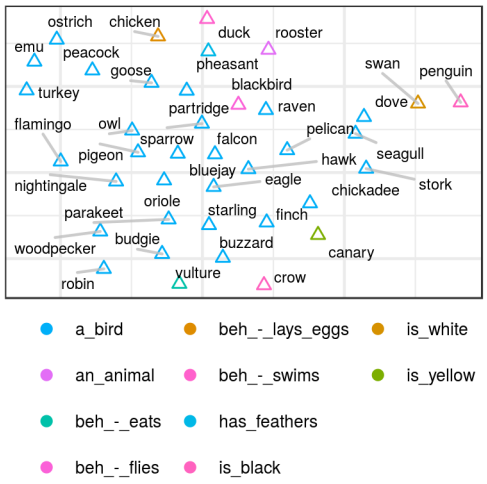
out of the total $M_{c,f}$ annotators judged features for concept c, then

$$\mathbf{o}_{c,f} \propto \frac{N_{c,f}}{M_{c,f}}.$$

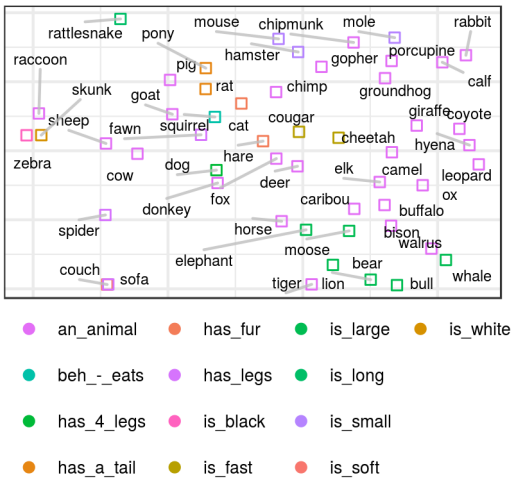
As Figure 5.8a shows, these 2,500 follow a typical power law. I show the top 50 most common features in Table 5.4b and Figure 5.8b.



(a) The full T-SNE plot of the oracle vectors. A small sample, given by the stars, shows a grouping of methods and means of transportation. For contrast two other areas have been provided in 5.9b and 5.9c.



(b) Bird distinctions (the triangles from Figure 5.9a).



(c) Some general defining animal characteristics (the squares from Figure 5.9a).

Figure 5.9: A T-SNE representation of the oracle MCRAE-QVEC vectors. Each point is a featurized argument. The color, consistent across the three plots, of each point indicates the most feature. "beh" indicates a behavior.

As when studying SPR in §5.3, I provide a qualitative t-SNE analysis of these oracle feature norm vectors in Figure 5.9. The color of each point represents the highest weighted feature per concept; in a way, we can think of these highest weighted features as being the most distinctive feature for that concept. Across all plots in Figure 5.9, the colors are consistent.[13] I examine three clusters: the first (transportation) serves as high-level contrast to the second and third (birds and other animals), while these latter two serve to constrast intuitive differences between those animals. The first, as given by the stars in Figure 5.9a, identifies certain methods of transportation which can all be most distinguished with three simple features: `has_wheels` (the effective default), `used_for_transportation`, and `is_fast`; this last feature best applies to "trains" and "subways," distinguishing them from other transportation methods, like "cars" and "buses." In constrast, consider the clusterings of birds (triangles, Figure 5.9b) and other animals (squares, Figure 5.9c). While the defaults simply label most as their category type—either as a bird or animal—we do notice some interesting differentiations. For instance, from Figure 5.9b we see that a vulture is the only bird most highly characterized by eating, while chickens lay eggs and canaries are yellow. On the other hand, we see in Figure 5.9c a clear clustering of "large" animals (bears, moose, elephants and whales), cats have fur, and squirrels—like vultures—eat (vultures and squirrels are the only concepts most identified with eating).

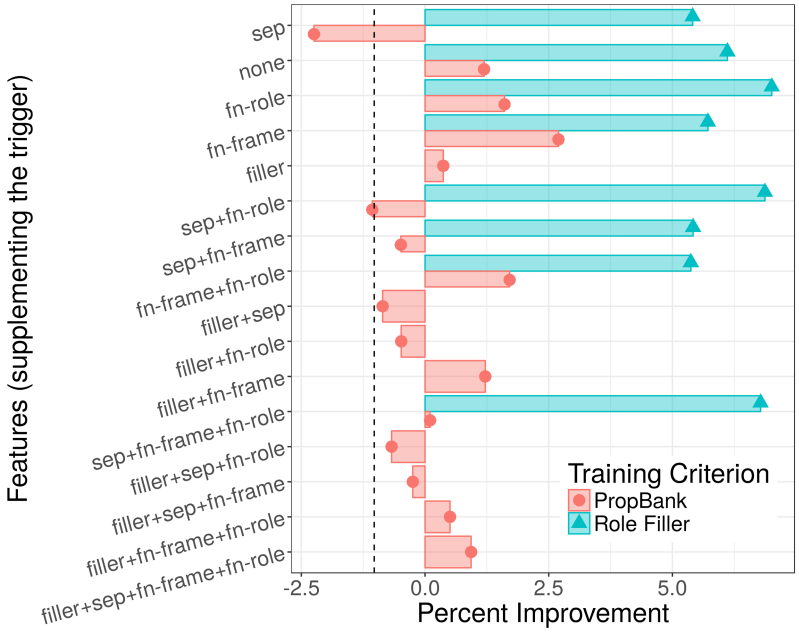
[13]For clarity, only a select few have these highest weighted features labeled. Note that in Figure 5.9a the legend only applies to the "star" points.

5.4.3 Results

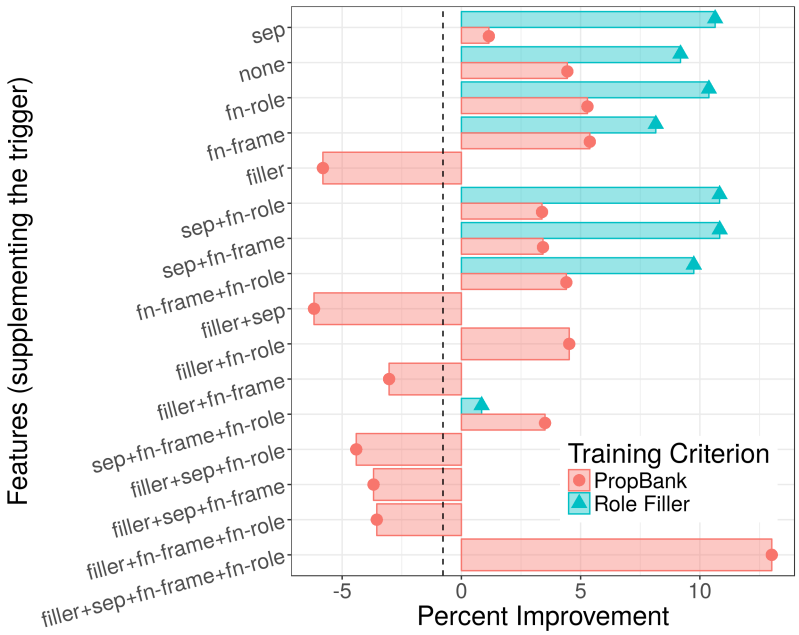
Here I present ablation results for both PropBank and role filler predicting models, on both VINSON-QVEC and MCRAE-QVEC. As in §5.3, I indicate additional contextual features being used with a `+`, the 0 line represents a plain `word2vec` baseline and the dashed line represents the 3-tensor baseline of Cotterell et al. (2017). Recall that both of these baselines are restricted to a local context; they do not use any information derived from frames, roles or associated lexical signals.

5.4.3.1 Vinson Event Norms

Figure 5.10 shows the overall percent change for VINSON-QVEC from the filler and role prediction models across different ablation models. In general, notice the pattern of improvement is similar to that of SPR-QVEC: frame and frame-derived information, such as trigger-filler token distance, improve upon both baselines. While the improvement on Wikipedia is roughly of the same magnitude as we observed with SPR, there is less improvement on newswire. As before, notice that the greatest improvements come when the vectors are trained on models that predict the lexical role fillers (green triangles), rather than predicting PropBank information (red circles). Like the syntactic evaluation from before, including FrameNet roles is helpful. Unlike the prior SPR or syntactic evaluations though, including the trigger-filler separations is generally harmful on newswire but helpful on Wikipedia.



(a) Changes in VINSON-QVEC for *Annotated NYT*.



(b) Changes in VINSON-QVEC for Wikipedia.

Figure 5.10: Effect of frame-extracted tensor counts on VINSON-QVEC. Deltas are shown as relative percent changes vs. the `word2vec` baseline. Each row represents an ablation model: `sep` uses the token separation distance between the trigger and filler, `fn-frame` (`fn-role`) uses FrameNet frames (roles), and `filler` uses the tokens filling the frame role. Only PropBank is predicted when `filler` is used.

5.4.3.2 McRae Feature Norms

Figure 5.11 shows the overall percent change for McRae-qvec from the PropBank and role filler prediction models across different ablation models. I reiterate that the vectors used in this section are effectively *by product* vectors: they are the context vectors learned from *frame*-oriented counts (rather than *role*-oriented counts).

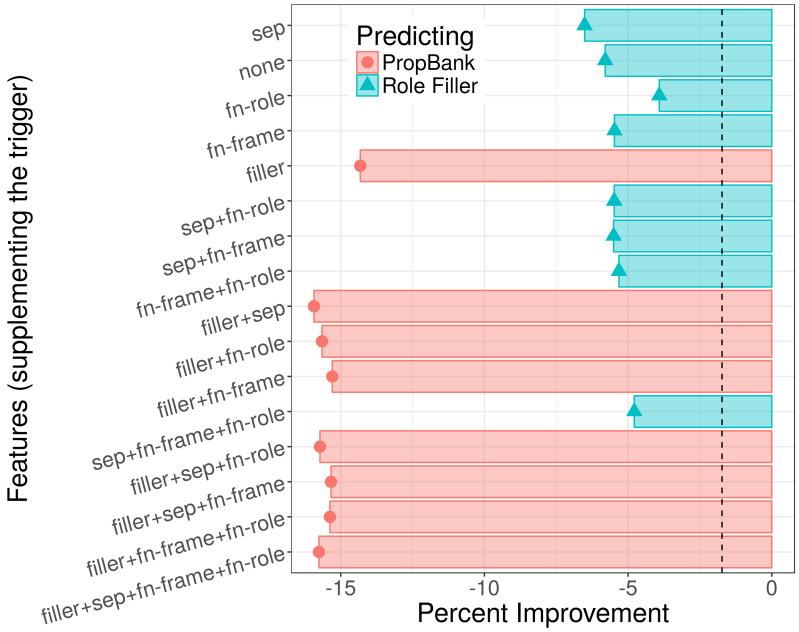
While Wikipedia-based vectors are able to outperform both baselines by between five and ten percent, the newswire vectors unfortunately fall short. Notice that the greatest improvements—or in the newswire case, the least harm—comes when the vectors are trained on models that predict the lexical role fillers (green triangles), rather than predicting PropBank information (red circles). This aligns with the earlier SPR-qvec results. We see the largest relative improvement (or least harm) when predicting role fillers given the frame trigger and FrameNet roles (the green triangles in the `fnrole` rows).

However, in contrast to the SPR, notice

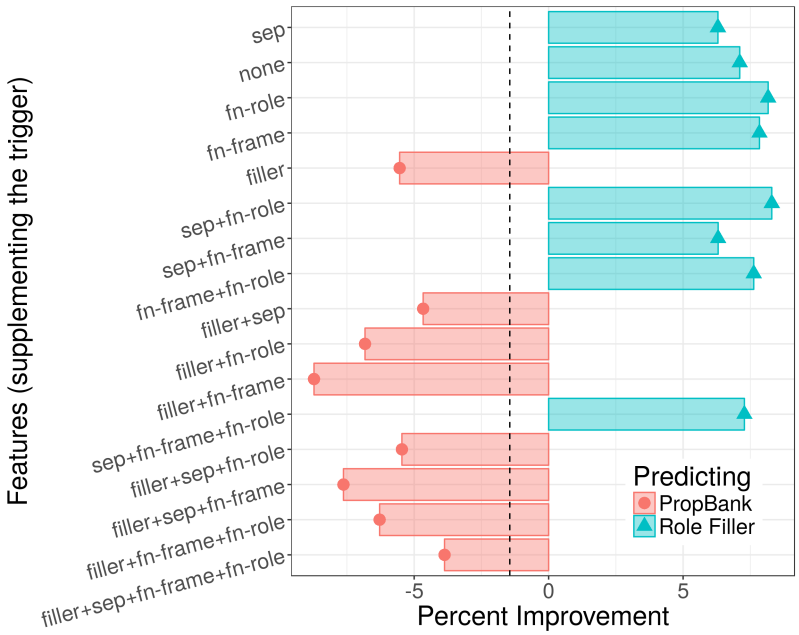
1. the overall changes are relatively uniform—given a type of prediction model, there is not one individual piece of information that is a panacea; and

2. including the trigger-filler separation is supplementary, rather than complementary as with SPR, to frame and role labels.

Moreover, there is still signal in including the frame-based trigger-filler separations, vs. the flat windowed separation information of (Cotterell et al., 2017): within Wikipedia



(a) Changes in McRae-qvec for *Annotated NYT*.



(b) Changes in McRae-qvec for Wikipedia.

Figure 5.11: Effect of frame-extracted tensor counts on McRae-qvec. Deltas are shown as relative percent changes vs. the `word2vec` baseline. Each row represents an ablation model: `sep` uses the token separation distance between the trigger and filler, `fn-frame` (`fn-role`) uses FrameNet frames (roles), and `filler` uses the tokens filling the frame role. Only PropBank is predicted when `filler` is used.

models, all filler-predicting models that include frame-based separation (`sep`) outperform the windowed token separation baseline.

5.4.4 Related Work

It is well known that humans are sensitive to priming influences; a number of efforts have shown this holds for event expectations and situation schema recall (Hare et al., 2009; Khalkhali et al., 2012, i.a.). That is, event primes provide a very strong signal of events and participants that "should" occur together. For instance, McRae et al. (1997b) examine Dowty-inspired verb-specific properties, much in the vein of SPR; they find that certain events lend themselves to verb-specific, property (feature)-structured roles. Hare et al. (2009) demonstrate that, even controlling for word association, nominals, like "sale," "trip" and "hospital," generally prime what are effectively roles ("shopper") and fillers ("luggage" and "doctor"). And Khalkhali et al. (2012) investigate, over four experiments, the extent that pairs of event primes, and the order in which they are presented, affect subsequent event expectancies: event pairs like *dating-engaged* and *marinate-grill* prime subjects to recognize follow-ups *wedding* and *chew*, respectively, more quickly.

Feature norms have not traditionally found much use within the NLP community at large. Though event priming may seem related to event-based tasks like slot filling (Walker et al., 2006), narrative cloze (Chambers and Jurafsky, 2008), language modeling (Rudinger et al., 2015), and intrusion detection (Chang et al., 2009), the

NLP-based data of which I am aware does not have controlled, cognitively-based human observations and elicitations. That said, there is a robust sub-area that does study them. Făgărăşan et al. (2015) map existing word embeddings into McRae et al. (2005)'s norms, while Greenberg et al. (2015) cluster verb and role pairs to improve the thematic fit of distributional models. Bulat et al. (2016) explore the multimodal nature of feature norms, and Bulat et al. (2017) use McRae et al. (2005)'s norms to generalize over metaphorical language.

5.5 Summary

In this chapter I presented a way to learn embeddings enriched with multiple, automatically obtained frames from large, disparate corpora. This method—a form of generalized tensor factorization—is a general framework for merging modern continuous vector semantics with more classic, structured representations of meaning. By learning continuous representations from millions of both newswire and Wikipedia articles, I empirically demonstrated how the method can be applied to larger data, in terms of the number of documents, the number of individual vocabulary items, and the number of dimensions in the tensor (the relevant features along which to predict words or condition predictions). Future chapters will consider different ways of incorporating semantic frame information to obtain more holistic discourse models and document-level frames.

The method allows multiple types of semantic information to be incorporated. In this chapter, I examined three semantic sources, all automatically obtained through the CAC (chapter 4): two FrameNet parses and one PropBank parse (per sentence). As particularly demonstrated by experiments on out-of-domain data (the Wikipedia-based models), the multiple types of semantic information can be complementary and result in improved correlation judgments. The ability to scale to larger amounts and types of semantic annotations is an enticing prospect, especially given the recent and increasing interest in semantic representations (Abend and Rappoport, 2017).

I considered attributive evaluations on three different datasets: one from the natural language processing community, that built on linguistic theory but could be guided toward information extraction tasks; the other two from the cognitive science community, that, controlled settings, measured humans' biases about how we use language to describe events and their participants. I showed how, overall, these learned embeddings correlate more highly with all three of these datasets, and include syntactic-semantic information that may not always be captured by existing word representation methods. The framework and evaluations presented provide a suite of linguistically- and cognitively-backed evaluations; these can be used to compare different styles of semantic annotation.

By changing the training loss criterion, i.e., whether the embedding model should predict PropBank information or if it should predict semantic role fillers, I was able to change the general types of nearby words; this indicates an ability to modify the

notion of *similarity*. For example, training to predict PropBank information results in models that learn grammatical inflections, while predicting role fillers results in models that learn thematically-related, but not necessarily grammatically-related, words. The ability to change the notion of similarity helps motivate the attributive oracles and evaluations used in this chapter.

Future work could examine incorporating different notions of semantic content in order to capture different notions of relatedness. For example, morphological analyses or ontological (WordNet or VerbNet) generalizations, in addition to semantic parses may help better capture the realizations of certain word forms,[14] generalize to event-bearing, non-verb-based nominals (called *deverbal* events),[15] and better leverage and combine information contained in knowledge-rich, human created resources.

Related to this, future work might also consider different ways of incorporating any *structure* within the semantic ontologies themselves. For instance, recall that FrameNet defines relationships between different semantic frames. If a more specific frame F_2 inherits from a more general frame F_1, then if there is a semantic parse involving F_2, then there also *could* be a semantic parse involving F_1: how easily can

[14]While Cotterell et al. (2017) examined morphological features in 3-tensor factorization, they did not include additional semantic information.

[15]Generally, a deverbal event is a nominal (non-verb) that represents some event. If we describe a fun-yet-tiring party in one of the following ways,

(5.6) We *partied* until the sun came up.

(5.7) The *party* lasted until the sun came up.

where both *partied* and *party* refer to the same party event, then the latter use is often referred to as a deverbal event. See Gurevich et al. (2007) for an overview of the challenges and importance deverbal nouns represent to general knowledge acquisition and representation.

the tensor factorization be modified to reflect this inheritance relation? In particular, would the approach for modeling inter-frame relationships generalize to the different types of relationship? Recall that FrameNet encodes alternations, such as the inchoative and causative. Would the same approach for modeling frame inheritance be able to model these?

Chapter 6

Memoized Sentential Frames

As compared to the previous chapter, which studied frame representations at the word level via *lexical* models, here I am concerned with frame representations at the phrase or clause level. I examine this by inducing deep and lexical grammars at the sentence level. The subsequent chapter will then consider representations at the discourse (document, or inter-sentence) level.

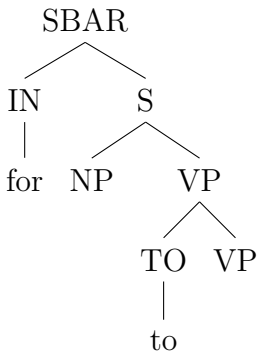
Context-free grammars (CFGs) are a useful tool for describing the structure of language, modeling a variety of linguistic phenomena while still permitting efficient inference. However, it is widely acknowledged that CFGs employed in practice make unrealistic independence and structural assumptions, resulting in grammars that are overly permissive.

One successful approach to learning more accurate grammars has been to refine the nonterminals of grammars, first manually (Johnson, 1998; Klein and Manning,

2003) and later automatically (Matsuzaki et al., 2005; Dreyer and Eisner, 2006; Petrov et al., 2006). In addition to improving parsing accuracy, the automatically learned *latent annotations* of these latter approaches yield results that accord well with human intuitions, especially at the lexical or preterminal level—for example, separating demonstrative adjectives from definite articles under the determiner (DT) tag. It is more difficult, though, to extend this analysis to higher-level nonterminals, where the long-distance interactions among latent annotations of internal nodes are subtle and difficult to trace.

In this chapter I provide a model that extends the split-merge framework of Petrov et al. (2006) to jointly learn latent annotations and Tree Substitution Grammars (TSGs). I argue that these are forms of basic construction grammars. I then conduct a variety of multilingual experiments with this model: first I induce latently annotated grammars from the Penn Treebank (Marcus et al., 1993) and the Korean Treebank 2.0 (Han et al., 2001; Han and Ryu, 2005). Second, I present qualitative, ablation analyses across the models and treebanks that demonstrate the complementary natures of these the latent annotations and memoized structure. These evaluations and analyses are meant to study what deep refinement patterns can be learned, and what linguistic phenomena and predicate argument structures can be derived and captured.[1]

[1]This chapter is an extended version of Ferraro et al. (2012b).



(a) A TSG fragment

SBAR → IN S
IN → for
S → NP VP
VP → TO VP
TO → to

(b) Equivalent CFG rules.

Figure 6.1: A simple example of a TSG fragment and an equivalent representation with a CFG.The SBAR fragment could be used to help analyze the bracketed portion of a sentence such as "Chris wrote the story [for readers to enjoy]," where "readers" and "enjoy" form the noun and verb phrases, respectively.

6.1 Extended Domains of Locality

Many researchers have examined the use of formalisms with an *extended domain of locality* (Joshi and Schabes, 1997), where the basic grammatical units are arbitrary tree fragments instead of traditional depth-one context-free grammar productions. In particular, Tree Substitution Grammars (TSGs) retain the context-free properties of CFGs (and thus the cubic-time inference) while at the same time allowing for the modeling of long distance dependencies. Fragments from such grammars are often considered intuitive (Post and Gildea, 2009b): they capture exactly the sorts of phrasal-level properties and longer-range dependencies(such as predicate-argument structure) that are not present in grammars comprised of rules solely of depth 1, such as standard Treebank CFGs or most grammars with symbol (non-terminal and ter-

		CFG	**TSG**
	none	Charniak '97	Cohn et al. '09
	manual	Klein & Manning '03	Bansal & Klein '10
Node Annotation	**automatic**	Matsuzaki et al. '05 Petrov et al. '06 Dreyer & Eisner '06	Shindo et al. '12 *This chapter*

Table 6.1: Representative prior work in learning refinements for context-free and tree substitution grammars, with zero, manual, or automatically induced latent annotations.

minal) refinements (Klein and Manning, 2003; Matsuzaki et al., 2005; Petrov et al., 2006, e.g.,).[2] This chapter is motivated by the complementarity of local latent refinements and the extended domains of locality. Table 6.1 situates this work among other contributions.

In addition to experimenting directly with the Penn and Korean Treebanks, I also conducted two experiments in this framework with the Universal POS tagset (Petrov et al., 2011). First, I investigate whether the tagset can be automatically derived after mapping all nonterminals to a single, coarse nonterminal. Second, I begin with the mapping defined by the tagset, and investigate how closely the learned annotations resemble the original treebank. Together with the TSG efforts, this chapter is aimed at increased flexibility in the grammar induction process, while retaining the use of Treebanks for structural guidance.

[2] A definition and examples of latent annotations will follow in §6.2.1.

6.2 Background

6.2.1 Latent variable grammars

Latent annotation learning is motivated by the observed coarseness of the nonterminals in treebank grammars, which often group together nodes with different grammatical roles and distributions (such as the role of NPs in subject and object position). Johnson (1998) presented a simple parent-annotation scheme that resulted in significant parsing improvement. Klein and Manning (2003) built on these observations, introducing a series of manual refinements that captured multiple linguistic phenomena, leading to accurate and fast unlexicalized parsing. Later, automated methods for nonterminal refinement were introduced, first splitting all categories equally (Matsuzaki et al., 2005), and later refining nonterminals to different degrees (Petrov et al., 2006) in a split-merge EM framework. This latter approach was able to recover many of the splits manually determined by Klein and Manning (2003), while also discovering interesting, novel clusterings, especially at the lexical level. Although Petrov et al. observed that these grammars *could* provide a deeper form of phrase-level analysis by representing long-distance dependencies through sequences of substates that place all or most of their weight on particular productions, such patterns must be discovered manually via extensive analysis. The automated induction of deep latent-variable grammars (those with extended domains of locality—see §6.2.2) is more difficult, as neither the deep grammatical template rules nor symbol refinements are observed.

6.2.2 Tree Substitution Grammars

Tree substitution grammars (TSGs) allow for complementary analysis. These grammars employ an *extended domain of locality* over traditional context-free grammars by generalizing the atomic units of the grammar from depth-one productions to fragments of arbitrary size. An example TSG fragment along with equivalent CFG rules are depicted in Figure 6.1. The two formalisms are weakly equivalent, and computing the most probable derivation of a sentence with a TSG can be done in cubic time.

Unfortunately, learning TSGs is not straight-forward, in large part because TSG-specific resources (e.g., large scale TSG-annotated treebanks) do not exist. One class of existing approaches, known as Data-Oriented Parsing, simply uses all the fragments (Bod, 1993, DOP). This does not scale well to large treebanks, forcing the use of implicit representations (Goodman, 1996a) or heuristic subsets (Bod, 2001). It has also been generally observed that the use of all fragments results in poor, overfit grammars, though this can be addressed with held-out data (Zollmann and Sima'an, 2005) or statistical estimators to rule out fragments that are unlikely to generalize (Zuidema, 2007). A number of researchers have found success employing Bayesian non-parametric priors (Post and Gildea, 2009a; Cohn et al., 2010), which put a downward pressure on fragment size except where the data warrant the inclusion of larger fragments. Unfortunately, proper inference under these models is intractable, and though Monte Carlo techniques can provide an approximation, the samplers can be

complex, difficult to code, and slow to converge.

This history suggests two approaches to state-split TSGs: (1) a Bayesian non–parametric sampling approach (incorporate state-splitting into existing TSG work), or (2) EM (incorporate TSG induction into existing state-splitting work). We choose the latter path, and in the next section will describe our approach which combines the simplicity of DOP, the intuitions motivating the Bayesian approach, and the efficiency of EM-based state-splitting.

Shindo et al. (2012) proposed a Bayesian non-parametric model following the former option. Their model is a twice-backed off hierarchical model: first latently-refined elementary trees $e \sim \text{PYP}$, using a base distribution factorized according to the constituent refined CFG rules. Each rule is distributed according to another Pitman-Yor process, specifying a uniform base distribution over root-coarsened rules. Using max-rule-product parsing, their full model achieved 91.1 F1 on §23, though without hierarchical backoff they achieved 86.4 F1.

Finally, Bansal and Klein (2010) and Sangati and Zuidema (2011) are modern approaches for DOP-style parsing. Bansal and Klein (2010) combine Goodman (1996a)'s implicit representation with a number of manual refinements described in Klein and Manning (2003), quantitatively demonstrating the complementary of local vs. deeper learning. However, the implicit approach is not able to learn arbitrary distributions over fragments, and the state splits are determined in a fixed pre-processing step. Our approach addresses both of these limitations. Alternatively, Sangati and

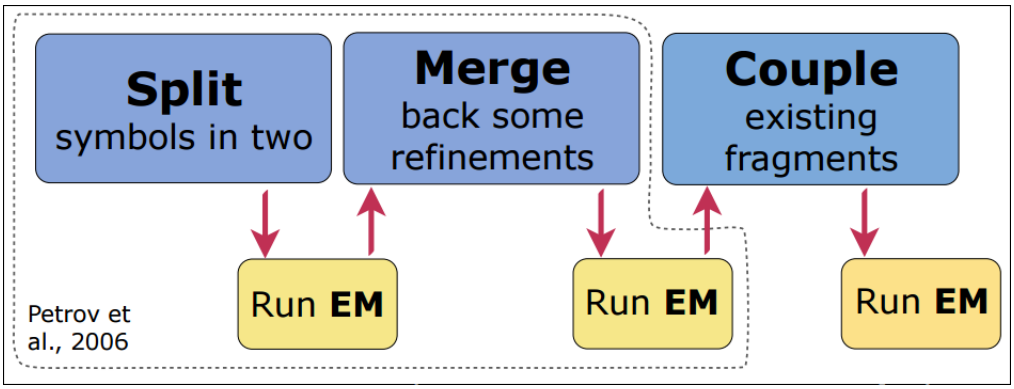


Figure 6.2: A sketch of this chapter's state-split TSG induction algorithm. The split-merge-couple cycle depicted here forms an iteration of the algorithm. Here, EM represents 50 iterations of the inside-outside algorithm (Jurafsky and Martin, 2008, see ch. 14).

Zuidema (2011) present a dynamic programming algorithm for estimating a PTSG from all subtrees that occur at least twice. They achieve competitive F1 by learning these DoubleDOP grammars after running split-merge. Our approach allows joint learning of refinements and larger fragments.

6.3 State-Split TSG Induction

In this section we describe how we combine the ideas of DOP, Bayesian-induced TSGs and Petrov et al. (2006)'s state-splitting framework. As shown in Figure 6.2, we add a **coupling** step to each iteration:

(1) **split** all symbols in two,

(2) **merge** 50% of the splits, and

(3) **couple** existing fragments.

In the split phase, a symbol like NP_i, indicating the ith latent refinement of the NP symbol, is split in two, resulting in new symbols NP_{2i} and NP_{2i+1}; this splitting happens for every non-terminal and terminal symbol. Initially, the *observed* NP symbol is assumed to stand for NP_0. Because every symbol is split in two, a single binary PCFG rule results in eight rules (where new rule probabilities are apportioned uniformly, with some uniform random noise to break ties).

The merging phase attempts to deal with the combinatorial increase in the number of rules from the split phase. This phase undoes some of the changes to the grammar made in the split phase; the extent of the revisions is done in a data-driven way. Briefly, the merging phase approximates the loss in log-likelihood that would occur if two split symbols, e.g., NP_j and NP_{j+1}, were merged back together. Please see Petrov et al. (2006) for more details.

Because every step results in a new grammar with novel symbols and rules, production probabilities are fit to observed data by running at most 50 rounds of EM (the inside-outside algorithm) after every step listed above.[3] We focus on our contribution—the coupling step—and direct those interested in details regarding splitting/merging to Petrov et al. (2006).

Let $\mathcal{T}$ be a treebank and let $\mathcal{F}$ be the set of all possible fragments in $\mathcal{T}$. Define a tree $T \in \mathcal{T}$ as a composition of fragments $\{F_i\}_{i=1}^{n} \subseteq \mathcal{F}$, with $T = F_1 \circ \cdots \circ F_n$. We use

[3]We additionally apply Petrov et al. (2006)'s smoothing step before and after **coupling**.

X to refer to an arbitrary fragment, with r_X being the root of X. Two fragments X and Y may compose (couple), which we denote by $X \circ Y$.[4] We assume that X and Y may couple only if $X \circ Y$ is an observed subtree.

6.3.1 Coupling Procedure

While Petrov et al. (2006) posit all refinements simulatenously and then retract half, applying this strategy to the coupling step would result in a combinatorial explosion. We control this combinatorial increase in three ways. First, we assume binary trees. Second, we introduce a constraint set $\mathcal{C} \subseteq \mathcal{F}$ that dictates what fragments are permitted to compose into larger fragments. Third, we adopt the iterative approach of split-merge and incrementally make our grammar more complex by forbidding a fragment from participating in "chained couplings:" $X \circ Y \circ Z$ is not allowed unless either $X \circ Y$ or $Y \circ Z$ is a valid fragment in the previous grammar (and the chained coupling is allowed by $\mathcal{C}$). Note that setting $\mathcal{C} = \emptyset$ results in standard split/merge, while $\mathcal{C} = \mathcal{F}$ results in a latently-refined DOP-1 model.

We say that $\langle XY \rangle$ represents a valid coupling of X and Y only if $X \circ Y$ is allowed by $\mathcal{C}$, whereas $\overline{\langle XY \rangle}$ represents an invalid coupling if $X \circ Y$ is not allowed by $\mathcal{C}$. Valid couplings result in new fragments. (We describe how to obtain $\mathcal{C}$ in §6.3.3.)

Given a constraint set $\mathcal{C}$ and a current grammar $\mathcal{G}$, we construct a new grammar

[4] Technically, the composition operator ($\circ$) is ambiguous if there is more than one occurrence of r_Y in the frontier of X. Although notation augmentations could resolve this, we rely on context for disambiguation.

$\mathcal{G}'$. For every fragment $F \in \mathcal{G}$, hypothesize a fragment $F' = F \circ C$, provided $F \circ C$ is allowed by$\mathcal{C}$. In order to add F and F' to $\mathcal{G}'$, we assign an initial probability to both fragments (§6.3.2), and then use EM to determine appropriate weights. We do not explicitly remove smaller fragments from the grammar, though it is possible for weights to vanish throughout iterations of EM.

Note that a probabilistic TSG fragment may be uniquely represented as its constituent CFG rules: make the root of every internal depth-one subtree unique (have unit probability) and place the entirety of the TSG weight on the root depth-one rule. This representation has multiple benefits: it not only allows TSG induction within the split/merge framework, but it also provides a straight-forward way to use the inside-outside algorithm (though increasing the grammar size in this way does affect even the pre-bracketed, linear inside-outside algorithm).

6.3.2 Fragment Probability Estimation

First, we define a count function c over fragments by

$$c(X) = \sum_{T \in \mathcal{P}(\mathcal{T})} \sum_{\tau \in T} \delta_{X,\tau}, \tag{6.1}$$

where $\mathcal{P}(\mathcal{T})$ is a parsed version of $\mathcal{T}$, τ is a subtree of T and $\delta_{X,\tau}$ is 1 iff X matches τ.[5] We may then count fragment co-occurrence by

$$\sum_{Y} c(X \circ Y) = \sum_{Y:\overline{\langle XY \rangle}} c(X \circ Y) + \sum_{Y:\langle XY \rangle} c(X \circ Y).$$

Prior to running inside-outside, we must re-allocate the probability mass from the previous fragments to the hypothesized ones. As this is just a temporary initialization, can we allocate mass as done when splitting, where each rule's mass is uniformly distributed, modulo tie-breaking randomness, among its refinement offspring? Split/merge only hypothesizes that a node should have a particular refinement, but by learning subtrees our coupling method hypothesizes that deeper structure may better explain data. This leads to the realization that a symbol may both subsume, and be subsumed by, another symbol in the same coupling step; it is not clear how to apply the above redistribution technique to our situation.

However, even if uniform-redistribution could easily be applied, we would like to be able to indicate how much we "trust" newly hypothesized fragments. We achieve this via a parameter $\gamma \in [0, 1]$: as $\gamma \to 1$, we wish to move more of $\mathbf{P}[X \mid r_X]$ to $\mathbf{P}[\langle XY \rangle \mid r_X]$. Note that we need to know which fragments L couple below with X ($\langle XL \rangle$), and which fragments U couple above ($\langle UX \rangle$).

For reallocation, we remove a fraction of the number of occurrences of top-

[5] We use a parsed version because there are no labeled internal nodes in the original treebank.

couplings of X:

$$\hat{c}\,(X) = 1 - \gamma \frac{\sum_{Y:\langle XY\rangle} c(X \circ Y)}{\sum_{Y} c(X \circ Y)}, \tag{6.2}$$

and some proportion of the number of occurrences of bottom-couplings of X:

$$\check{c}(X) = \frac{\sum_{U:\langle UX\rangle} c(U \circ X)}{\sum_{\substack{U,L:\langle UL\rangle \\ r_X = r_L}} c(U \circ L)}. \tag{6.3}$$

To prevent numerical inconsistencies, such as those possibly caused by sparse pre-terminal counts, (6.2) returns 1 and (6.3) returns 0 as necessary.

Given any fragment X in an original grammar, let ρ be its conditional probability: $\rho = \mathbf{P}\,[X \mid r_X]$. For a new grammar, define the new conditional probability for X to be

$$\mathbf{P}\,[X \mid r_X] \;\propto\; \rho \cdot |\hat{c}(X) - \check{c}(X)|, \tag{6.4}$$

and

$$\mathbf{P}\,[\langle XY\rangle \mid r_X] \;\propto\; \gamma\rho \frac{c(X \circ Y)}{\sum_{Y} c(X \circ Y)} \tag{6.5}$$

for applicable Y.

Taken together, equations (6.4) and (6.5) simply say that X must yield some percentage of its current mass to its hypothesized relatives $\langle XY\rangle$, the amount of which is proportionately determined by $\hat{c}$. But we may also hypothesize $\langle ZX\rangle$, which

Require: Access to a treebank
$S \leftarrow \emptyset$
$\mathcal{F}_{\langle 1,K \rangle} \leftarrow$ top K CFG rules used
for $r = 2$ to R **do**
$S \leftarrow S \cup \{\text{observed 1-rule extensions of } F \in \mathcal{F}_{\langle r-1,K \rangle}\}$
$\mathcal{F}_{\langle r,K \rangle} \leftarrow$ top K elements of $\mathcal{F}_{\langle r-1,K \rangle} \cup S$
end for
return $\mathcal{F}_{\langle R,K \rangle}$, the K most common tree fragments of size at most R

Figure 6.3: The ExtractFragments subtree counting algorithm. This extracts the K most common tree fragments of size (number of decision points) at most R.

has the effect of removing (partial) occurrences of X.[6]

Though we would prefer posterior counts of fragments, it is not obvious how to efficiently obtain posterior "bigram" counts of arbitrarily large latent TSG fragments (i.e., $c(X \circ Y)$). We therefore obtain, in linear time, Viterbi counts using the previous best grammar. Although this could lead to count sparsity, in practice our previous grammar provides sufficient counts across fragments.

6.3.3 Coupling from Common Subtrees

I now turn to the question of how to acquire the constraint set $\mathcal{C}$. Drawing on the discussion in §6.2.2, the constraint set should, with little effort, enforce sparsity. Figure 6.3 provides a simple yet effective method of obtaining this set; in Ferraro et al. (2012a), I detail a downstream grammaticality judgment evaluation of this set, indicating a generalizability beyond constituency parsing. The algorithm extracts a

[6] If $\hat{c}(X) = \check{c}(X)$, then define Eqn. (6.4) to be ρ.

list of the K most common subtrees of size at most R, which I refer to as $\mathcal{F}_{\langle R,K\rangle}$. Note that if $F \in \mathcal{F}_{\langle R,K\rangle}$, then all subtrees F' of F must also be in $\mathcal{F}_{\langle R,K\rangle}$.[7] This algorithm incrementally builds $\mathcal{F}_{\langle R,K\rangle}$ in the following manner: given r, for $1 \leq r \leq R$, maintain a ranking S, by frequency, of all fragments of size r. The key point is that S may be built from $\mathcal{F}_{\langle r-1,K\rangle}$. Once all fragments of size r have been considered, the algorithm retains only the top K fragments of the ranked set $\mathcal{F}_{\langle r,K\rangle} = \mathcal{F}_{\langle r-1,K\rangle} \cup S$, increases r, and repeats the above process.

This incremental approach is appealing for two reasons: (1) practically, it helps temper the growth of intermediate rankings $\mathcal{F}_{\langle r,K\rangle}$; and (2) it provides two tunable parameters R and K, which relate to the base measure and concentration parameter of previous work (Post and Gildea, 2009a; Cohn et al., 2010). I threshold every iteration to enforce sparsity.[8]

In Figure 6.4, I show the number of different rule (fragment) types that the ExtractFragments algorithm returns when extracting fragments from the training portion of the Penn Treebank; specifically, the top 50,000 fragments of at most 31 decision points. Note the y-axis counts over the number of different rules, at the type level: it does not show the total number of *occurrences* of the extracted fragments. Prior to extraction, the input trees were binarized; this will be a necessary step for

[7] Analogously, if an n-gram appears K times, then all constituent m-grams, $m < n$, must also appear at least K times.

[8] Alternatively, I could have adapted Sangati and Zuidema (2011)'s method of acquiring common subtrees. However, Sangati and Zuidema (2011) only extract the maximum common fragments between any two trees. While this limits the exponential growth, it does not easily allow incremental couplings.

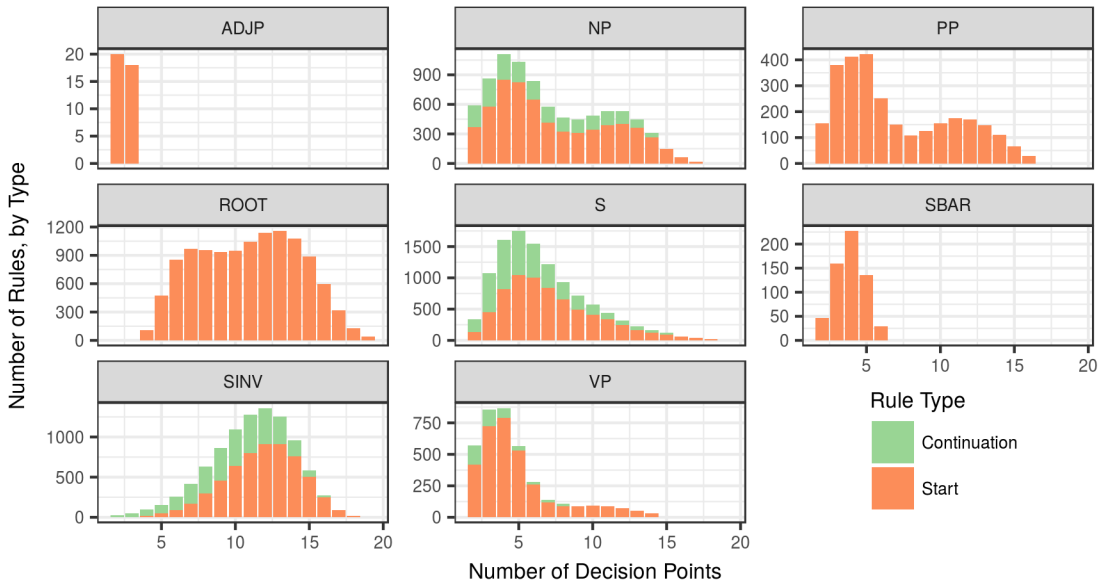


Figure 6.4: Counts, by rule type, of the extracted, non-CFG fragments returned by the ExtractFragments subtree counting algorithm (Figure 6.3). These plots show how many rule (types) begin with a certain grammar symbol vs. the number of decision points (fragment expansions) in that rule: a standard CFG rule has one decision point. Fragments identified as having a "Start" type begin a proper rule; "Continuation" types are a result of the (necessary) binarization step.

parsing. As a result of this, we can identify fragments as starting a rule in the original tree ("Start"), or as a continuation. While many of the resulting distributions are unimodal, ROOT (the topmost node in the tree), NP, and PP fragments are bimodal, indicating the algorithm encodes many different high level composite forms of the respective phrase type; e.g. with ROOT, the algorithm encodes a lot of different ways to form the at-times complex sentences in the treebank.

6.3.4 Construction Grammar

> What makes a theory that allows constructions to exist a "construction-based theory" is the idea that the network of constructions captures our grammatical knowledge of language *in toto*, i.e. it's constructions all the way down.—Goldberg (2006, 18)

Recall from §3.1.2.2 that a construction grammar combines lexical and semantic rules and requirements with the syntactic productions. While I do *not* claim that a state-split TSG *is* a formal construction grammar, I do want to point out some broad similarities. Construction grammar allows phrases and (gappy) sentence fragments to be represented with tree-like structures. Despite this, construction grammars are neither generative nor compositional in the same way that context free grammars (or restricted context sensitive grammars) are: a constructionist analysis relies on "superimposing" (Kay, 1995) tree structures atop one another.

In contrast to the small, yet present, interest in construction grammars in the NLP community over the past two decades, there has been a recent increase in that interest (Hwang et al., 2010; van Trijp et al., 2012; Chen et al., 2011b; Marques and Beuls, 2016). There is not yet a canonical construction grammar standard or treebank in the NLP community; while this could be due to the nascent renewed interest, it could also be due to the number of competing variants of construction grammar on the theoretical side. However, the theoretical fragmentation has not prevented all computational efforts—with van Trijp et al. (2012) developing a system for and theory of one variant ("fluid construction grammar,") and Dodge and Petruck (2014) examining another ("embodied construction grammar").

Consistent in these computational approaches is a representation of meaning through deep syntactic and lexical "constructions." Most *conceptually* relevant to this chapter is the work of Hwang et al. (2010), which develops systems to classify certain reduced tree fragments as encoding a construction or not. They specifically examine constructions that encode "Cause-Motion" events, exemplified by:

(6.6) Chris *shooed* Pat out of the house.

In this example, Chris caused Pat to undergo a (physical) motion. According to their methodology, Hwang et al. only examine "Cause-Motion" events within a very particular syntactic construction.[9] Although Hwang et al. improved detection accuracy of "Cause-Motion" constructs, note that their focus was on both a single construction and a single syntactic pattern. In contrast, this chapter asks what kinds of constructions, at a high level, can be learned.

Above, I covered the complementary nature of latent annotations on nodes and tree substitution grammars. While I do not explicitly force morphological analyses into the latent states or semantic requirements onto the tree fragments, as I describe in below, there is a simple, ready method for including some (naïve) notion of morphology). Similarly, the constraint set and trust parameter γ provide ready ways to prime the induction algorithm to learn (certain types of) verb phrase or sentential constructions.

[9]They only considered sentences whose tree contained the reduced fragment (NP-SUBJ (V NP PP)): that is, a subject noun phrase with a right sibling of a verb phrase, where that verb phrase was built from a verb, noun phrase and prepositional phrase.

6.4 Evaluations and Datasets

In this section I perform two evaluations. The first is a standard parsing evaluation on the English *Wall St. Journal* portion of the Penn TreeBank (Marcus et al., 1993). Second, I perform a number of qualitative analyses of fragments learned on datasets for two languages: the Korean Treebank v2.0 (Han and Ryu, 2005) and the PTB. The Korean Treebank (KTB) has predefined training, development, and testing splits (partitions of the available data); unless otherwise stated, I use the standard PTB splits, which I refer to as WSJ (2-21 for training, 22 for development, 24 for tuning, and 23 for final evaluation). As described in Chung et al. (2010), although Korean presents its own challenges to grammar induction, the KTB yields additional difficulties by including a high occurrence of very flat rules (in 5,000 sentences, there are 13 NP rules with at least four righthand side NPs) and a coarser nonterminal set than that of the Penn Treebank. I run the EM procedure for the same number of iterations on both sets.

Petrov et al. (2011) provided a set of coarse, "universal" (as measured across 22 languages), part-of-speech tags. I explore the interaction of this tagset in the model on WSJ by replacing the original part-of-speech tags with their universal equivalents; I call this modified version UWSJ. Then, as an extreme, in the PTB I replace all POS tags with the same generic symbol "X"; I call this set XWSJ.[10] By further coarsening

[10]While the universal tag set has a Korean mapping, the symbols do not coincide with the KTB symbols.

the PTB tags, I can ask questions such as:

1. What are the refinement patterns?
2. Can we identify linguistic phenomena in a different manner than we might without the universal tag set?
3. What predicate argument relationships can be derived?

6.4.1 Preprocessing

The algorithm is designed to induce a state-split TSG on a binarized tree; as neither dataset is binarized in native form I apply a left-branching binarization across all trees in both collections as a preprocessing step. Petrov et al. (2006) found different binarization methods to be inconsequential, and I have not observed a significant impact of this binarization decision.

I also replace rare words, during both training and evaluation, with naïve morphological analyses. For example, this procedure will take unknown words, such as "wugging" and replace it with "UNK-ing." This is the same morphological analysis module that Petrov et al. uses.

6.4.2 Parsing the English Penn TreeBank

In Table 6.2 I present final English parsing results on the standard evaluation set of the PTB. From earlier development runs, I set the trust parameter γ to 0.5, and I

	≤ 40	all
Petrov et al. (2006)	88.3	87.9
Post and Gildea (2009a)	82.6	–
Cohn et al. (2010)	83.6	82.7
Shindo et al. (2012) : $P^{\text{sr-tsg}}$	–	86.4*
Shindo et al. (2012) : $P^{\text{sr-tsg,sr-cfg}}$	–	89.7*
Shindo et al. (2012) (full)	**91.6***	**91.1***
Shindo et al. (2012) (prod.)	**92.9***	**92.4***
This work	88.3	87.9

Table 6.2: Parsing results on §23 of the WSJ portion of the PTB.

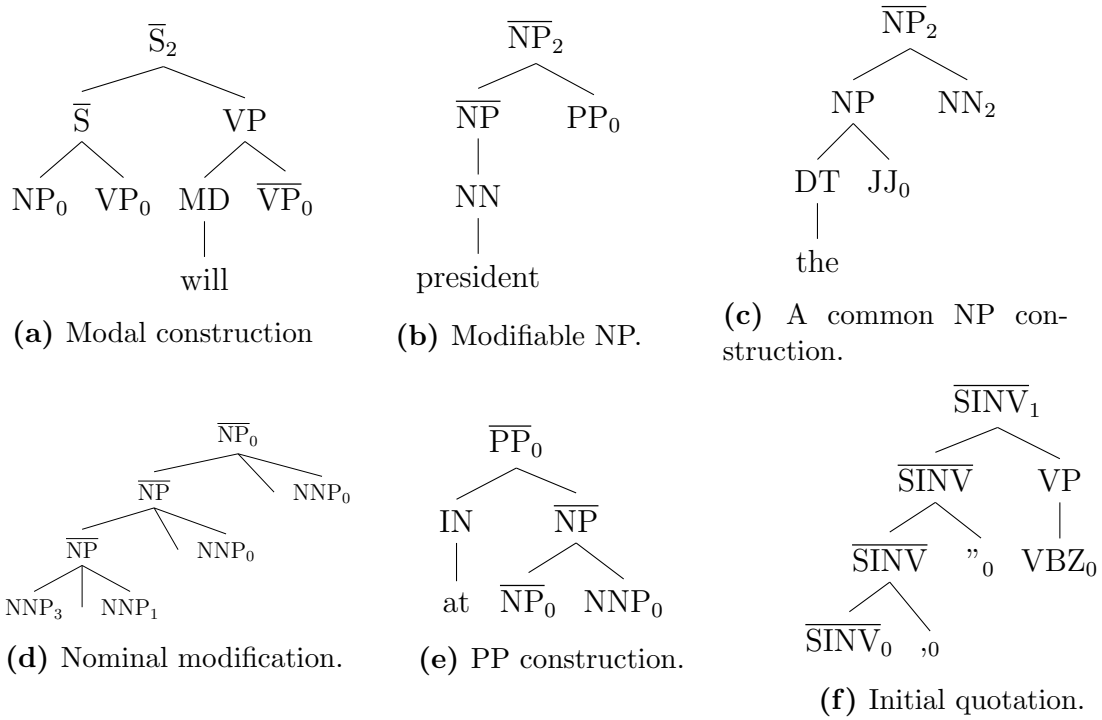
used a constraint set with $R = 31$ and $K = 50,000$. I ran EM for five iterations; like Petrov et al. (2006), I found that going beyond five iterations overfit the resulting grammar.

I compare against one EM-based latent annotation baseline (Petrov et al., 2006), two Bayesian TSG baselines (Post and Gildea, 2009a; Blunsom and Cohn, 2010), and four Bayesian, latent annotation TSG baselines, all from (Shindo et al., 2012, work done concurrently and independently). The EM latent annotation baseline is effectively this latent annotation TSG induction algorithm without any coupling of fragments. The four baselines from Shindo et al. are: (1) $P^{\text{sr-tsg}}$, their Bayesian state split induction with only the first of three hierarchical distributions; (2) $P^{\text{sr-tsg,sr-cfg}}$, their Bayesian state split induction with the first two of three hierarchical distributions; (3) the full model, using all three hierarchical distribution levels; and (4) a product of experts version using their full model.

Scores are computed from Viterbi parses, unless an asterisk is used; in that case,

max-rule parses (Goodman, 1996b) are scored. While Viterbi parses represent derivations that overall had the highest probability of the tree, max-rule parses represent minimum Bayes risk derivations. That is, they are the derivations that maximized the number of expected correctly used rules, according to a variational approximation of the fragment distribution. Typically, max-rule parses result in higher evaluation, indicating a mismatch between the learned distribution and the actual (unknown) one over correctly *annotated* trees (Petrov, 2011).

While the LAPTSG induction does not result in an improvement over the EM baseline, it does improve on the other TSGs without latent annotations. Shindo et al. report overall very strong results, particularly when all three backoff levels, and a product of experts combination, are used. While Shindo et al.'s full system outperforms mine, note that my Viterbi results outperform their max-rule results on just the symbol-refined TSGs (when they do not include backoff model). Shindo et al.'s model without backoff is the most similar to the one described in this chapter; that their backoff models outperform this chapter's model, while the non-backoff model does not, suggests that more aggressive smoothing, coupling estimation (§6.3.2), or constraint set selection could result in higher F_1 scores. However, the EM procedure presented here is more competitive, provides a more straight-forward way to inject prior knowledge, and is arguably simpler than the Bayesian approach.



(a) Modal construction

(b) Modifiable NP.

(c) A common NP construction.

(d) Nominal modification.

(e) PP construction.

(f) Initial quotation.

Figure 6.5: Example fragments learned on WSJ.

6.4.3 Fragment Analysis

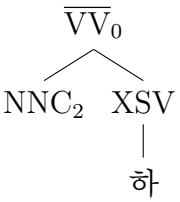
In this section I analyze hand-selected preliminary fragments and lexical clusterings the system learns.

The Wall Street Journal: Penn TreeBank

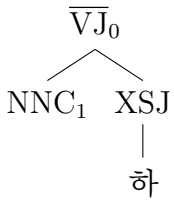
As Figure 6.5 illustrates, after two iterations we learn various types of descriptive lexicalized and unlexicalized fragments. For example, Figure 6.5a concisely creates a four-step modal construction (*will*), while 6.5b demonstrates how a potentially useful nominal can be formed. Further, learned fragments may generate phrases with

	NNC		
0	경우	이날	현재
	case	*this day*	*at the moment*
1	국제	경제	세계
	international	*economy*	*world*
2	관련	발표	보도
	related	*announcement*	*report*

(a) Common noun refinements



(b) Verbal inflection.



(c) Adjectival inflection

Figure 6.6: Clusters and fragments for the KTB.

multiple nominal modifiers (6.5d), and lexicalized PPs (6.5e).

Phrases such as $\overline{\text{NP}}_0$ and $\overline{\text{VP}}_0$ are often lexicalized themselves with determiners, common verbs and other constructions. These lexicalized phrases could be very useful for 6.5a (given the incremental coupling employed, 6.5a could not have been further expanded in two iterations). Figure 6.5d demonstrates how TSGs and latent annotations are naturally complementary: the former provides structure while the latter describes lexical distributions of nominals.

Figure 6.5f illustrates a final example of syntactic structure, as we begin to learn how to properly analyze a complex quotation. A full analysis requires only five TSG rules while an equivalent CFG-only construction requires eight.

Korean TreeBank

To illustrate emergent semantic and syntactic patterns, we focus on common noun (NNC) refinements. As seen in Table 6.6a, top words from NNC_0 represent time- and planning-related expressions. As a comparison, two other refinements, NNC_1 and NNC_2, are not temporally representative. This distinction is important as NNC_0 easily yields adverbial phrases, while the resultant adverbial yield for either NNC_1 or NNC_2 is much smaller.

Comparing NNC_1 and NNC_2, we see that the highest-ranked members of the latter, which include *report* and *announcement*, can be verbalized by appending an appropriate suffix. Nouns under NNC_1, such as *economy* and *world*, generally are subject to adjectival, rather than verbal, inflection. Figures 6.6b and 6.6c capture these verbal and adjectival inflections, respectively, as lexicalized TSG fragments.

***The Wall Street Journal*, Universal Tag Set**

In the small study done here, we find that after a small number of iterations we can identify various cluster classifications for different POS tags. Figures 6.7a, 6.7b and 6.7c provide examples for NOUN, VERB and PRON, respectively. For NOUNs we found that refinements correspond to agentive entities (refinements 0, 1, e.g., corporations or governments), market or stock concepts (2), and numerically-modifiable nouns (7). Some refinements overlapped, or contained common nouns usable in many different contexts (3).

	NOUN		
0	Corp	Big	Co.
1	Mr.	U.S.	New
2	Bush	prices	trading
3	Japan	September	Nissan
7	year	%	months

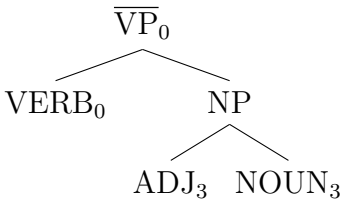
(a) Noun refinements

	VERB				
0	says	said	sell	buy	rose
1	have	had	has	been	made
2	said	says	say	added	believe
3	sold	based	go	trading	filed
7	is	are	be	was	will

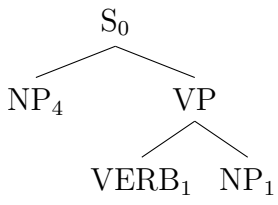
(b) Verb refinements

	PRON		
1	its	his	your
2	who	whom	—
3	what	whose	What
5	it	he	they
6	it	them	him

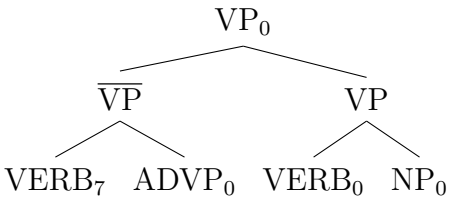
(c) Pronoun refinements



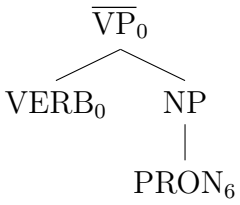
(d) VP structure



(e) Declarative sentence



(f) Multiple VP interactions



(g) Accusative use

Figure 6.7: Highest weighted representatives for lexical categories (6.7a-6.7c) and learned fragments (6.7d-6.7g), for UWSJ.

Similarly for VERBs (6.7b), we find suggested distinctions among action (1) and belief/cognition (2) verbs.[11] Further, some verb clusters are formed of eventive verbs, both general (3) and domain-specific (0). Another cluster is primarily of copula/auxiliary verbs (7). The remaining omitted categories appear to overlap, and only once we examine the contexts in which they occur do we see they are particularly useful for parsing FRAGs.

Though NOUN and VERB clusters can be discerned, there tends to be overlap among refinements that makes the analysis more difficult. On the other hand, refinements for PRON (6.7c) tend to be fairly clean and it is generally simple to describe each: possessives (1), personified *wh*-words (2) and general *wh*-words (3). Moreover, both subject (5) and object (6) are separately described.

Promisingly, we learn interactions among various refinements in the form of TSG rules, as illustrated by Figures 6.7d-6.7g. While all four examples involve VERBs it is enlightening to analyze a VERB's refinement and arguments. For example, the refinements in 6.7d may lend a simple analysis of financial actions, while 6.7e may describe different NP interactions (note the different refinement symbols). Different VERB refinements may also coordinate, as in 6.7f, where participle or gerund may help modify a main verb. Finally, note how in 6.7g, an object pronoun correctly occurs in object position. These examples suggest that even on coarsened POS tags, our method is able to learn preliminary joint syntactic and lexical relationships.

[11] The next highest-ranked verbs for refinement 1 include *received*, *doing* and *announced*.

	X			Universal Tag
0	two	market	brain	NOUN
1	's	said	says	VERB
2	%	company	year	NOUN
3	it	he	they	PRON
5	also	now	even	ADV
6	the	a	The	DET
7	10	1	all	NUM
9	.	–	...	.
10	and	or	but	CONJ
12	which	that	who	PRON
13	is	was	are	VERB
14	as	of	in	ADP
15	up	But	billion	ADP

Table 6.3: Top-three representatives for various refinements of the general lexical preterminal "X," with reasonable analogues to Petrov et al. (2011)'s tags. Universal tag recovery is promising.

The Wall Street Journal, Preterminals as X

In this experiment, we investigate whether the manual annotations of Petrov et al. (2011) can be re-derived through first reducing one's non-terminal tagset to the symbol X and splitting until finding first the coarse grain tags of the universal set, followed by finer-grain tags from the original treebank. Due to the loss of lexical information, we run our system for four iterations rather than three.

As observed in Table 6.3, there is strong overlap observed between the induced refinements and the original universal tags. Though there are 16 refinements of X, due to lack of cluster coherence not all are listed. Those tags and unlisted refinements seem to be interwoven in a non-trivial way. We also see complex refinements of both

open- and closed-class words occurring: refinements 0 and 2 correspond with the open-class NOUN, while refinements 3 and 12, and 14 and 15 both correspond with the closed classes PRON and ADP, respectively. Note that 1 and 13 are beginning to split verbs by auxiliaries.

6.5 Summary

In this chapter, I have shown that tree substitution grammars may be encoded and induced within a framework of syntactic latent annotations. The specific induction algorithm was a constrained EM estimation; the constraints are simply lists of allowable tree fragments. In doing so, I presented two algorithms: a constraint extraction algorithm and a grammar induction algorithm. I provide external validation for the former elsewhere (Ferraro et al., 2012a).

The grammar induction algorithm provides competitive performance against strong baselines. It also out-performs two other (basic) tree substitution algorithms as well as an independent latently annotated TSG induction system, when the latter has not been trained with extensive backoff smoothing. Given the centrality of aggressive smoothing and backoff to the improved performance, and due to the deterministic nature of both obtaining and utilizing the constraint set, this suggests that using an enlarged, or more precisely targeted, constraint set could improve performance.

I provide qualitative analyses for learned, latently annotated tree fragments for both the English portion of the Penn TreeBank and the Korean Treebanks, thereby demonstrating the induction algorithm's ability to apply to other languages. I also experimented with the Universal Part of Speech tagset to represent the initial preterminal symbols. My constraint-based induction algorithm learns nested fragments that handle the internals of complex verb phrases, reporting constructions and nominal modification. In the extreme, it can also reconstruct parts of speech. This also suggests that while the induction algorithm is not *fully* unsupervised—it still requires an input tree structure—that structure can be very minimal. This is encouraging for any future constructions of treebanks: even with a relatively small amount of human-labeled data, the algorithm can still extrapolate to deeper linguistic patterns and phenomena.

This chapter's main focus was on inducing deeper syntactic frames. Unlike the previous chapter, where an *explicit* semantic representation was included to better induce word meanings, here semantic representations are defined more implicitly, through a probabilistic grammar. Nevertheless, deep structures governing verbal and sentential compositions can readily be extracted. The probabilistic nature of these deep structures—such as those constructing nominal modifiers (Figure 6.5d), inflection patterns (Figure 6.6), or different verbal constructs (Figure 6.7)—compactly represents likely "fillers" for these syntactic frames. Meanwhile, the symbol refinement enforces cohesion among those fillers.

For the experiments in this chapter, the constraint set was obtained deterministically, as simply the most common tree fragments up to a certain size that are found in a treebank corpus. While in external work I showed that this counting method was effective for an educational task, future efforts could explore using a more targeted constraint set. One example of targeting might be to require any verb phrase fragment to be lexicalized—similar to the adaptor grammars presented by Johnson et al. (2007). This forced lexicalization might be able to better tie predicates and their arguments together.

In chapter 5, I examined multiple efforts at encoding expectations and default meanings through the use of features, or attributes. Future work could also consider incorporating those types of features—be they inherent to a particular object, such as that "sledgehammers" are likely to be heavy, or derived from a joint consideration of predicates and their arguments—either explicitly into the tree structures, or implicitly through verification with ontologies. That is, the tree fragments could themselves be modified to include features, as in head-driven phrase structure grammar (Pollard and Sag, 1994), or the split, merge, or coupling phases of Figure 6.2 could be modified to use ontological features to help re-estimate and reweight new fragments. Along a similar line, feature incorporation could arise when obtaining the constraints. Regardless, these improvements could incorporate deeper meanings into syntactic analysis.

Chapter 7

A Unified Bayesian Model of Scripts, Frames and Language

Recall from §3.1 that frames or scripts describe prototypically complex situations in terms of certain events, actions, actors and other pieces of information we expect to be involved. These theories posit that for many situations we encounter, there is a **template** with a number of **slots** that need to be filled in order to understand the situation. For example, we partially describe a BOMBING situation with a `Detonation` action, along with those involved, e.g., BOMBERS and VICTIMS.

In chapter 5 I examined how semantic frames can be used to enhance the meaning of individual words. In chapter 6 I then considered how inducing latent structure over individual phrases—a type of "syntactic template"—helped enhance the overall meaning of sentences containing those phrases. Now in this chapter, I am concerned

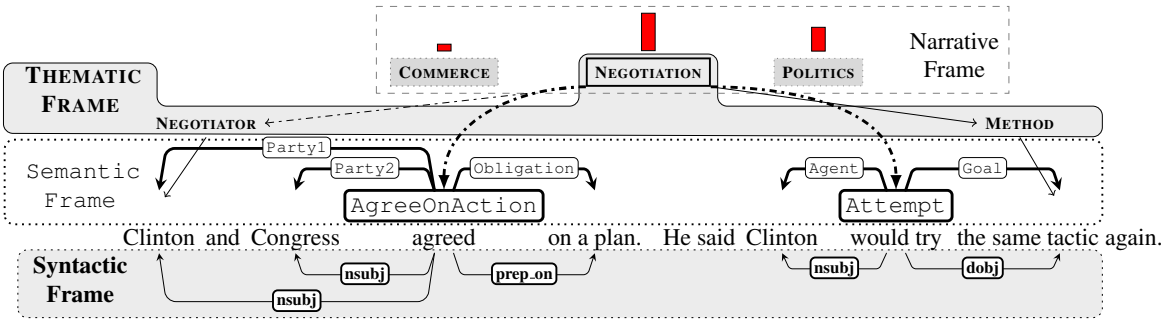


Figure 7.1: An interpretation of Minsky's four frame levels on two newswire sentences, adapted from the automatically labeled version of NYT_ENG_19980330.0346 in Ferraro et al. (2014).

with modeling discourse at the document level. Here, my goal is more closely tied to trying to model our intuitions about commonly reported events, much in line with the earlier goals of classic AI and cognitive science (Minsky, 1974; Schank, 1975; Fillmore, 1975). I will examine the question of downstream applicability in chapter 8.

In this chapter, I present the first probabilistic model to capture all levels of the Minsky Frame structure, with the goal of corpus-based induction of scenario definitions. This model unifies prior efforts in discourse-level modeling with that of Fillmore's related notion of frame, as captured in sentence-level, FrameNet semantic parses. As part of this, I resurrect the *theoretical* coupling among Minsky's frames, Schank's scripts and Fillmore's frames, as originally laid out by those authors. Empirically, I examine the effect of semantic frame information on narrative schemas learned via this unified model, finding that incorporating FrameNet-based semantic frames yields improved scenario representations, reflected quantitatively in lower

surprisal and more coherent latent scenarios.[1]

7.1 A Deeper Look at Frames

Syntactic-based corpus statistics have repeatedly been used to induce approximate, probabilistic versions of these templates; these approaches generally compute verb and syntactic relation cooccurrences from automatically generated dependency parses (Cheung et al., 2013; Bamman et al., 2013; Chambers, 2013, i.a.). These parses can serve as a limited proxy for sentence meaning, owing to information conveyed via the syntax/semantics interface. Rudinger and Van Durme (2014) argue, however, that they do not however fully and explicitly represent a semantic analysis.

Fillmore's notion of frame semantics ties a notion akin to Minsky's frames to individual *lexical items* (Fillmore, 1976, 1982). Word meaning is defined in terms of the roles words play in situations they typically *invoke*, and in how they interact with other lexical items.

In the following I present a probabilistic model which unifies discourse-level Minskian frames with Fillmore's frame semantics. Despite the historical and intellectual connections between these theories, previous empirical efforts have focused on just one or the other: this model is the first to make the connection explicit. I show how current efforts in discourse modeling, and semantic frame induction and identification can be combined in a single model to capture what classic AI theory posited. Quan-

[1] This chapter is an extended version of Ferraro and Van Durme (2016).

titatively, by using a frame-semantic parser pre-trained on FrameNet (Baker et al., 1998), I show that incorporating frame information provides both a better fit to held-out data and improved coherence (Mimno et al., 2011). This unified probabilistic model provides a principled mathematical way of restating Minsky's argument for the four frame levels, and the results show that it is a legitimate way to capture what Minsky proposed.

Minsky, along with a number of contemporaries, believed in schematizing common situations and experiences into *"chunks"*, or *frames*. These frames contain world knowledge that would allow artificial intelligence systems to encounter various occurrences and react appropriately. For Minsky, frames were data structures, with *slots*, to "[represent] a stereotyped situation." Some slots and conditions could have default values; entities (references to an "object" in the world) and pointers to other frames could fill slots. [2]

Minsky (1974) outlined four different "levels" of frames:

Surface Syntactic Frames "Mainly verb and noun structures. Prepositional and word-order indicator conventions."

Surface Semantic Frames "Action-centered meanings of words. Qualifiers and relations concerning participants, instruments, trajectories and strategies, goals,

[2] In addition, Minsky (1974) described systematic and algorithmic ways for handling frames—a frame framework, if you will—as much he described frames themselves. However, I focus in this thesis on the *structural* aspects, rather than the *algorithmic*. As discussed briefly in §3.3.2, there have been efforts to incorporate more complete narrative theories — such as rhetorical structure theory (William and Thompson, 1988) into template induction.

consequences and side-effects."

Thematic Frames "Scenarios concerned with topics, activities, portraits, setting."

Narrative Frames "Skeleton forms for typical stories, explanations, and arguments. Conventions about foci, protagonists, plot forms, development, etc., designed to help a listener construct a new, instantiated Thematic Frame in his own mind."

Figure 7.1 illustrates an interpretation of these four levels on newswire automatically tagged with syntactic and semantic frames, and example thematic and narrative frames.

These hierarchical levels require attention to different aspects of language; as one changes levels, details highly relevant to one may become displaced by more appropriate aspects of another. Information important for the syntactic level may be relevant to, e.g., the thematic or narrative level through an abstracted or "coarsened" version. For instance, in Figure 7.1 the syntactic (below) and surface semantic (above) frames provide the lowest-level intrasentential analyses of this abbreviated document (Latin font). The NEGOTIATION template (thematic frame) fills two of its *slots*, NEGOTIATOR and METHOD, intersententially, with "Clinton" and "tactic," using predicate and dependency information from some combination of the syntactic and surface semantic frames. Here, "Clinton" is highlighted twice stressing that thematic frames may both produce and rely on information across sentences. The narrative frame invokes the NEGOTIATION thematic frame, though related themes PASSING LEGISLATION and POLITICS may appear elsewhere in the document.

This chapter adopts the interpretation of Figure 7.1. Specifically, I assume the lower-level syntactic and surface semantic frames are localized analyses, restricted to sentences, while the higher-level thematic and narrative frames allow for a global analysis, aggregating information across sentences.

While many people are familiar with Schank and Abelson (1977)'s formulations of scripts, the connection between frames and scripts is at times forgotten:

> ... a frame is a general name for a class of knowledge organizing techniques that guide and enable understanding. Two types of frames that are necessary are SCRIPTS and PLANS. Scripts and plans are used to understand and generate stories and actions — Schank (1975).

Schankian scripts are thus a distinct sub-type of Minskian frames. Broadly, scripts introduce a mechanism for ordering events within frames. For simplicity this chapter's model does not encode order. It does, though, provide a framework for future efforts to incorporate ordering, perhaps utilizing some prior ordering efforts. I discuss this later on.

Fillmore's case grammar and frame semantics (Fillmore, 1967, 1976, 1982) posit that word meaning is defined in terms of the roles they play in situations they typically *invoke*, and then in how they interact with other lexical items. We can think of Fillmore as being 'Minsky over words,' where Fillmore's ideas can be realized within the broader development of frames during the 1970s:

> [frames are] certain schemata or frameworks of concepts or terms which link together as a system, which impose structure or coherence on some aspect of human experience, and which may contain elements which are simultaneously parts of other such frameworks. — Fillmore (1975).

As discussed in §3.3, the FrameNet Project (Baker et al., 1998) is an ongoing effort to implement Fillmore's frames.

7.2 Unlabeled Induction with Frames

The model, detailed formally in Figure 7.2 and informally in Figure 7.1, captures the *"ingredients"* of a frame structure at all frame levels posited by Minsky (1974): Surface Syntactic (syntactic dependencies), Surface Semantic (FrameNet semantic parses), Thematic (templates), and Narrative (document-level mixtures over templates). Prior work has either conflated multiple levels together, or otherwise ignored levels entirely: inclusion of these levels as distinct model components is novel to this work.

7.2.1 Generative Story

Following prior efforts I assume that both coreference resolution and a syntactic analysis have been performed on the documents as part of corpus processing (Bamman et al., 2013; Chambers, 2013, i.a.). To learn a model, I assume an automatically produced semantic frame analysis, such as from FrameNet, too.[3] Overall, I analyze each document as a bag of entities (coreference chains), with each entity having one or more mentions. Each entity mention is syntactically governed through a typed

[3]In order for fair comparisons, I treat the semantic frame analysis as latent during heldout evaluation.

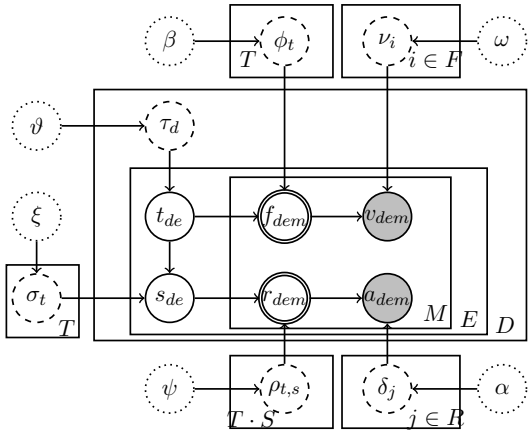
dependency arc (a) to a verb lemma (v). Each verb evokes a surface semantic frame (f), which is related to the entity mention through a frame role (r).

The observations and latent assignments are discrete; I place conjugate Dirichlet priors with symmetric hyperparameters on each. See Figure 7.2 for a formal diagram and variable gloss table. The narrative frame of a document d is represented as a mixture over the set of templates T (Minsky's thematic frames), $\tau_d \sim \text{Dir}(\vartheta)$.

Each template t, such as representing NEGOTIATION, is represented by a distribution σ_t over S unique slots, such as the NEGOTIATOR, and a distribution ϕ_t over F semantic frames (which will come from FrameNet). Both sets of distributions have Dirichlet priors, $\sigma_t \sim \text{Dir}(\xi)$, $\phi_t \sim \text{Dir}(\beta)$.

Every semantic frame i has a distribution over verb lemmas, $\nu_i \sim \text{Dir}(\omega)$, and each slot has a distribution $\rho_{t,s}$ over R frame roles, $\rho_{t,s} \sim \text{Dir}(\psi)$. Just as every semantic frame has a distribution over verb lemmas, every role j has a distribution over syntactic relations $\delta_j \sim \text{Dir}(\alpha)$.

An entity e is assigned to a single (latent) template $t_{d,e}$ and slot $s_{d,e}$, where $t_{d,e} \sim \text{Cat}(\tau_d)$ and $s_{d,e} \sim \text{Cat}\left(\sigma_{t_{d,e}}\right)$. For every mention m of e, the entity template $t_{d,e}$ directly influences the selection of the mention's frame assignment $f_{d,e,m} \sim \text{Cat}\left(\phi_{t_{d,e}}\right)$, and the slot $s_{d,e}$ directly influences the frame role $r_{d,e,m} \sim \text{Cat}\left(\rho_{s_{d,e}}\right)$. For instance, in Figure 7.3 we could replace Clinton's ⟨LATENT⟩ template and slot values with NEGOTIATION and NEGOTIATOR, respectively. The semantic frames `AgreeOnAction` and `Attempt` would both be attributed to the NEGOTIATION template, while the corre-



(a) Shaded nodes, such as verbs and relations (v_{dem}, r_{dem}), are always observed, while double-edged nodes may or may not be observed; all others are latent. Solid-edged nodes such as t_{de} have collapsed priors (dashed edges, e.g.: τ_d) with optimized hyperparameters (dotted edges, e.g.: ϑ).

Variable	Meaning	Minsky
τ_d	document's dist. of templates (themes)	Narrative
σ_t	dist. of template-specific slots	Thematic
ϕ_t	dist. of template-specific semantic frames	Semantic
$\rho_{t,s}$	dist. of slot-specific semantic roles	Semantic
ν_i	dist. of semantic frame's syntactic realization	Syntactic
δ_j	dist. of semantic role's syntactic realization	Syntactic
$t_{d,e}$	template of entity e	Thematic
$s_{d,e}$	template-specific slot of entity	Thematic
$f_{d,e,m}$	semantic frame governing mention	Semantic
$r_{d,e,m}$	mention's semantic role	Semantic
$v_{d,e,m}$	governing predicate of mention	Syntactic
$a_{d,e,m}$	predicate-typed dependency of mention	Syntactic

(b) Brief meaning gloss of the model's variables, with the corresponding Minsky frame levels, given a document d, each of its corefence chains e, and each mention m of e. For simplicity, the hyperparameters (the dotted nodes in Figure 7.2a) are omitted.

Figure 7.2: The unified probabilistic frames model.

sponding roles would be attributed to the NEGOTIATION-*specific* slot NEGOTIATOR.

Finally, the syntactic verb and syntactic relation surface forms are chosen given the frame and role, respectively: $v_{d,e,m} \sim \text{Cat}\left(\nu_{f_{d,e,m}}\right)$, and $a_{d,e,m} \sim \text{Cat}\left(\delta_{r_{d,e,m}}\right)$. For instance, in Figure 7.3, "agree" is attributed to `AgreeOnAction` and "nsubj-agree" is attributed to the typed semantic role `Party1-AgreeOnAction`.

7.2.2 Model Discussion

Like many other research efforts, while I observe syntax ($v_{d,e,m}$), I assume that syntactic dependencies $a_{d,e,m}$ are predicate specific: for the syntactic subject of the verb "attempt," the dependency is "nsubj-attempt." These kinds of observed relations are called *typed dependencies*. I assume that the semantic frame roles are typed as well. Beyond linguistic arguments for this typing (Ruppenhofer et al., 2006, § 3.2), I, like Chambers (2013), have found the learned model to be more amenable to introspection when r and a are typed by their corresponding frame or verb.[4]

This model views these as separate observations without any direct (statistical) influence between the two. In the past, these typed dependencies have not been modeled directly (Chambers, 2013; Cheung et al., 2013). While Lorenzo and Cerisara (2012) use separate distributions for each verb and Bamman and Smith (2014) use an exponential family parametrization, they operate at different scales than I do: Lorenzo and Cerisara use fewer verb types, while Bamman and Smith use a significantly

[4]This verb/frame duplication reflects the strong linguistic intuition that syntactic preferences heavily influence semantic roles (Chomsky, 1981, θ-criterion).

Entity: Clinton			
template	⟨LATENT⟩		
slot	⟨LATENT⟩		
	Mention #1	**Mention #2**	
frame	`AgreeOnAction`	`Attempt`	
role	`Party1-AgreeOnAction`	`Agent-Attempt`	...
verb/pred.	agree	would-try	
dep. arc	nsubj-agree	nsubj-would-try	

Entity: tactic		
template	⟨LATENT⟩	
slot	⟨LATENT⟩	
	Mention #1	
frame	`Attempt`	
verb/pred.	would-try	
dep. arc	dobj-would-try	...

⋮

Figure 7.3: A view of the observed semantic and syntactic levels, as well as the latent thematic level, on the example document in Figure 7.1. Notice how entities do not have to be animate. The highlighted variables (**t**, **s**, **f**, **r**, **v** and **a**) correspond to those in Figure 7.2.

reduced relation set.

The model observes *at most* the syntactic and semantic levels. The thematic and narrative levels are always latent. Figure 7.3 demonstrates this on a portion of the Figure 7.1 document.

Due to the preprocessing requirements, this model is limited to languages with sufficient resources. However, recent efforts in low-resource semantic role labeling (Naradowsky et al., 2012; Gormley et al., 2014) and multilingual (semantic) frame induction (Lorenzo and Cerisara, 2012; Modi et al., 2012; Henderson et al., 2013)

suggest promising avenues for future work.

7.2.3 Comparison to Contemporary Frame Learning

There have been various styles of models in the spirit of this chapter, though none capture all four levels of the Minsky hierarchy. The most similar are the three concurrent Bayesian template models (Bamman et al., 2013; Chambers, 2013; Cheung et al., 2013). Like this work, the former two view documents as collections of prespecified entities and mentions. They similarly incorporate narrative, thematic and syntactic levels, as documents are modeled as mixtures over templates relying on syntactic information. Subsequent work from Bamman and colleagues has refined event participant descriptions or ascribing temporal attributes to atomic events, rather than exploring hierarchical event substructure, as I do (Bamman et al., 2014; Bamman and Smith, 2014). None of these efforts have incorporated separate semantic and syntactic Minskian frames.

Cheung et al. (2013) model ordering of syntactic clauses, grouping predicates into latent events, and a predicate's arguments to event slots. A latent "frame" assignment stratifies templates more coherently across the clauses and throughout the document. In the Minskian terminology used here, they have two layers of thematic frame, but, as above, no layer of semantic frame.

A number of other efforts in learning semantic frames consider syntactic information, though there has not been a presentation incorporating both narrative and thematic components (Titov and Klementiev, 2011; Materna, 2013; Modi et al., 2012; Lorenzo and Cerisara, 2012; Bejan, 2008; Modi and Titov, 2014). Temporal scripts have been learned with graph algorithms (Regneri et al., 2010), Bayesian model merging (Orr et al., 2014), and permutation priors (Frermann et al., 2014), i.a.. These incorporate a rich narrative level, though without thematic frames: the narrative level deals directly with the semantic or syntactic frames.

While other efforts have focused on both generative and discriminative models for less-than-supervised frame induction (Minkov and Zettlemoyer, 2012; Huang and Riloff, 2013; Patwardhan and Riloff, 2009, i.a.), of particular note are those incorporating event "triggers," reminiscent of Rosenfeld's trigger language models (Rosenfeld, 1994, 1996; Van Durme and Lall, 2009). Some of those efforts have identified which verbs trigger events (Chen et al., 2011a, working between the syntactic and semantic levels), while others have focused on discourse relation (Maslennikov and Chua, 2007, working between the narrative and syntactic levels).

Multiple efforts have formulated global (document-level) and local (sentence-level) constraints for supervised graphical models. Reichart and Barzilay (2012)'s factor graph with global and local potentials presents an extensive narrative level that incorporates both thematic and syntactic levels, but excludes the semantic. Both Liao and Grishman (2010) and Li et al. (2013) encode the thematic, semantic and syntactic

levels, but no narrative level.

The Penn Discourse Treebank (Prasad et al., 2008, PDTB) provides both explicit and implicit discourse and causality annotations atop original syntactic annotations of the WSJ portion of the Penn Treebank. As PDTB annotations are both cross-sentential and intrasentential discourse relations, we can view the PDTB as a type of thematic frame. Although with some additional effort Minsky's surface semantic frames could be incorporated—e.g., by aligning PDTB with shallow semantic annotations, such as from PropBank—the narrative level is missing.

7.3 Inference via Collapsed Gibbs Sampling

I fit the model via Gibbs sampling, collapsing out the priors on all latent and observed variables and optimizing the hyperparameters with fixed-point iteration (Wallach, 2008). Posterior inference follows Griffiths and Steyvers (2004). In the following, I derive the complete conditionals of the template variables, with the respective priors integrated out; the calculations for slot, frame and role variables are similar.

In general, for a set of conditionally i.i.d. Categorical variables $z_i|\theta \sim \text{Cat}(\theta)$, where θ has a $\text{Dir}(\alpha)$ prior, the joint probability of all $\mathbf{z}$ is given by the Dirichlet-

Multinomial compound distribution $\mathrm{DMC}(\mathbf{z}|\alpha)$:

$$p_\alpha(\mathbf{z}) = \int_\theta p(\mathbf{z}|\theta) p_\alpha(\theta) d\theta \tag{7.1}$$

$$= \frac{\Gamma\left(\sum_k \alpha_k\right)}{\Gamma\left(\sum_k \left(c(k) + \alpha_k\right)\right)} \prod_k \frac{\Gamma\left(c(k) + \alpha_k\right)}{\Gamma\left(\alpha_k\right)} \tag{7.2}$$

$$= \mathrm{DMC}(\mathbf{z}|\alpha) \tag{7.3}$$

where $c(k)$ is the number of z_i with value k. This can be generalized to a gated version: given a collection of i.i.d. M Dirichlet samples $\theta_m \sim \mathrm{Dir}(\alpha)$ and indicator variables y_i, if $z_i|y_i, \theta \overset{i.i.d}{\sim} \mathrm{Cat}\left(\theta_{y_i}\right)$, then we may consider the collection $[\mathbf{z}]_{\mathbf{y}=m}$ — only those z_i such that $y_i = m$. Then

$$p_\alpha(\mathbf{z}; \mathbf{y}) = \prod_{m=1}^{M} \left(\mathrm{DMC}\left([\mathbf{z}]_{\mathbf{y}=m} | \alpha\right)\right) \tag{7.4}$$

$$= \prod_{m=1}^{M} \left(\frac{\Gamma\left(\sum_k \alpha_k\right)}{\Gamma\left(\sum_k \left(c(m,k) + \alpha_k\right)\right)} \times \prod_k \frac{\Gamma\left(c(m,k) + \alpha_k\right)}{\Gamma\left(\alpha_k\right)}\right), \tag{7.5}$$

where $c(m,k)$ is the number of z_i with value k whose corresponding $y_i = m$.

For our unified frames model, the complete conditionals follow the basic form and derivation given by (Griffiths and Steyvers, 2004). Note that multiple obsevations are attributable to a single latent choice, e.g., for every entity e, all $\#(f \in e)$ instances of frame $f \in e$ are attributable to the template choice $t_{d,e}$. Due to this model topology, we appeal to the general form of the Gamma factorial expansion: for real x and

integral n, $\Gamma(x+n) = \left(\prod_{i=0}^{n-1}(x+i)\right)\Gamma(x)$. The conditional is then

$$p_{\vartheta,\beta,\xi}\left(t_{d,e} = \hat{t} | \mathbf{t}^{\setminus(d,e)}, \mathbf{s}, \mathbf{f}\right) = \frac{\mathrm{DMC}\left(\mathbf{t}|\vartheta\right)}{\mathrm{DMC}\left(\mathbf{t}^{\setminus t_{d,e}}|\vartheta\right)} \times \frac{\mathrm{DMC}\left(\mathbf{s}|\xi\right)}{\mathrm{DMC}\left(\mathbf{s}^{\setminus t_{d,e}}|\xi\right)} \times \frac{\mathrm{DMC}\left(\mathbf{f}|\beta\right)}{\mathrm{DMC}\left(\mathbf{f}^{\setminus t_{d,e}}|\beta\right)}. \quad (7.6)$$

Substituting the value of each Dirichlet-multinomial compound, and applying the Gamma function expansion, yields a value proportional to

$$\overbrace{\left(c^{\setminus t_{d,e}}(d,\hat{t}) + \vartheta_{\hat{t}}\right)}^{\text{smoothed template usage}} \times \overbrace{\frac{c^{\setminus t_{d,e}}(\hat{t}, s_{d,e}) + \xi_{s_{d,e}}}{\sum_s c^{\setminus t_{d,e}}(\hat{t}, s) + \xi_s}}^{\text{smoothed template-specific slot frequency}} \times \overbrace{\frac{\prod_{f \in e}\left[\prod_{l=0}^{\#(f \in e)-1} c^{\setminus t_{d,e}}(\hat{t}, f) + \beta_f + l\right]}{\sum_f c^{\setminus t_{d,e}}(\hat{t}, f) + \beta_f}}^{\text{smoothed per-template frame frequencies}} \quad (7.7)$$

Here I have used the $\setminus t_{d,e}$ notation to indicate the assignment to $t_{d,e}$ removed from the given quantity. The slot sampling equation is analogous, as are the ones for the frame and role.

7.3.1 Implementation Considerations

In practice, the iterative multiplication in (7.7) will run into numerical issues if computed directly. Performing operations step-by-step in log-space is one straightforward solution, though at the cost of implementation efficiency. In initial pilot studies, I achieved up to a 40% speed-up within the sampling inner-loop by "directly" computing the log variant of (7.6). This involves computing $\log\Gamma(x)$, for which there are

numerous publicly available implementations. In the publicly available C++ implementation,[5] I use GSL. I demonstrate this speed-up in Figure 7.4, which shows the relative speed-up of computing $\log \frac{\Gamma(x+c)}{\Gamma(x)}$ directly, using GSL, as

```
#include <gsl/gsl_sf_gamma.h>
gsl_sf_lngamma(x+c) - gsl_sf_lngamma(x);
```

vs. as the reduced iterative sum

$$\sum_{i=0}^{c} \log(x + c - i).$$

The results in Figure 7.4 compute the log Gamma ratios 100 times over 500,000 sampled values x and c, with $2 < x \leq 1000$, and $2 < c \leq 25$. These values were chosen both to cover common values seen in development, and to study possible asymptotic behavior. Finally, note that all Gamma function arguments in (7.7) are integral, so $\log \Gamma(x) = \log(x-1)!$. I also experimented with using Sterling's approximation

$$\log n! \approx n \log n - n.$$

While this provided an additional speed-up (even against the GSL computation), it introduced errors, particularly as either x or c increased.

[5] `https://github.com/fmof/unified-probabilistic-frames`

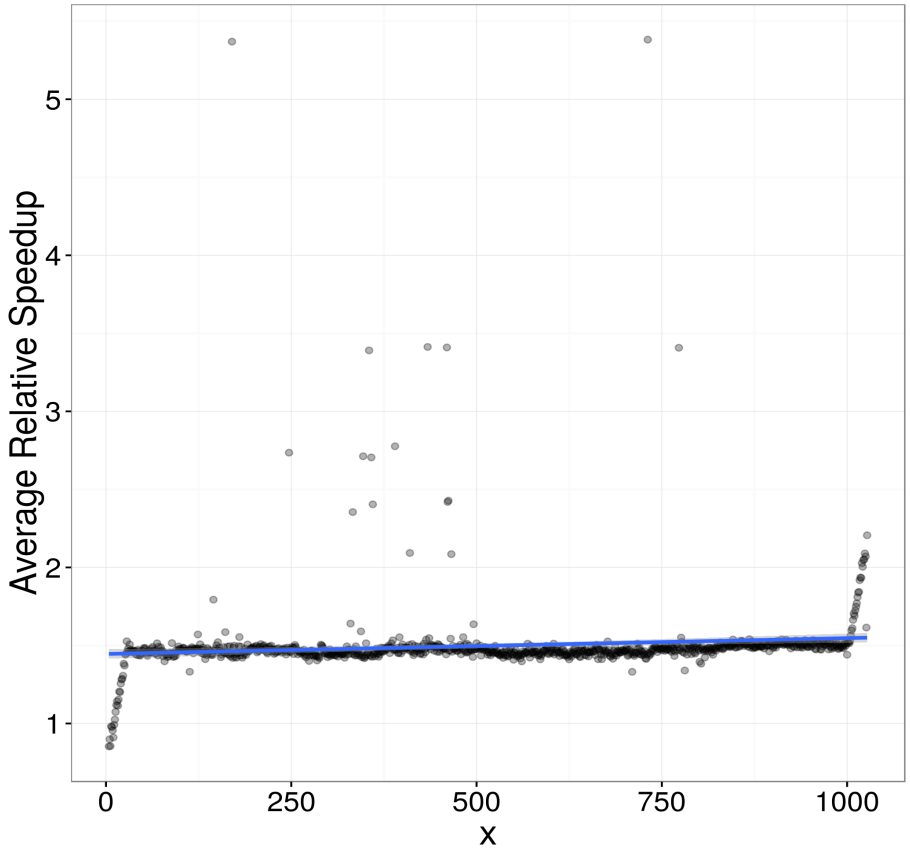


Figure 7.4: The relative speedup obtained computing $\log \frac{\Gamma(x+c)}{\Gamma(x)}$ using the scientific library GSL vs. a straightforward iterative sum.

7.4 Learning from Newswire

Minsky's, Schank's, and Fillmore's motivations were focused on matters of classic AI and cognitive science: the goal was to model human intuitions about everyday affairs (Minsky, 1974; Schank, 1975; Fillmore, 1975). In the following experiments, I address the question of how recent statistical approaches bear on the early proposals to discourse understanding, and consciously divorce the model from specific *downstream*

tasks. This division should not be taken to mean that the downstream tasks are not important or "bad." Rather, it is one way to distinguish the scientific questions representative potential and model expressiveness from the engineering questions of current downstream utility.

While I will return to the question of downstream use in chapter 8, I would first like to highlight some of the potential confounding factors of the downstream tasks. While various applications make use of the notion of an event template, such as MUC (Sundheim, 1992, 1996) and ACE (Walker et al., 2006), these tasks are defined by rather limited domains. It is not clear how well these tasks get at the more generalizable background knowledge of importance to the AI pioneers. First, those tasks' restricted domains mean the *evaluated* templates or relations are constrained not only by the domain, but also by the needs of the "target consumer," and what he or she deems to be "relevant." For instance, in MUC some events (*killing*) that would normally evoke a domain-relevant template (ATTACK) do not evoke any because the *killing* event involved specific types of entities deemed irrelevant to the consumer. Second, nearly 80% of MUC-4 only has one labeled template, despite an average of (at least) three templatable events in the text (Reichart and Barzilay, 2012). Third, subleties of evaluation can drastically affect the overall score and end ranking, introducing confounding variables into meta-analyses (Chambers, 2013, § 5). My goal in this chapter is to bring modern efforts in discourse and event modeling closer to Minsky's proposal; therefore the evaluations reflect these desiderata.

In the spirit of past efforts to learn general domain narrative schemas, I use 10,000 training and 1,000 held-out *New York Times* articles sampled uniformly at random from all years of Concretely Annotated Gigaword (Ferraro et al., 2014). Further, I note that, like many modern probabilistic models, that of §7.2 is not lightweight – though the concerted efforts of the past decade on optimizing topic models (Hoffman et al., 2012, i.a.) indicate the models can be made to scale; I address some of these issues in chapter 8. As general newswire, the NYT tends to be longer, contain more entities, and more diverse in how actions and participants are characterized than previous datasets used for unlabeled template induction (c.f., Chambers and Jurafsky, 2009; Cheung et al., 2013; Chambers, 2013; Bamman et al., 2013; Bamman and Smith, 2014).

I examine the effect of frame semantics on learned templates. Quantitatively, I ask if frame semantics result in better

1. model fit (heldout log-likelihood) and

2. semantic coherence (Mimno et al., 2011).

Within each of these, I further examine the impact that two aspects of the model have on these evaluations: (1) the impact of slots and how they are used, and (2) the impact of withholding surface semantic frame observations.

Name	semantic frame	typed semantic role	lexical predicate	typed syntactic dep.
Variable	f	r	v	a
Vocab. Size	642	2,515	2,522	24,696

Table 7.1: Statistics of the 10,000 training documents, after the preprocessing of §7.4.1. The "variable" row corresponds with those in Figure 7.2.

7.4.1 Pre-Processing

I extracted the CoreNLP (Manning et al., 2014) "collapsed cc" dependency parses and entity coreference chains, and SEMAFOR (Das et al., 2010, 2014) semantic frame parses, from Concretely Annotated Gigaword (Ferraro et al., 2014). While at the time of pre-processing SEMAFOR was a state-of-the-art FrameNet parsing system, its overall performance is still significantly lower than that of dependency parsing. To allay concerns about errant FrameNet annotations, I applied a high-precision filtering step: I only included an entity mention if

(1) the syntactic governor v of the mention's head word is a verb, or is part of an auxuliary or `xcomp` construction;

(2) its "verb" v triggers a frame f;

(3) r, one of f's frame roles, points to some span within the mention; and

(4) the mention was not contained within any other mention.

I qualitatively observed in development that these filters compensated for some of the gap in FrameNet and syntactic parsing, albeit by tying frames closely to syn-

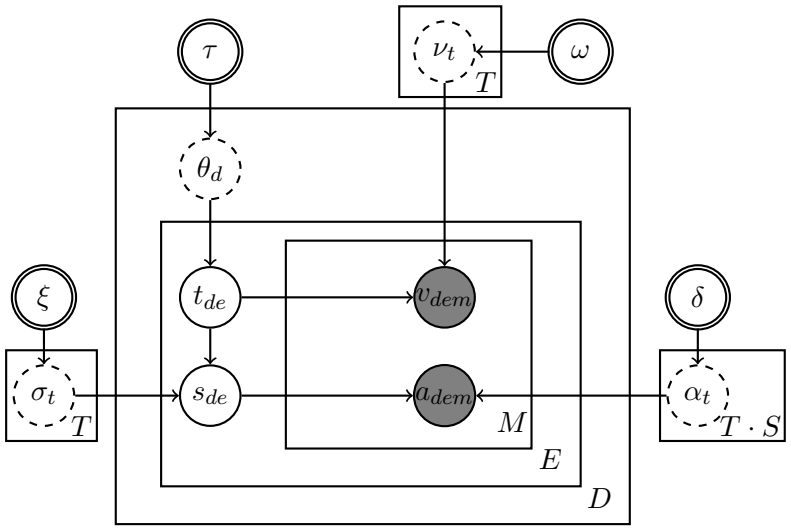


Figure 7.5: The baseline probabilistic frames model, using the same variable names and meanings as in Figure 7.2.

tax. Table 7.1 shows the number of type observations this preprocessing step, on the 10,000 training set, yielded. Comparing Table 7.1 to Table 5.1, the semantic frame (f) coverage is high against a much larger portion of the Concretely Annotated Corpora(roughly 80% coverage). However, the typed semantic roles, predicates and typed syntactic dependencies have much lower coverage.

7.4.2 Baseline

The baseline model, shown in Figure 7.5, is a simplification of our proposed model: it does not consider either frame or role information. This way, I can examine the effect of incorporating semantic frames in our unified model. Verbs are drawn directly from the template selection, and the arcs directly from the slots. That is, I directly

draw $v_{d,e,m} \sim \text{Cat}\left(\nu_{t_{d,e}}\right)$ and $a_{d,e,m} \sim \text{Cat}\left(\delta_{s_{d,e}}\right)$, resizing and reindexing the number of predicate and dependency distributions ν_t and δ_s as needed. I remove the discrete variables $f_{d,e,m}$ and $r_{d,e,m}$; the priors ϕ and ρ; and the hyperparameters β and ψ. I note that this is also one of Chambers (2013)'s models, and it can also be viewed as very similar to Bamman et al. (2013)'s generative model. The evaluation methodology—observing semantic frames only during training—provides a fair comparison between this baseline model and our own.

7.4.3 Quantitative Evaluation 1: Perplexity

I argue that evaluating perplexity (i.e., held-out log-likelihood) makes particular sense in the context of *surprisal* (Attneave, 1959; Hale, 2001; Levy and Jaeger, 2006; Levy, 2008). Used successfully to explain people's syntactic processing difficulties, the suprisal of a word w, given prior seen words $\mathbf{h}$ and "extra-sentential context" C (Levy, 2011) is as

$$\text{suprisal}\,(w|\mathbf{h}) \propto -\log p(w|\mathbf{h}, C). \tag{7.8}$$

Ignoring C yields a quantity proportional to held-out log-likelihood. Surprisal of an entire document d then follows the model's topology and factorization over d. Because my model and the baseline do not examine sequences of predicate/dependency pairs, the prior history $\mathbf{h}$ is removed from the computation. For this work, I effectively examine semantic and discourse approaches to expanding out this extra-sentential

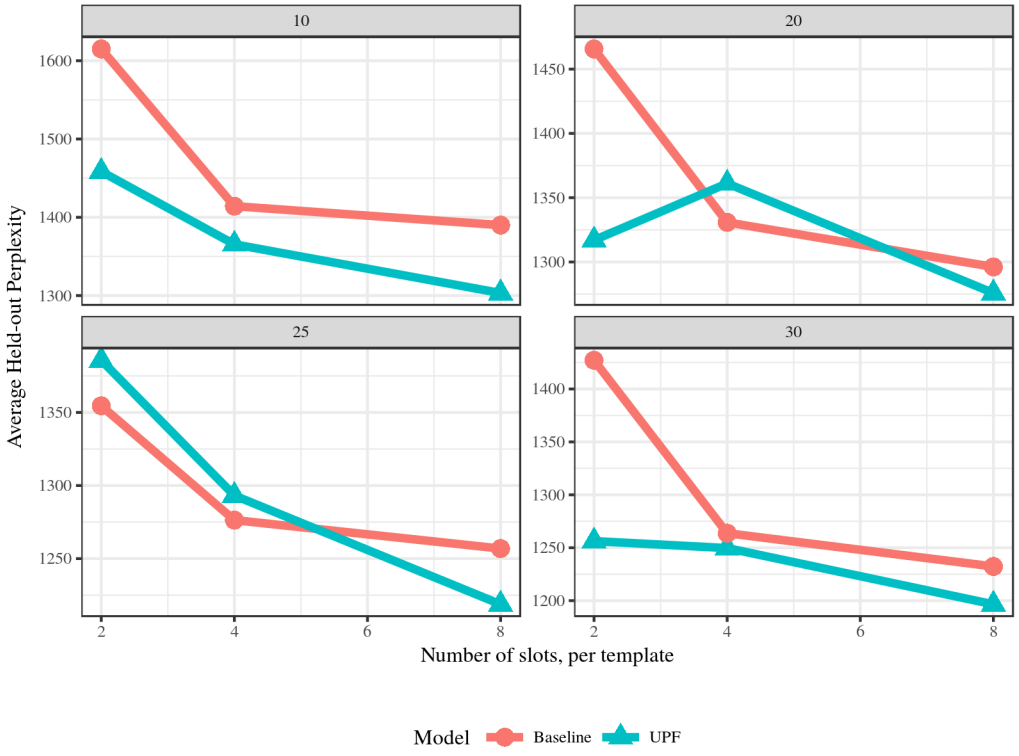


Figure 7.6: The held-out averaged perplexity of this chapter's model versus the baseline, with hyperparameters optimized.

context C, from within a bag-of-words view.

In Figure 7.6 I compare the average heldout perplexity on the 1,000 test documents run for 1,000 samples; the hyperparameters for these models are optimized. I treat frames/roles as latent during the heldout evalution of Figure 7.6, but as observed during training. Overall, the general trend is that the additional frame information allows the model to better fit held-out data, indicating a lower surprisal. In particular, increasing the number of model parameters tends to decrease perplexity.

Given sufficient training data, this in itself is not surprising. What is interesting

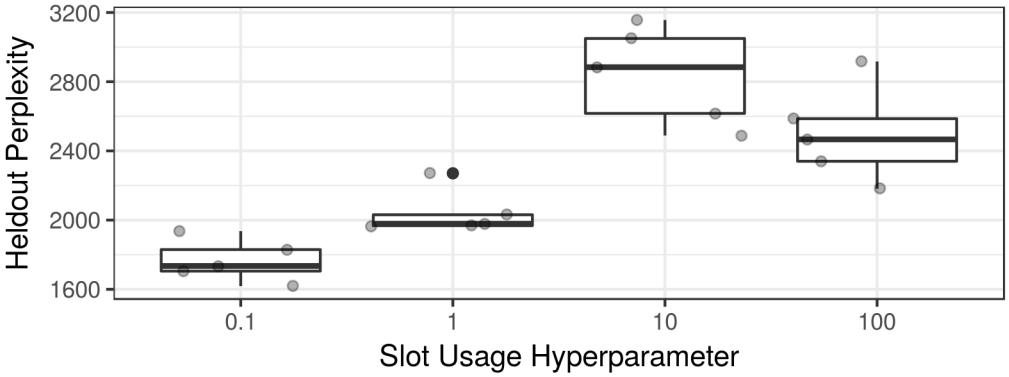


Figure 7.7: Heldout perplexity as a function of fixing the slot usage hyperparameter ξ. Each point represents a different training and evaluation run.

are the drivers of this perplexity decrease. While perplexity tends to decrease as the number of templates is increased, the most evident decreases come as the number of slots per template is increased. Recall that while the templates generate predicates and slot assignments, the slots are responsible for generating typed dependencies—and at both the semantic and syntactic layers, there are many more of these typed dependencies than predicates. Increasing the number of slot parameters can help control this larger vocabulary.

Slot Usage and Perplexity

The models in Figure 7.6 were learned with optimized hyperparameters. In Figure 7.7 I present perplexity results where all hyperparameters are fixed and I vary the values of the slot usage hyperparameters ξ from 0.1 to 100.[6] These hyperparameters

[6]All other hyperparameters are set to 0.1 to encourage peakier distributions. Note that *none* of the hyperparameter values were optimized in this set of experiments.

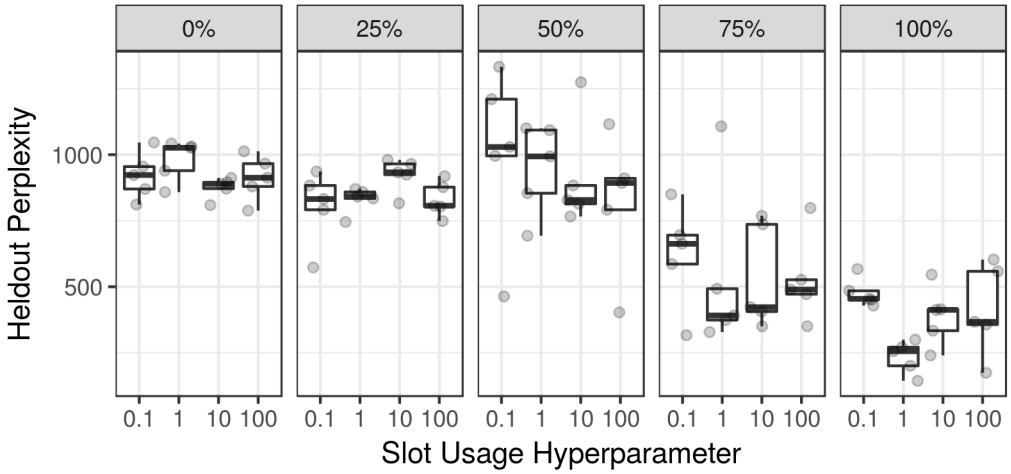


Figure 7.8: Heldout perplexity as a function of the proportion of documents for which surface semantic frames were unobserved, as the slot hyperparameter varies. Each point represents a template's verb coherence, marginalizing out semantic frames, for a different training and evaluation run. Models with "0%" observed all semantic frames; models with "100%" observed none (sampling them).

control how slots are used within each template. These models are trained and evaluated on the same 10,000 and 1,000 documents as above, but because computing the marginalized perplexity is computationally expensive I only perform 250 heldout sample iterations. This allows a greater number of (noisier) evaluations to be computed in parallel. All models were trained with 20 templates and 8 slots per template.[7] Although lower ξ (peakier σ_t) tends to be better, the high, uniform-inducing ξ value of 100 often outperforms the lower value of 10.

[7]The community has not settled on the expected number of slots to learn: MUC primarily uses four, Reichart and Barzilay (2012) use eight on a very small set of NYT articles, and Balasubramanian et al. (2013) use up to fifteen actors. Given these, the strongest results from Figure 7.6, and the overall computational requirements, I decided to use 20 templates with 8 slots a piece.

Surface Semantics and Perplexity

In Figure 7.8 I examine how inferring varying amounts of surface semantic labels, both frames and roles, affects perplexity. Specifically, each facet shows *semantic dropout* rate—the percent of documents that had semantic *labels* hidden. Thus models trained with 0% semantic dropout observed all semantic labels, while models trained with 100% dropout observed no labels. As above, all models were trained with 20 templates and 8 slots per template. Note that the predicate and hierarchical template and observation structures were not affected: the model still had 20 templates and 160 slots in total to learn, and neither entities nor mentions were removed when semantic information was hidden.

Given the role that slots and their usage plays, I couple this semantic dropout experiment with different values of ξ. Because some proportion of semantic frames and roles must be inferred during training, I significantly decrease the size of the training set, from 10,000 to 250, as well as the number of training samples; I still evaluate on the same 1,000 heldout documents.[8] Each training instance uniformly samples its 250 documents from the 10,000 training collection.

We see a general trend that as more semantic information is occluded perplexity decreases. In particular, models that inferred all semantic labels routinely halved, at a minimum, the perplexity of models that observed even up to 50% of semantic labels. This can be interpreted as having fewer constraints for the same number of

[8]The models with full dropout (100%) were more than 60 times slower to train than those with no dropout (0%).

parameters: the (sampled) semantics can be redirected to act as needed between the templates (the thematic layer) and the observed syntactic layer. Notice that for high dropout models, perplexity is both less variable and less dependent on slot usage parametrization.

7.4.4 Quantitative Evaluation 2: Coherence

Chang et al. (2009) showed that improvements in topic model held-out log-likelihood do not always correlate with human quality scores. In response, Mimno et al. (2011) developed an automatic *coherence* measure that does correlate (positively) with human quality scores.[9]

Despite being developed for topic models, there is nothing inherent in its definition that limits its application to just topic models. Given a list of vocabulary words X, sorted by weight (probability), the coherence score measures the log-relative document frequencies of the M-highest probability elements of X:

$$\text{coherence}(X, M) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D(X_{(m)}, X_{(l)}) + 1}{D(X_{(l)})},$$

where $D(\cdot)$ is the number of documents that have at least one occurrence of each of its arguments. The intuition behind coherence is a modified distributional hypothesis: topics composed of co-occurring words are likely to be "better" than those composed

[9]Other researchers also proposed alternatives (Aletras and Stevenson, 2013; Newman et al., 2010; Lau et al., 2011).

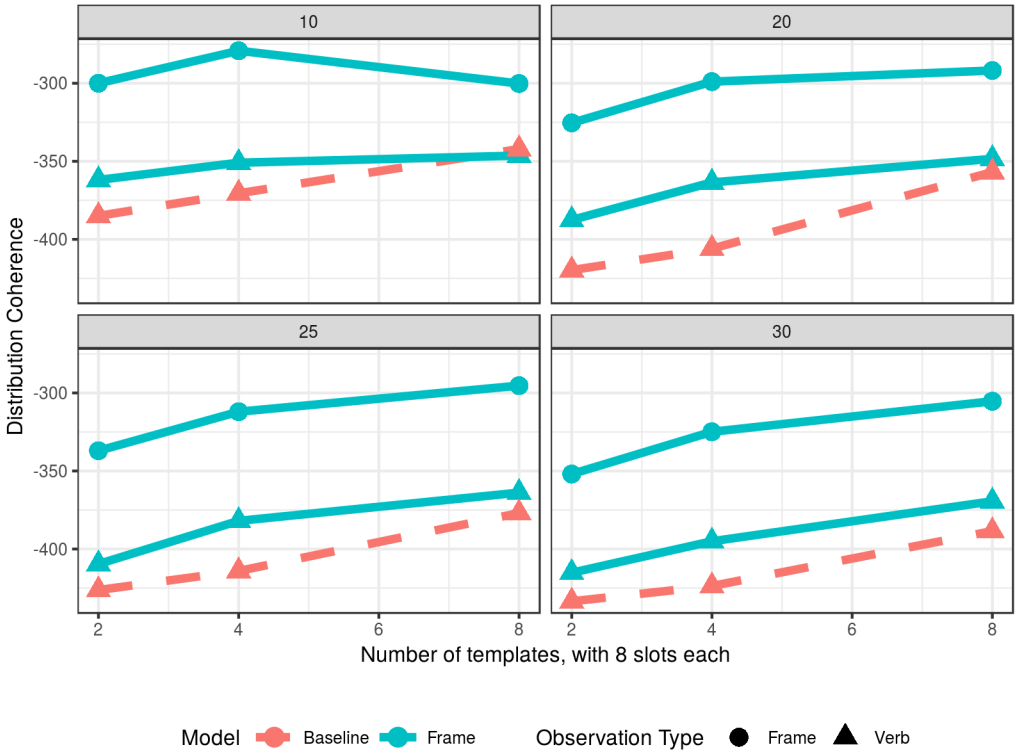


Figure 7.9: Topic coherence at $M = 20$. For the unified model, I also provide two measures of coherences per template: one at the frame level and one given a template. The latter marginalizes over frames. Hyperparameters are optimized. Higher is better (more coherent).

of randomly occurring words. Lau et al. (2014) find that this coherence measure is a competitive automatic evaluation that tends to reflect overall model quality. I adopt this measure, as the models examined here produce distributions over predicates, frames, and other observations.

Therefore, the second evaluation is Mimno et al.'s topic coherence, evaluated at the syntactic frame level, i.e., against observable verbs. To compute verb coherence in the UPF model, I marginalize over frame (and role) assignments. In Figure 7.9 I

show coherence at top 20, when model hyperparameters are optimized. The unified, semantic-marginalized model (blue triangles) generally results in higher coherence than the baseline (red triangles), even as the number of templates and slots varies. (While not fully comparable, I also show the pure semantic frame coherence—the blue circles—of the unified frames model as a point of comparison.)

In contrast to the hyperparameter optimized perplexity results in Figure 7.6, the verb coherence is less variable and follows clearer trends: increasing the number of slots per template improves coherence, but increasing the number of templates decreases coherence. The inverse relationship between the number of templates and coherence may seem counter-intuitive, but it follows what has been observed previously with coherence (Mimno et al., 2011): coherence is a measure relative to the parametrization, which must be controlled for.[10] While it is inappropriate to compare coherences as the number of templates change (across the facets of Figure 7.9), it *is* appropriate to compare coherences as the number of slots per template change (within the facets).

Slot Usage and Coherence

In Figure 7.10 I examine how inferring varying amounts of surface semantic labels, both frames and roles, affects the verb coherence (semantic frames marginalized out). Each point in this figure represents the coherence for a particular template. As before,

[10]Coherence evaluates the top M words per topic, for a fixed value of M. Using more topics means that each one can be more specialized, resulting in lower entropy distributions.

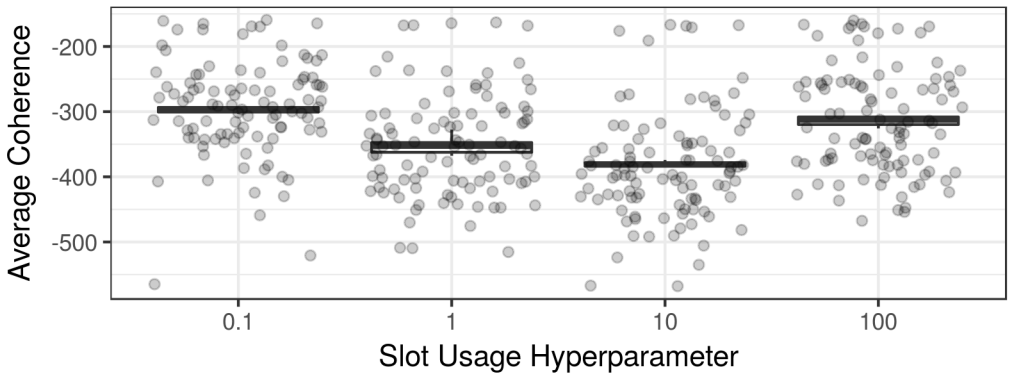


Figure 7.10: Template-verb coherence as a function of fixing the slot usage hyperparameter ξ. Each point represents a different training and evaluation run.

all models were trained with 20 templates and 8 slots per template.

There is the same non-direct relationship observed before, but now between ξ and coherence: very lower ξ (0.1) results in higher coherence, but it is matched by very high ξ (100). Unlike perplexity, though, the hyperparameter optimized coherence of Figure 7.9 is outperformed by the fixed hyperparameter coherences here. This reflects the fact that the hyperparameters optimize the log evidence (Wallach, 2008), and that what optimizes likelihood does not necessarily optimize other metrics.

Surface Semantics and Coherence

In Figure 7.11 I examine how inferring varying amounts of surface semantic labels, both frames and roles, affects verb coherence. As when examing surface semantic dropout before, each facet shows *semantic dropout* rate and all models were trained with 20 templates and 8 slots per template. I also use the same training subsampling:

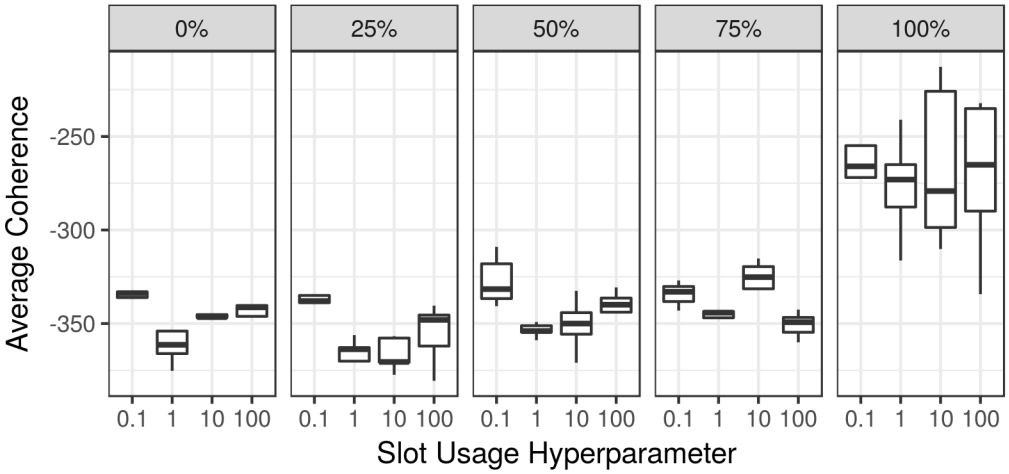


Figure 7.11: Template-verb coherence as a function of the proportion of documents for which surface semantic frames were unobserved, as the slot hyperparameter varies. Each point represents a template's verb coherence, marginalizing out semantic frames, for a different training and evaluation run. Models with "0%" observed all semantic frames; models with "100%" observed none (sampling them).

each model uniformly trains on 250 uniformly sampled documents from the 10,000 training collection, but evaluation occurs on the 10,000 set. Thus, the results in Figure 7.11 are comparable to those in Figure 7.10.

Overall, the semantic dropout rate does not depend highly on the semantic dropout rate—up until *all* semantic labels must be inferred. At that point, coherence is vastly improved, though at greater variability. Across all dropout rates, the general pattern observed in Figure 7.10—the best coherences result from very low and very high ξ—is evident here too.

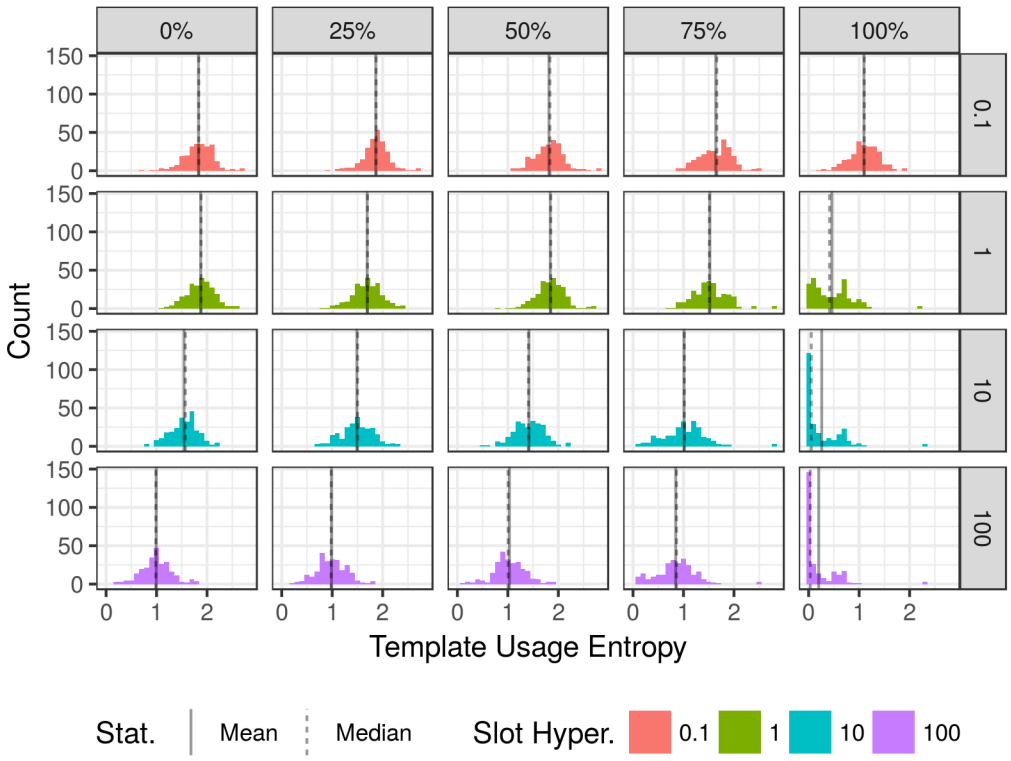


Figure 7.12: The inferred template usage entropy of non hyperparameter optimized template models with 20 templates 8 slots, varying semantic frame dropout and the value of the slot usage hyperparameter.

7.4.5 Qualitative Exploration

In the previous section I explored how varying how slots are used and how many surface semantic frame forms are observed affects both perplexity and template-verb coherence. In these experiments we saw that giving the model as much leeway as possible regarding the surface semantic layer, i.e., using a dropout rate of 100%, (1) resulted in the lowest perplexities, (2) yielded the highest coherences, and (3) varied the least. In this section I explore these results qualitatively.

Analyzing Template Usage

First, consider Figure 7.12, which illustrates the distributions of entropy of the learned per-document template proportions θ_d, across different slot hyperparameter values (down, and colored) and semantic frame dropout rates (across). The average and median entropies are shown as well. Going from the upper left (all semantic labels and peaky slot priors) to the lower right (no semantic labels and more uniform slot priors), we see that the entropy consistently decreases. The entropy tends to be more sensitive to the slot hyperparameter than the percent of observed semantic frame labels; the previous ablation results (Figs. 7.8 and 7.11) display this pattern too.

The severely decreasing entropies of the fully occluded semantic label models help explain the increasing variability seen in Figure 7.11. Namely, each document prefers to use fewer and fewer templates: templates are therefore more likely to represent

co-occurring words more homogeneously.

Learning the Syntax-Semantics Interface

In Table 7.2 I examine how semantic frame occlusion affects ν, the learned syntactic-semantic interface for predicates. For this, I include four extra dropout levels at 90%, 92.5%, 95% and 99%. Note that the 0% dropped column represents a smoothed maximum likelihood estimate. There are two primary observations from the table: first, that each semantic frame's MLE distribution has low enough entropy that the ability to reconstruct the MLE is fairly high—even when having to infer 90% of semantic frame labels. Reconstructions at 95% are somewhat accurate; however, the reconstructed distributions at 99% and 100% are very poor. This suggests that, if computation were not a concern, one could use significantly fewer, but not no, frame annotations, successfully reconstructing the empirical semantic frame distribution while yielding improved perplexity and coherence. Second, notice that a common problem in unsupervised learning occurs: while the 99% and 100% dropout distributions do reflect similarities in syntactic predicates, they are divorced from the actual frame labels.[11]

0% dropped	25% dropped	50% dropped	75% dropped	90% dropped	92.5% dropped	95% dropped	99% dropped	100% dropped
				Resolve Problem				
agree (0.896)	agree (0.883)	agree (0.736)	throw (0.095)	agree (0.1)	move (0.08)	agree (0.019)	go (0.012)	say (0.014)
relent (0.05)	relent (0.046)	relent (0.067)	charge (0.095)	light (0.05)	increase (0.041)	make (0.01)	make (0.012)	go (0.005)
take (0.000318036)	take (0.000352968)	work (0.001)	remain (0.095)	dry (0.05)	return (0.04)	get (0.006)	require (0.011)	violate (0.004)
make (0.000276938)	call (0.000349337)	take (0.001)	tend (0.094)	take (0.006)	deny (0.04)	take (0.006)	use (0.009)	turn (0.004)
call (0.000236526)	make (0.000307347)	cover (0.000998673)	spot (0.094)	make (0.005)	import (0.04)	increase (0.005)	know (0.007)	think (0.004)
turn (0.000195986)	get (0.000305046)	call (0.000934042)	negotiate (0.047)	require (0.004)	agree (0.04)	bring (0.005)	take (0.006)	work (0.004)
find (0.000195774)	work (0.000259992)	hold (0.000841811)	shrink (0.047)	turn (0.004)	horrify (0.039)	feel (0.004)	think (0.006)	know (0.004)
get (0.000157382)	find (0.000259673)	allow (0.000840108)	pack (0.047)	describe (0.004)	call (0.004)	head (0.004)	begin (0.005)	include (0.004)
look (0.00015704)	cover (0.000258685)	need (0.000759234)	proofread (0.047)	consider (0.003)	enter (0.004)	call (0.004)	want (0.005)	sign (0.004)
work (0.000156769)	fill (0.000258327)	turn (0.000757683)	frame (0.047)	believe (0.003)	require (0.004)	reflect (0.004)	show (0.005)	bring (0.003)
				Cause Harm				
hurt (0.232)	hurt (0.237)	hurt (0.254)	discuss (0.142)	hurt (0.135)	cross (0.052)	hurt (0.043)	go (0.012)	say (0.026)
stab (0.116)	stab (0.104)	stab (0.127)	knock (0.142)	reduce (0.109)	hurt (0.052)	visit (0.029)	make (0.012)	blame (0.018)
knock (0.101)	knock (0.089)	beat (0.127)	hurt (0.102)	take (0.084)	call (0.005)	play (0.017)	use (0.009)	anger (0.018)
beat (0.087)	beat (0.074)	strike (0.048)	stab (0.081)	strike (0.081)	enter (0.005)	view (0.016)	know (0.007)	make (0.015)
strike (0.072)	strike (0.074)	burn (0.048)	leap (0.081)	establish (0.054)	require (0.005)	strike (0.015)	take (0.006)	cross (0.014)
wound (0.058)	wound (0.059)	please (0.048)	attend (0.041)	cover (0.028)	make (0.005)	explain (0.015)	think (0.006)	count (0.014)
burn (0.058)	burn (0.059)	punch (0.048)	climb (0.041)	convert (0.028)	take (0.004)	cross (0.015)	begin (0.005)	name (0.014)
injure (0.043)	bruise (0.044)	bruise (0.048)	strike (0.041)	ascend (0.027)	hold (0.004)	obtain (0.015)	want (0.005)	challenge (0.014)
bruise (0.043)	punch (0.044)	knock (0.032)	wound (0.041)	ruin (0.027)	go (0.004)	decorate (0.015)	show (0.005)	talk (0.01)
punch (0.043)	injure (0.044)	injure (0.032)	suffer (0.041)	make (0.003)	allow (0.004)	injure (0.015)	lead (0.005)	watch (0.01)
				Make Agreement on Action				
agree (0.896)	agree (0.883)	agree (0.736)	throw (0.095)	agree (0.1)	move (0.08)	agree (0.019)	go (0.012)	say (0.014)
relent (0.05)	relent (0.046)	relent (0.067)	charge (0.095)	light (0.05)	increase (0.041)	make (0.01)	make (0.012)	go (0.005)
take (0.000318036)	take (0.000352968)	work (0.001)	remain (0.095)	dry (0.05)	return (0.04)	get (0.006)	require (0.011)	violate (0.004)
make (0.000276938)	call (0.000349337)	take (0.001)	tend (0.094)	take (0.006)	deny (0.04)	take (0.006)	use (0.009)	turn (0.004)
call (0.000236526)	make (0.000307347)	cover (0.000998673)	spot (0.094)	make (0.005)	import (0.04)	increase (0.005)	know (0.007)	think (0.004)
turn (0.000195986)	get (0.000305046)	call (0.000934042)	negotiate (0.047)	require (0.004)	agree (0.04)	bring (0.005)	take (0.006)	work (0.004)
find (0.000195774)	work (0.000259992)	hold (0.000841811)	shrink (0.047)	turn (0.004)	horrify (0.039)	feel (0.004)	think (0.006)	know (0.004)
get (0.000157382)	find (0.000259673)	allow (0.000840108)	pack (0.047)	describe (0.004)	call (0.004)	head (0.004)	begin (0.005)	include (0.004)
look (0.00015704)	cover (0.000258685)	need (0.000759234)	proofread (0.047)	consider (0.003)	enter (0.004)	call (0.004)	want (0.005)	sign (0.004)
work (0.000156769)	fill (0.000258327)	turn (0.000757683)	frame (0.047)	believe (0.003)	require (0.004)	reflect (0.004)	show (0.005)	bring (0.003)

Table 7.2: Learned semantic-to-syntactic frame distributions for five different semantic frame dropout rates. The model is a 20 template, 8 slot-per-template model with slot hyperparameters of 0.1.

Executive Decisions	*Negotiation Ending*	*Heated Negotation*
Leadership	Statement	Perception experience
Operating a system	Causation	Self motion
Activity start	Respond to proposal	Arriving
Becoming a member	Arriving	Evidence
Statement	Verification	Residence
Traversing	Judgment communication	Appearance
Causation	Cause to end	Grasp
Commerce pay	Intentionally act	Telling
Possession	Releasing	Death
Chatting	Cause to start	Emotion directed

[North Korea]$_1$ [restored]$_1$ regular border crossings for traffic going to South Korean factories in the North on Tuesday, while [its leader, Kim Jong Il]$_2$, [reiterated]$_{2,\ 3}$ his government's call for a peace treaty with [the United States]$_3$.

"We can ease tensions and remove the danger of war on the peninsula when the [United States]$_3$ [abandons]$_3$ its hostile policy and signs a peace treaty with us," Kim said in a commentary carried on Pyongyang Radio, which broadcasts North Korean government statements abroad.

Meanwhile, on Tuesday, [North Korea]$_1$ [restored]$_1$ regular traffic for South Korean companies that have operations in a joint industrial park in the North Korean border city of Kaesong. [The North]$_4$ had sharply [curtailed]$_4$ such traffic in December.

[Ian Kelly , a State Department spokesman]$_5$, [said]$_5$ Monday that Washington was " encouraged " by the North 's recent gestures toward the South , but [he]$_5$ [said]$_5$ [he]$_5$ [had]$_5$ no comment on the North 's call for a peace treaty .

[Kelly]$_5$ [urged]$_5$ North Korea to [return]$_6$ to [six-nation talks]$_6$ with regional powers about the dismantling of its nuclear weapons programs . The North , which prefers a bilateral dialogue with the United States , has said the six-party framework is dead .

Washington has said that [negotiating]$_4$ a peace treaty with [the North]$_4$ is possible only as part of a broader process that addresses the North 's nuclear disarmament . [North Korea]$_1$ [conducted]$_1$ its second nuclear test in May , and there is a growing suspicion among analysts in Seoul that [the North]$_4$ is trying to win diplomatic recognition from Washington while also being [accepted]$_7$ as [a nuclear power]$_7$.

Figure 7.13: Example output from a 20 template, 8 slot per template UPF model. I labeled the three templates (semantic frame distributions shown).

7.5 Discussion and Additional Challenges

In this chapter, I have presented a model for probabilistic frame induction. This model is the first to explicitly capture all levels laid out by Minsky (1974). In so doing I have combined the notion of Fillmore's frame semantics with a discourse-level notion of a Minsky frame, or Schankian script. I have shown that this leads to improved topic coherence and, overall, a better explanation of held-out data.

I have also explored some issues regarding the model's parametrization and ability to cope with missing semantic frames. Even with a large portion of semantic frame labels hidden, the model was still able to reconstruct the syntactic-semantic interface. While fully observing all semantic frames did lead to perplexity and coherence improvements, allowing more of those parameters to be optimized automatically produced much larger improvements. Moreover, hyperparametrization mattered some, with the model prefering peakier (low entropy) and flatter (higher entropy) distributions over how to use some of the hiearachical latent variables.

Significant challenges remain. While *learning* unified models with fully observed semantic levels have reasonable computational requirements, those requirements quickly become onerous as semantics are withheld. *Evaluating* these models is also very computationally expensive, since the evaluation (at least to compare against a Chambers (2013)-style baseline) is equivalent to testing at the 100% dropout level.

[11]The top ten items and weights for the `Possession` and `Cause Harm` 99% and 100% dropped columns are not typos: they are indeed the same.

More efficient inference techniques need to be explored.[12]

This chapter relied on a pipeline of previous NLP tools, making the model subject to propagated pipeline errors. To see this, consider some of the coreference errors in Figure 7.13, which shows a partially-labeled document: "the North" (entity 4) is not properly merged with "North Korea" (entity 1). This coreference error results in two different entities, with two different template (and slot) assignments.

Future efforts may wish to consider imposing additional syntactic constraints on the template and slot assignments. although there is not a requirement that syntactic arguments of the same verb are assigned the same template. Notice that while there is not this requirement currently, it can happen organically: both "its leader, Kim Jong Il" (entity 2) and "the United States" (entity 3) are arguments of "reiterate;" correctly, as separate entities, they can be assigned different templates. Notice though that here they are assigned to the same template.

While not considered here due to scope, alternative data present additional testbeds for research. Movie (Bamman et al., 2013) or book (Bamman and Smith, 2013) summaries may use multiple, partial or repeated templates to tell an involved story. Weblogs (Burton et al., 2009) represent a wealth of personal narratives.

Parallel template models are also an intriguing area of future work. Local newspapers contract through newswire services, running those articles verbatim or somewhat modified derivative articles. Alternatively, because movies are often based off

[12]Of course, deployed systems using these models may wish to withhold semantic frames for an entirely different reason: obtaining the semantic frames can itself be a time consuming chore.

of books, they form a type of pseudo-parallel corpora. Thus, much like a polylingual topic model analyzes two (translated, potentially paraphrastic) sources of input, parallel template models could analyze linked, different reportings of the same event. This is a particularly interesting notion for summarized narratives (Huang et al., 2016b).

Having shown the feasibility of inducing a unified representation of the language found in documents, motivated by historical AI accounts, in the next chapter I consider a modified approach that is motivated more by the real world style tasks set aside in this chapter.

Chapter 8

Semi-Supervised Featurized Event Templates

In Chapter 5, I presented an unstructured, conditional method for attributive lexical semantics that aggregates multiple semantic frame analyses from large corpora; while this provided rich word representations, representing super-lexical structural information is not trivial. Meanwhile in Chapter 7, I presented a structured, generative method for document modeling that strategically used semantic frame analyses; while effective, this method is computationally expensive, particularly when not every semantic frame is observed. The goal of this chapter, then, is to provide a method for scalable, structured event inference that can easily incorporate noisy, possibly missing, semantic frames.

When presented with a collection of text documents, users may be interested in

discovering overarching themes within the collection, effectively asking, "what's in this collection?" Or perhaps we have some *type* of documents in mind—those about basketball, politics or finance—and want to find similar documents. Therefore, in addition to the more intrinsic document understanding metrics of Chapter 7, I will examine how these scalable event learners aide extrinsic, human-oriented understanding of the overarching themes withing documents, looking specifically at document classification.

8.1 Adding Signal to Bayesian Models

Imagine a collection of documents that have been clustered into groups of like documents—documents reporting on basketball games or basketball players are clustered into a "basketball" group, while documents reporting on market volatility are clustered into a "finance" group. We would say that these documents have been *labeled* with their group's name. With access to a sufficient number of documents of interest that were already labeled, we could build a *document classifier*—an automated system to classify new, unseen documents with one of our known labels.[1] However, to get any labeled documents, annotators must start somewhere; depending on the complexity of the documents, the complexity of the labels, and any time or cost constraints for the annotation process, labeling a sufficient number of documents

[1] A document classifier represents a type of information extraction system: the label for a document typically corresponds to its core, or central, elements and story.

could present a significant hurdle.

While classifiers should be accurate, it would be ideal if the annotation process could reprioritize user-time and leverage a potentially massive amount of *unlabeled* data. One way to handle all of this unlabeled data is to induce a compact representation of the corpus. Topic models, such as Latent Dirichlet Allocation (Blei et al., 2003, Example 2.1 this thesis), have repeatedly shown an ability to induce approachable representations: their output is often interpretable, with "similar" (often thematically-related) words grouping together (Chang et al., 2009; Mimno et al., 2011).[2] Further, it has been shown that learning LDA models can scale easily (Hoffman et al., 2013). While these attributes position topic models to encourage thematic exploration, there is an observed tradeoff between document classification and topic discovery (May et al., 2015).

Researchers have also found LDA to be an effective building block or starting point for other models that can be used in downstream systems. The one that I will consider and extend in this chapter is the Dirichlet Multinomial Regression (DMR) topic model (Mimno and McCallum, 2008). The DMR conditions the per-document topic proportion draw $\theta_d \in \Delta^{(K-1)}$ on a weighting, generally log-linear, of F arbitrary features. Specifically, where as in LDA $\theta_d \sim \text{Dir}(\alpha)$, where α are *global* hyperparameters, the DMR topic model uses the interpolation weights $\delta \in \mathbb{K} \times \mathbb{F}$ and document

[2]In the terminology of Example 2.1, the grouped thematic words would all have some of the highest mass for some topic ψ_k. This topic might then be recognizable as a topic representing that theme. For example, if there were a topic, some of whose top weighted words included "basketball," "court," and "buzzer," then that topic might become known as the "basketball" topic. A similar type of reasoning can be seen through chapter 7, in particular in Table 7.2 and the top of Figure 7.13.

features $y_d \in \mathbb{R}^F$, following

$$\theta_d \sim \text{Dir}\left(\alpha_d\right)$$

$$\alpha_d = \alpha \odot \exp \delta y_d,$$

where $x \odot y$ represents the Hadamard (point product) of vectors x and y. A similar approach has been used by Paul (2015).

The DMR still uses global hyperparameters α as a general guide of what topics are overall more likely. By using (observed) features y_d to construct α_d, the DMR leverages extra information in order to fine-tune the prior beliefs on what topics are likely to be in each document. Because the model conditions on the features, we do not need a proper (or any) generative story for them.

Ramage et al. (2009)'s Labeled LDA, like Mimno and McCallum (2008)'s DMR, treated y_d as a conditioned observable. This model considered classification response items and induced topics to be one-to-one. This observed variable obtained using these labels directly perturbe the usage proportions, only allowing that document to use the topics associated with its labels. During evaluation, all topics can be used.

Some efforts, the Mimno and McCallum's DMR and Ramage et al. among them, have relied on the posterior topic usage $p\left(\theta \mid \{\mathbf{x}_d\}\right)$ carrying enough signal for a post-hoc classifier (Eisenstein et al., 2011; May et al., 2015).

A straight-forward approach is to model a document's label y_d generatively

through a generalized linear model (GLM). Both McAuliffe and Blei (2008) and Chen et al. (2015) do this; McAuliffe and Blei model y_d according to all of the topic assignment choices $z_{d,n}$, while Chen et al. model y_d according to the topic usage parameter θ_d. Neither of these models (as written) allow external, conditional features to influence the classification decision.

While Chen et al. (2015)'s modeling choices can be an appropriate way to construct document classification systems: it embodies the post-hoc classification approaches, while still being fully generative. Yet, it is not obvious where or how to include arbitrary features, and the effect of this classification module on inference. If a generalized linear model is used as the generative classifier, this poses an issue for Bayesian inference, as GLMs and Dirichlet distributions are not conjugate.

Moreover, note that while the *model* they present is generative, they actually employ a discriminative one (Minka, 2005). Specifically, this new model requires learning an additional set of topic parameters, potentially presenting robustness issues when labeled data are scarce.[3] Likely as a result of steps Chen et al. needed to take to perform inference, I found in my own experiments with their model that it was difficult to find patterns in the learned topics: they lost interpretability.

[3]The number of additional parameters grows multiplicatively in the number of topics and the size of the vocabulary (roughly, $K \times V$). While a standard benchmark academic topic model may use 100 topics for a 10,000 word vocabulary, amounting to an additional one million parameters to learn, learning a topic model over 500,000 words (as one may get from a web corpus) requires learning 50 million additional parameters.

8.2 A Conditionally Generative Model of Discourse

As discussed above, DMR has been an effective method of combining supervisory features in ways that can guide inference in the model, impacting both introspective and downstream results. However, an issue arises when the features may be too expensive to obtain, too errorprone to obtain, or too targeted toward a specific type of document. These complicating factors are common elements of standard supervised machine learning, but I argue that these complications are nuanced. First, the complications can be soft ones: perhaps the automatically obtained features have large compute requirements. So the features may be useful but have diminishing downstream utility, or have a disproportionate *relative* cost. Second, they can be hard complications: when requiring human intervention or labeling, the features may simply be too costly in terms of time or money, neither of which may exist. Similarly, it can be difficult for humans and automated system to provide, either consistently or at all, accurate annotations.

In this section, I describe a DMR-inspired event model that accounts for noisy or missing features, that I call **bpDMR-Events**—or backpropagation through DMR for Events. This method augments DMR with a generic, generative story. When the features are observed, the generic prior captures a lightly-regularized maximum likelihood estimation of the overall strengths of the features. When features are un-

observed, this prior provides the sufficient statistics for obtaining initial estimates of the features. In both cases, it allows the features to be rescaled, such as to the range $[0, 1]$—a standard classification preprocessing procedure. Employing a Gumbel softmax reparametrization (Jang et al., 2017; Maddison et al., 2016), this model handles the rescaling in a principled manner, without needing to specify complete conditional distributions of the features (as would be needed for sampling or variational inference).

8.2.1 Generative Story

The bp-DMR model has two components: an observation component and a feature component. Like the DMR topic model, the feature component conditionally informs the observation component.

The Observation Component

The observation component builds off of the baseline model of chapter 7, but with one important change. An event template is still a distribution over predicates and a distribution of slots, where the latter reference distributions over roles/relations. But here I model the slots as being globally shared, rather than unique to particular templates. This is inline with others' modeling decisions (Nguyen et al., 2015) and evaluation (Cheung et al., 2013).

The observations and latent assignments are discrete and I place conjugate Dirich-

let priors on each. The narrative frame of a document d is represented as a mixture over the set of templates T (Minsky's thematic frames):

$$\tau_d \sim \text{Dir}\,(\vartheta).$$

Each template t, such as representing NEGOTATION, is represented by a distribution σ_t over S *shared* slots, such as the NEGOTIATOR, and a distribution ν_t over V types of observed syntactic (i.e., lexical) predicates

$$\sigma_t \sim \text{Dir}\,(\xi),\ \phi_t \sim \text{Dir}\,(\beta).$$

Each slot s has a distribution ρ_s over R typed syntactic dependencies:

$$\rho_s \sim \text{Dir}\,(\phi).$$

An entity e is assigned to a single template $t_{d,e}$ and slot $s_{d,e}$, where:

$$t_{d,e} \sim \text{Cat}\,(\tau_d), \text{ and } s_{d,e} \sim \text{Cat}\left(\sigma_{t_{d,e}}\right).$$

For every mention m of e, the entity template $t_{d,e}$ directly accounts for the mention's governing predicate $v_{d,e,m}$:

$$v_{d,e,m} \sim \text{Cat}\left(\nu_{t_{d,e}}\right),$$

and the slot $s_{d,e}$ accounts for the mention's syntactic dependency (how it is used in the syntactic frame):

$$r_{d,e,m} \sim \text{Cat}\left(\rho_{s_{d,e}}\right).$$

The Feature Component

The feature component builds off of the DMR: assume some feature representation y_d of a document defined over F features. These features will directly influence $\tau_d \in$

Δ^{T-1}, how the T templates are used in the document, via a non-linear interpolation. Specifically, the features $y_d \in \mathbb{R}^F$ will be interpolated with $\delta \in \mathbb{R}^{T \times F}$ as $\delta y_d \in \mathbb{R}^T$, and then passed through a differentiable non-linear function $f : \mathbb{R}^T \to \mathbb{R}^T$. The result of the non-linearity will be multiplied component-wise with the global ϑ hyperparameters, to get ϑ_d, a document-specific parametrization for the template proportion's Dirichlet prior. Aside from generalizing the exponential function to a differentiable non-linearity, this is the DMR specification.

However, that assumes that every document has features y_d. What happens when documents are missing features, i.e., the features are fully unobserved? For instance, one could consider treating any human-provided labels as features, in effect as a generalized and soft version of Ramage et al. (2009); then, particularly in low-label settings, the vast majority of training documents, and all at test time, will be without features. Given the centrality of ϑ_d to inference, sampling feature values, or *directly* optimizing them, would be difficult; they could be particularly prone to local, poor optima. The optimization could be a mixed mode of both constrained and unconstrained optimization, and it would need to encode a lot of knowledge about what each of the features means, such as what are valid values for a particular feature. In standard classification, this can be mitigated in part by scaling and normalizing the feature values. Going forward, I assume that y_d has been scaled to be between 0 and 1.

A similar problem exists within discrete neural networks: the network's forward

specification may require a particular discrete value, which can effectively block the gradient from properly backpropagating through the network. To address this, both Maddison et al. (2016) and Jang et al. (2017) independently arrived at a procedure for providing accurate and tunable continuous approximations to discrete selections that allow a unified unconstrained optimization. I adopt Jang et al.'s terminology and refer to the reparametrization as the Gumbel softmax estimator.

If z is sampled from a K dimensional Categorical distribution with probabilities π, i.e., $z \sim \text{Cat}\left(\text{softmax}(\log \pi)\right)$, then the Gumbel softmax estimator approximates this softmax sampling with $x \in \Delta^{K-1}$ as

$$x_k \propto \exp\left(\frac{g_k + \log \pi_k}{\omega}\right), \tag{8.1}$$

where $\omega > 0$ is an annealing parameter and g_k are i.i.d. samples from a Gumbel$(0, 1)$ distribution. As $\omega \to \infty$, the resulting x becomes flatter (higher entropy), representing a more uniform selection; as $\omega \to 0$, x becomes peakier. Note that given fixed ω and Gumbel samples g_k, we can optimize the (soft) Categorical assignment by optimizing the probabilities π instead.

Using the Gumbel softmax estimator

To use the Gumbel softmax estimator at all, there must be *some* distribution over feature values. I opt for document-specific collections of F independent softmax distributions parametrized by $\pi_{d,f}$. Each component of π is a drawn from a global,

$$
\begin{aligned}
\tau_d &\sim \text{Dir}\left(\vartheta_d\right) \\
\vartheta_d &= \vartheta \odot f(\delta y_d), \\
\delta_{k,f} &\sim \text{Normal}(0, 1) \\
y_{d,f} &\sim \text{softmax}(\pi_{d,f}) \\
\pi_{d,f} &\sim \text{Normal}(\pi^{(0)}, 1) \\
\pi^{(0)} &\sim \text{Normal}(0, 1)
\end{aligned}
$$

(a) The basic story for the feature component when the feaures are observed.

$$
\begin{aligned}
\tau_d &\sim \text{Dir}\left(\vartheta_d\right) \\
\vartheta_d &= \vartheta \odot f(\delta y_d), \\
\delta_{k,f} &\sim \text{Normal}(0, 1) \\
y_{d,f} &\sim \text{softmax}\left(\frac{g_{d,f} + \log \pi_{d,f}}{\omega}\right) \\
g_{d,f} &\sim \text{Gumbel}(0, 1) \\
\pi_{d,f} &\sim \text{Normal}(\pi^{(0)}, 1) \\
\pi^{(0)} &\sim \text{Normal}(0, 1)
\end{aligned}
$$

(b) The basic story for the feature component when the feaures are unobserved. Note that ω is a positive annealing parameter.

Figure 8.1: The feature component of bpDMR-Events. Recall that all feature values $y_{d,f}$ are scaled between 0 and 1.

univariate Gaussian with unit variance and mean $\pi^{(0)}$. Specifically, each scaled feature has a simple distribution $y_{d,f} = \frac{\exp(\pi_{d,f})}{1+\exp(\pi_{d,f})}$, where $\pi_{d,f} \sim \text{Normal}(\pi^{(0)}, 1)$. With the lightweight generative backoff story of π_d and $\pi^{(0)}$, we can easily adopt the Gumbel softmax estimator. I show the full feature component story in Figure 8.1.

To fully instantiate this feature component, I also need to specify the non-linearity f. While a componentwise exponential function $f((x)_i) = (\exp x_i)_i$ is often used, I found that convergence, of the optimization and the model inference overall, was

better with a componentwise sigmoid:

$$\sigma(x) = \frac{1}{1+\exp(-x)} \qquad f(\delta y_d) = \left(\sigma(\delta_k^\intercal y_d)\right)_k . \tag{8.2}$$

8.3 Scalable Posterior Inference

As in chapter 7, posterior inference is intractable in this model. The go-to scalable posterior inference algorithm is either a stochastic EM or stochasic variational inference. However, to the best of my knowledge, DMR or DMR-based models tend to involve a hybrid of alternating sampling and MAP estimation, rather than variational inference. I therefore consider both stochastic variational inference (Hoffman et al., 2013, SVI) and streaming collapsed Gibbs sampling (Gao et al., 2016, SGCS). In both cases, I will perform Bayesian inference when possible, and obtain MAP estimates otherwise: under SVI, the MAP estimates are a result of optimizing the ELBO, while under SGCS, they are from optimizing the joint log-likelihood of the collapsed model. In particular, while the observation model yields Bayesian inference algorithms, the feature component generally employs MAP inference.

Both SVI and SGCS require gradients with respect to the feature interpolation weights δ and the feature use priors π_d. The partial derivative of ϑ_d wrt $\pi_{d,f}$ is

$$\frac{\partial \vartheta_{d,k}}{\partial \pi_{d,f}} = \vartheta_k \sigma'(\delta_k^\intercal y_d) \delta_k^\intercal \nabla_{\pi_{d,f}} y_d. \tag{8.3}$$

Meanwhile, the gradient of y_d when it is unobserved is

$$\frac{\partial y_{d,i}}{\partial \pi_{d,f}} = \frac{y_{d,i} 1[i == l] - y_{d,i} y_{d,f}}{\omega \pi_{d,f}}. \tag{8.4}$$

Simply remove the $\omega \pi_{d,f}$ for when features are observed.

8.3.1 Stochastic Variational Inference

I use a fully-factored mean field approximation $q(\tau, \sigma, \phi, \nu, \rho, \delta, t, s)$ that treats all latent variables as independent from one another. This factorization covers the observation component, and I obtain MAP estimates for the feature component.

Optimizing the Observation Component

Each latent variable x will be goverened by its own *variational parameter* $x^{(\lambda)}$: for instance, every ϕ_t will be governed by its own $\phi_t^{(\lambda)}$. To limit the notation, variational parameters will have the same base orthographic form as their corresponding model parameters, but with a special variational symbol $\cdot^{(\lambda)}$. The variational family has the form

$$\overbrace{\prod_t q(\phi_t|\phi_t^{(\lambda)}) \prod_t q(\sigma_t|\sigma_t^{(\lambda)}) \prod_s q(\rho_s|\rho_s^{(\lambda)}) \prod_{i \in F} q(\nu_i|\nu_i^{(\lambda)}) \prod_{j \in R} q(\delta_j|\delta_j^{(\lambda)})}^{\text{global parameters}} \times \tag{8.5}$$

$$\underbrace{\prod_d q(\tau_d|\tau_d^{(\lambda)}) \prod_{d,e} q(t_{d,e}|t_{d,e}^{(\lambda)}) q(s_{d,e}|s_{d,e}^{(\lambda)})}_{\text{local parameters}}. \tag{8.6}$$

I require each variational distribution q to be in the same exponential family as the corresponding distribution in the full model; as discussed in §2.4.4, this permits the natural gradient to be taken, and analytic variational updates to be derived cleanly. I denote all natural parameters by $\eta(\cdot)$.

For the most part, the derivation follows a straight forward application of the mathematical steps from §2.3.2. However, I would like to focus on the derivation of the expectation for the (global) slot parameters. Computing $\mathbb{E}_{q(s_{d,e})q(t_{d,e})q(\sigma)}[\log p(s_{d,e}|t_{d,e},\sigma)]$, and using

$$A^{\mathrm{D}}(x) = \sum_k \log\Gamma(x_k) - \log\Gamma(\sum_k x_k)$$

as the log partition of the Dirichlet (see §2.1.1 and Table 2.1), we have

$$
\begin{aligned}
&= \mathbb{E}_{q(s_{d,e})q(t_{d,e})q(\sigma)}\left[\log \sigma_{t_{d,e}} \cdot \chi\left(s_{d,e}\right)\right] && (8.7)\\
&= \mathbb{E}_{q(t_{d,e})q(\sigma)}\left[\log \sigma_{t_{d,e}}\right] \cdot \mathbb{E}_{q(s_{d,e})}\left[\chi\left(s_{d,e}\right)\right] && (8.8)\\
&= \mathbb{E}_{q(t_{d,e})q(\sigma)}\left[\log \sigma_{t_{d,e}}\right] \cdot \nabla_{\log s^{(\lambda)}_{d,e}} A^{\mathrm{C}}\left(\log s^{(\lambda)}_{d,e}\right) && (8.9)\\
&= \mathbb{E}_{q(t_{d,e})}\left[\mathbb{E}_{q(\sigma)}\left[\log \sigma_{t_{d,e}}\right]\right] \cdot \nabla_{\log s^{(\lambda)}_{d,e}} A^{\mathrm{C}}\left(\log s^{(\lambda)}_{d,e}\right) && (8.10)\\
&= \mathbb{E}_{q(t_{d,e})}\left[\nabla_{\eta\left(\sigma^{(\lambda)}_{t_{d,e}}\right)} A^{\mathrm{D}}\left(\eta\left(\sigma^{(\lambda)}_{t_{d,e}}\right)\right)\right] \cdot \nabla_{\log s^{(\lambda)}_{d,e}} A^{\mathrm{C}}\left(\log s^{(\lambda)}_{d,e}\right) && (8.11)\\
&= \left[\sum_{i=1}^{T} q(t_{d,e}=i \mid t^{(\lambda)}_{d,e}) \nabla_{\eta\left(\sigma^{(\lambda)}_{i}\right)} A^{\mathrm{D}}\left(\eta\left(\sigma^{(\lambda)}_{i}\right)\right)\right] \cdot \nabla_{\log s^{(\lambda)}_{d,e}} A^{\mathrm{C}}\left(\log s^{(\lambda)}_{d,e}\right) && (8.12)\\
&= \underbrace{\left[\sum_{i=1}^{T} \overbrace{t^{(\lambda)}_{d,e,i}}^{\mathbb{R}^1} \overbrace{\nabla_{\eta\left(\sigma^{(\lambda)}_{i}\right)} A^{\mathrm{D}}\left(\eta\left(\sigma^{(\lambda)}_{i}\right)\right)}^{\mathbb{R}^S}\right]}_{\mathbb{R}^S} \cdot s^{(\lambda)}_{d,e} && (8.13)
\end{aligned}
$$

Note that, as expected and needed, the final result is that $\mathbb{E}_q\left[\log p(s_{d,e}|t_{d,e},\sigma)\right]$ is a scalar. To see this more intuitively, the last line (8.14) could be concisely stated as

$$
\left[\underbrace{\overbrace{\nabla_{\eta\left(\sigma^{(\lambda)}\right)} A^{\mathrm{D}}\left(\eta\left(\sigma^{(\lambda)}\right)\right)^{\intercal}}^{\mathbb{R}^{S\times T}} \cdot t^{(\lambda)}_{d,e}}_{\mathbb{R}^{T\times S}}\right] \cdot s^{(\lambda)}_{d,e}. \qquad (8.14)
$$

That is, reweight the gradients of all template-specific slot parameters by how likely that template is to be chosen at all. Finally, reweight this all by how likely the slots actually under consideration are (the right-most product). If the slots were fully

observed, rather than latent, then we could remove the $q(s)$ distribution entirely, so the $s_{d,e}^{(\lambda)}$ would be replaced by a one-hot vector. The remaining derivations have similar forms.

Because I maintained conjugacy in the variational approximation, it is straight forward to obtain the natural gradient,

$$s_{d,e}^{(\lambda)} \propto \exp\left\{\nabla_{\eta\left(\sigma^{(\lambda)}\right)} A^{\mathrm{D}}\left(\eta\left(\sigma^{(\lambda)}\right)\right)^{\intercal} \cdot t_{d,e}^{(\lambda)} + \sum_{m \in e} \nabla_{\eta\left(\rho^{(\lambda)}\right)} A^{\mathrm{D}}\left(\eta\left(\rho^{(\lambda)}\right)\right) \cdot r_{d,e,m}^{(\lambda)}\right\}. \quad (8.15)$$

Optimizing the Feature Component

To infer the feature component variables I optimize the MAP augmented ELBO $\mathcal{L}^{\star}$. That is, even though I do not place variational distributions on the variables in the feature component, they still appear in the ELBO, i.e., variational inference treats MAP-inferred variables (and their distributions in the original model) as constants that get passed through the expectations.

The variables I optimize are the template-feature interpolation weights $\delta \in \mathbb{R}^{T \times F}$, the per-document feature backoff distributions π_d, and the global feature backoff parameter $\pi^{(0)}$. Optimizing both δ and π_d requires backpropagating through the MAP augmented ELBO. Let's consider optimizing π_d. We can write the portion that

is relevant to a document d as

$$\mathcal{L}_d^\star = \langle \log p(\tau \mid \vartheta_d) \rangle + \langle \log p(\pi_d | \pi^{(0)}) \rangle + \langle \log p(y_d | \pi_d) \rangle \tag{8.16}$$

$$= \vartheta_d^\intercal \langle \log \tau_d \rangle - A^{\mathrm{D}}(\vartheta_d) + \log p(\pi_d | \pi^{(0)}) + \log p(y_d | \pi_d). \tag{8.17}$$

Using (8.3) and (8.4), we can write the gradient of $\mathcal{L}_d^\star$ with respect to π_d as

$$\frac{\partial}{\partial \pi_{d,l}} \mathcal{L}_d^\star = \sum_k \langle \log \tau_{d,k} \rangle \frac{\partial \vartheta_{d,k}}{\partial \pi_{d,l}} - \sum_k \psi(\vartheta_{d,k}) \frac{\partial \vartheta_{d,k}}{\partial \pi_{d,l}} + \psi(\sum_k \vartheta_{d,k}) \sum_k \frac{\partial \vartheta_{d,k}}{\partial \pi_{d,l}}. \tag{8.18}$$

Recall that ψ is the digamma function—the derivative of the log gamma function—readily computable through standard scientific libraries.

The gradient for δ is similar. With $\pi^{(0)}$ we have a hierarchical Gaussian model with diagonal covariances: the gradient for $\pi^{(0)}$ is much simpler.

8.3.2 Streaming Collapsed Gibbs Sampling

Variational inference can readily be parallelized and turned into a streaming algorithm. While sampling-based inference has generally used particle filters in streaming settings, Gao et al. (2016) demonstrated an effective alternative that relied on multiplicative discounting.

As discussed in §2.3.3.1, the essence of collapsed Gibbs sampling (in topic models) is the maintenance of joint and marginal counts c, of which words in which documents

are assigned to particular topics. Working in a streaming, mini-batch setting, Gao et al. perform inference and maintain c as normal. At the end of each mini-batch, they discount the counts by $\lambda \in [0, 1]$ as $c = \lambda c$ and then continue on to the next mini-batch.[4]

To apply streaming collapsed Gibbs sampling, I perform mini-batch inference, where in each mini-batch I alternatively sample the observation component and the optimize the feature component.

Inferring the Observation Component

I collapse out all conjugate priors: τ_d, the template-slot distributions σ_t, the template-predicate observation distributions ν_t, and the slot-relation distributions ρ_s. Given this conjugacy, the sampling equations can be derived by following the procedure in §§ 2.3.3.1 and 7.3.

Optimizing the Feature Component

Similar to the variational inference setup, I optimize the feature component variables according to the log joint (collapsed) distribution. Optimizing the feature components is, in part, optimizing the hyperparameters of the per-document template proportions; as is standard, this amounts to optimizing the (log) evidence of some

[4]Care must be taken in the discounting: if any discounted count λc is less than the maximum number of observations any given latent variable is responsible for, then the sampling will likely have negative counts. In plain topic models this threshold is 1, while it is variable for the event template model. I found rounding up to this minimum threshold was sufficient.

"observations," which under this model is given by the (log) Dirichlet-Multinomial compound distribution (Wallach, 2008, and §2.1.1). The resulting objective F_d uses the counts $c_t(d,k)$, reflecting the number of times the **t**emplate k was used in document d, that are accumulated when sampling the observation component:

$$\begin{aligned} F_d = &\log\Gamma\left(\sum_k \vartheta_{d,k}\right) - \log\Gamma\left(\sum_k \vartheta_{d,k} + c_t(d,k)\right) + \\ &\sum_k \log\Gamma\left(\vartheta_{d,k} + c_t(d,k)\right) - \sum_k \log\Gamma\left(\vartheta_{d,k}\right). \end{aligned} \tag{8.19}$$

We can write the gradient of F_d with respect to, e.g., π_d as

$$\begin{aligned} \frac{\partial}{\partial \pi_{d,l}} F_d = &\left(\psi\left(\sum_k \vartheta_{d,k}\right) - \psi\left(\sum_k \vartheta_{d,k} + c_t(d,k)\right)\right)\sum_k \left(\frac{\partial}{\partial \pi_{d,l}}\vartheta_{d,k}\right) + \\ &\sum_k \left(\psi\left(\vartheta_{d,k} + c_t(d,k)\right) - \psi\left(\vartheta_{d,k}\right)\right)\frac{\partial}{\partial \pi_{d,l}}\vartheta_{d,k}. \end{aligned} \tag{8.20}$$

Comment on Optimizing the Feature Component

In initial development of both SVI and SCGS, I experimented with "heavy" (L-BFGS), "medium" (gradient ascent with backtracking line search using Armijo-Wolfe conditions (Armijo, 1966; Wolfe, 1969, 1971)), and "light" (automatically adapting stepsizes) optimizations. Using AdaGrad (Duchi et al., 2011) for the lightweight optimization (see §2.4.3) did not have a large impact on any of the results. However, it was faster than both the medium- and heavy-weight options (significantly faster against L-BFGS).

8.4 Evaluations

In this section I compare bpDMR-Events against just the observation component. Note that this is also the baseline of chapter 7.

For training data, these experiments used a combination of just MUC 3/4 training (1300 documents), just Concretely Annotated *New York Times*, or a combination of the two; heldout data was the MUC 3/4 (200 documents). The vocabulary is held constant throughout all experiments. I extracted *all* semantic frame names as multinomial features.

In initial experiments, I found that the preprocessing values for the vocabulary were important. The experiments here used the 50,000 most frequent predicates (verbs) from the *NYT* after predicates with an inverse document frequency value, computed as

$$\text{idf}(w) = \log \frac{|D|}{|\{d \in D \mid w \in d\}|}$$

of 1 and below were removed (roughly, words that appeared in 660,000 or more of the 1.8M newswire articles). Here, D represents the corpus of all documents $d \in D$. This process removed a number of typographical errors, such as "acccelerate"; improper tokenization, as with "reducethe"; potentially novel words that are a result of the general productivity of derivational morphology in English, as with "overglobalize"; and other pipeline errors such as part of speech errors, as when "viola" was tagged as a verb.

Variational Inference vs. Streaming Collapsed Gibbs

Both implementations were faster the unified models of chapter 7, which I will call UPF. While it is difficult to compare, given the different parametrizations of bpDMR-Events vs. UPF, namely global vs. unique slots, I noticed while the bpDMR-Events models were anywhere from two to ten times faster than the UPF models. Note that while I parallelized both variational and streaming bpDMR-Events models for the following experiments, I controlled for this in the above paragraph.

Overall, I noticed that both variational inference and collapsed Gibbs displayed the same trends. For example, when a parametrization caused perplexity to decrease in, e.g., variational inference, a similar decrease was observed in the sampling methods. However, the actual *values* for variational inference were consistently worse: training and testing on MUC could give sampling perplexities between 300 and 330, but between 7,500 and 9,000 for variational inference. I observed similar results when training on *NYT* and testing on MUC.

Note that, for evaluation consistency, the vocabulary is the same across all experiments and models in this chapter. Overall, the MUC vocabulary is roughly 5% of the entire, processed vocabulary. When training and testing on MUC, this means that many of the defined vocabulary items are unlikely to be observed. This suggests that the streaming sampling is better able to handle these de facto spurious vocabulary items than variational inference, as the inferred sampling posteriors are directly updated according to discrete counts. This further suggests that variational inference

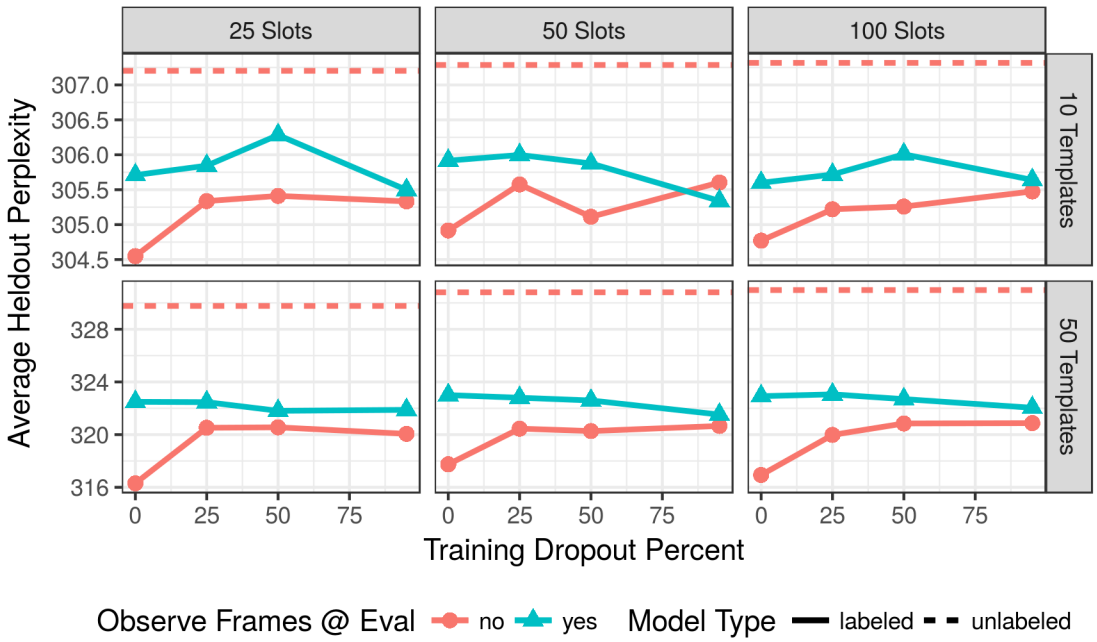


Figure 8.2: Averaged heldout perplexity on MUC, comparing sampling-based bp-DMR models (solid lines) against non-DMR sampling models (dashed lines). The bp-DMR models can either have semantic frame features observed (triangles) during evaluation or not (circles). At 0% dropout, semantic frame features for all 1,300 MUC training documents were observed, while at 25% dropout roughly 975 documents observed semantic frame features and at 50%, 650 documents observed these features.

may benefit from a more aggressive initialization strategy.[5]

Perplexity

In Figure 8.2, I show the averaged perplexity on heldout MUC documents, when trained on MUC. I evaluate models with 10 and 50 templates, each with 25, 50, and 100 (shared) slots, as I vary the percentage of frames that were withheld: 0%

[5]In the variational setting, I initialized the variational observation parameters (i.e., those corresponding to ν_t and ρ_s) as, e.g., $\nu_{t,v} \sim C\text{Gamma}(1,1)$, where C was the average number of documents per parameter to learn. This initialization was very similar to that used by Hoffman et al. (2012).

dropout means that all frame features are observed, while 100% dropout means no features are observed. I compare sampling bpDMR-Event models (solid lines) against sampling non-bpDMR baselines, i.e., just the observation component (dashed lines); I also compare whether bpDMR-Event models have access to the frame features during evaluation (blue triangles do have frame features) or not (red circles). Notice that (1) bpDMR-Events consistently improves perplexity over just the observation component, even when nearly all training features are withheld; (2) bpDMR-Events is able to impute useful, regarding perplexity, heldout features even when they are not observed.

Document Classification

Finally, I consider MUC document classification with bpDMR-Events. In previous work, colleagues and I found that, even under very low resource constraints, a bag-of-words baseline was very difficult to beat (May et al., 2015); in fact, in those experiments, it was almost never beaten by any of the dimensionality reduction techniques they studied.

Overall, experiments with bpDMR-Events yielded nearly the same conclusions, even in the combined settings of learning bpDMR-Events models with both MUC and *NYT*: a simple bag-of-words classifier presented a baseline, that, unfortunately bested all of the event models, whether features were included explicitly, imputed, or completely disregarded. The end conclusions were the same, whether I classified

the learned document representations τ_d using a logistic regression or Naïve Bayes classifier (confirming the experiences of May et al. (2015)).

Chambers and Jurafsky (2011) perform document classification on MUC as well, but there are a number of issues preventing an apples-to-apples comparison. First, they augment MUC with an unspecified portion and amount of newswire data that was selected because it was sufficiently similar to the MUC documents; parameters for this document selection are unspecified. Second, they say that the "average per-token conditional probability" of the document meets or exceeds a "strict threshold" which they "optimized on the training set" yet left unspecified (Chambers and Jurafsky, 2011, pg. 7, under Table 4).

Event Norms and Attribute Expectations

In chapter 5, I learned type-level frame trigger embeddings that were featurized based on frame information. Using the (global) template-feature interpolation weights δ and (global) template-predicate weights ν, the bpDMR-Events models also can yield type-level predicate embeddings.[6] Specifically, I can take the product

$$\nu\delta^\intercal$$

[6]Of course, the template-predicate weights ν themselves can yield embeddings, but the size of the embeddings is constrained to the number of templates.

in order to get predicate embeddings with dimensionality the size of the number of features. To compare against embeddings of size K, we can then use PCA to get components of size K. Applying this procedure and evaluating with SPR-QVEC yields performance competitive with, but generally somewhat lower than, that obtained in chapter 5. As bpDMR-Events models were trained with a higher dropout rate, performance tended to decrease. This indicates that the bpDMR-Events model can yield reasonable attributive embeddings, even when imputing a majority of the features. Note though that the embeddings obtained from bpDMR-Events are not trained on the *same* observations as those in chapter 5—here, which predicates and syntactic relations are modeled is constrained according to (noisy) entity coreference output. The chapter 5 embeddings had no such restriction.

8.5 Summary

In this chapter, I presented bpDMR-Events, a conditionally, generative model of discourse. This model combines Dirichlet Multinomial regression topic models (Mimno and McCallum, 2008), which allows user-provided features to (conditionally) affect the learned topic proportions, with a syntax-only UPF model of chapter 7, which models predicates, syntactic relations, and entity constraints in order to better explain documents. Especially as the syntactic and semantic layers were automatically obtained, the generative nature of the chapter 7 UPF model could, for some

useres, be a limitation. By conditioning on semantic frames, rather than explicitly generating them as part of the document, the bpDMR-Events model provides an alternative method for incorporating much of the same information.

In order to learn the bpDMR-Events model, I presented two methods of inference: sampling and variational inference. One of the limitations of chapter 7 was the general lack of scalability of the models. I presented two easily parallelizable inference algorithms: the first was stochastic variational inference and the second was a sampling approach. The stochastic variational inference algorithm follows the general recipe from Hoffman et al. (2013), while the sampling approach adapts Gao et al. (2016)'s count decay approach. While both inference algorithms were faster than the sampling from chapter 7, I found that parallel sampling resulted in significantly better document modeling performance. While specific experimental decisions likely had a large effect on this difference, the experimental design, and in particular the vocabulary definition, suggests that the variational inference may benefit from initialization that is more targeted to the training corpus.

The UPF model of chapter 7 was generative—it therefore could accommodate documents with missing semantic frames. However, the DMR model (or rather, the DMR *part* of the Mimno and McCallum (2008) model) is not generative. Using a reparametrization of a Categorical distribution (Jang et al., 2017), I adapted the DMR aspect in order to impute and account for missing features at the document level. I demonstrated that this imputation resulted in improved document modeling

against a UPF-style baseline, even as percentage of missing features approached 100% (i.e., as more and more documents had their features hidden from the model). The imputation also resulted in competitive predicate embeddings, as compared to those in chapter 5, despite being trained on different observed data.

There are a number of future directions for the bpDMR-Events model. First, while the Gumbel softmax estimator allowed discrete-output features (such as binary features or count features) to be imputed, a benefit of the DMR model in general is that, in principle, *any* feature could be included. While the bpDMR-Events model accounts for unobserved discrete features, extending it to handle any type of unobserved features would significantly broaden the areas in which it could be applied. For example, using features provided by the computer vision community, which are often just real-valued vectors, bpDMR-Events could learn grounded, generative narrative models (Huang et al., 2016b).

Second, models of events and discourse should be able to handle the productivity of a language. For example, to describe a person learning about a topic, twenty years ago one might have said, "Chris read the book," but now one might say, "Chris googled it." Zhai and Boyd-Graber (2013) approached this general problem for topic modeling by introducing a generative character distribution over the vocabulary items. Incorporating this, or a related, generative sub-model into the larger bpDMR-Events model may allow the model to handle new actions, thereby generalizing to new experiences.

Third, I only considered scenarios where a document's features were fully observed,

or were fully unobserved. This setting, as in chapter 7, reflected a workflow where running an analytic on any given document might be simple, but running that analytic on *many* documents could be difficult. Future work can examine some of a document's features were observed, but some of them weren't. This setting would subsume the current one; it would correspond to a workflow where regardless of how easy or difficult it is to run an analytic over documents, the output of that analytic might not be as trustworthy as desired. This improvement might allow bpDMR-Events to aggregate more information more confidently.

Chapter 9

Conclusion

In this thesis I explored multiple types of unsupervised induction of meanings of words, sentences and documents.

At the word level (chapter 5), I presented a general tensor factorization method, linked to standard methods in the word embedding community, and I explored how to turn multiple, overlapping, and noisy semantic annotations into usable, decomposable counts. I compared these against three different attribute-based datasets; the frame-enriched word embeddings against higher correlation against these datasets, indicating frame better encode semantic properties and expectations.

At the sentence level (chapter 6) I presented an EM-based algorithm to learn large, refined, and possibly lexicalized syntactic tree fragments. Quantitatively, these fragments can be used for syntactic parsing to achieve competitive performance, while simultaneously presenting analyses of commonly occuring phrases and constructions

in the dataset. The fragment induction algorithm uses a user-provided constraint set. While a simple counting based constraint set helps cheaply emulate more complex statistical methods, this algorithm could be adapted to learn different kinds of syntactic frames. For example, composing the constraint set of common verb frames, or hard-to-analyze prepositional phrases could make subsequent induced grammars more aware of syntactically-manifested semantic ambiguities. The analyses of chapter 6 identify what predicate argument structures, and other deep refinement patterns, can be learned automatically. Whereas chapter 5 studied how to enrich word meanings, this chapter demonstrates an ability to learn deeper, lexicalized syntactic frames.

At the document level (chapters 7 and 8) I present two Bayesian models for templated event induction. These models make theoretical (AI) contributions, propose methods to make these models scale to larger corpora, allow auxiliary input features to help guide the induction, and demonstrate how to overcome missing features. While some implementation insights helped these models scale, scalability was achieved primary through parallelization. Scalability is still a challenge: these hierarchical models are not lightweight.

Throughout this thesis, I used data from the Concretely Annotated Corpora (CAC Ferraro et al., 2014, chapter 4). This is a large corpus of more than 15 million documents, of which more than a third are freely available for download, that have been automatically processed and annotated with different NLP tools.[1] While this was

[1]The remaining portion are available through the Linguistic Data Consortium: `https://www.ldc.upenn.edu/`.

tremendous asset to be able to use, linguistically, the models, particularly the event models, require a lot of pre-existing annotations. Errors in coreference resolution are a concern.

I explicitly demonstrated how the unsupervised syntactic frame induction of chapter 6 can be applied to a morphologically rich language like Korean. It would also be interesting to examine the evaluation framework of chapter 5 in additional languages: this would necessitate obtaining attributive judgments (or judgments describing how likely certain properties are to be true) in those languages.

In general, however, it is an open question how to adapt the majority of the methods presented in this thesis to languages other than English. Though the tensor factorization method from chapter 5 demonstrates an ability to leverage noisy annotations, the method—and motivating story behind the application of the method—still presupposes the existence of semantic analyzers. Chapters 7 and 8 rely on syntactic and entity coreference annotations; applying those discourse models to additional languages would first require identifying acceptable syntactic and entity analytics in those languages. The core machine learning explored in chapter 8 might be applied to multilingual settings—where discriminative features may be difficult to acquire—though the application of that machine learning would likely need to be modified.

9.1 Future Directions

In addition to the challenges identifed throughout this thesis and highlighted above, there are a number of interesting, holistic future directions one can take this work. I consider three below.

Semantics-bearing Applications

Frames (§§ 3.1.2 and 7.1) are meant to schematize common experiences and knowledge, and help us (and systems) make sense of complicated, interwoven concepts. Linguistically, there are also exciting possibilities of how to handle generic knowledge—characterizing situations as deviations from what is generally expected to be true. It is an interesting question the extent to which frames—at the type level through resources, or at an instance level through word, sentence, or document representations—can help improve semantic- and meaning-based user applications. For instance, can improved word representations be combined with memoized syntactic frames to present more (semantically) coherent predicate argument analyses? Can improved document modeling help in summarization tools, or aide in the extraction and analysis of significant events (as determined by domain-specific experts)?

Grounding Event Meanings

When we describe an experience, we tend to highlight the most salient events and ignore many of the prerequisite (or less interesting) aspects. For instance, when we

describe eating at a restaurant, we talk about ordering and eating a meal, but we probably gloss over waiting for a maître d' to seat us (or standing at the back of a line). Human-interacting systems, such as assistive technologies for visually impaired users, need to account for these. One way to accomplish this is through a more comprehensive understanding of multimodal semantics; meaning induced solely from text is going to encode well those experiences that are deemed salient enough to be reported at all, yet struggle to represent "silent," or background, events.

Event Modeling with Graphical Models and Neural Nets

Chapter 8 used a shallow neural network to aid document representation induction. And in general, neural nets have repeatedly demonstrated their ability to construct robust and useful representations of their input. On the other hand, graphical models provide principled methods for specifying rich priors or structures, especially when training data is sparse. In what other ways can neural nets and graphical models be combined?

Bibliography

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

Apoorv Agarwal, Daniel Bauer, and Owen Rambow. Using frame semantics in natural language processing. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, pages 30–33, Baltimore, MD, USA, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W14-3008`.

Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Polyglot: Distributed Word Representations for Multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W13-3520`.

BIBLIOGRAPHY

Nikolaos Aletras and Mark Stevenson. Evaluating topic coherence using distributional semantics. In *Proceedings of the Tenth International Workshop on Computational Semantics (IWCS-10)*, 2013.

Shun-Ichi Amari. Differential geometry of curved exponential families-curvatures and information loss. *The Annals of Statistics*, pages 357–385, 1982.

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Richard C. Anderson. The notion of schemata and the educational enterprise: General discussion of the conference. In Richard C. Anderson, Rand J. Spiro, and William E. Montague, editors, *Schooling and the Acquisition of Knowledge*. Erlbaum, Hillsdale, NJ, 1977.

Apache UIMA Community. Apache uimaFIT guide and reference. Technical report, Apache, 2013.

K.R. Apt and M. Wallace. *Constraint logic programming using ECLiPSe*. Cambridge University Press, 2006.

Larry Armijo. Minimization of functions having lipschitz continuous first partial derivatives. *Pacific Journal of mathematics*, 16(1):1–3, 1966.

Fred Attneave. *Applications of Information Theory to Psychology: A summary of basic concepts, methods, and results*. Holt, 1959.

BIBLIOGRAPHY

Collin Baker, Michael Ellsworth, and Katrin Erk. Semeval'07 task 19: Frame semantic structure extraction. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 99–104. Association for Computational Linguistics, 2007.

Collin F Baker, Charles J Fillmore, and John B Lowe. The Berkeley Framenet Project. In *ACL*, 1998.

Niranjan Balasubramanian, Stephen Soderland, Mausam, and Oren Etzioni. Generating coherent event schemas at scale. In *EMNLP*, Seattle, Washington, USA, October 2013. URL `http://www.aclweb.org/anthology/D13-1178`.

David Bamman and Noah Smith. Unsupervised discovery of biographical structure from text. *TACL*, 2(10):363–376, 2014.

David Bamman and Noah A. Smith. New Alignment Methods for Discriminative Book Summarization. *CoRR*, abs/1305.1319, 2013.

David Bamman, Brendan O'Connor, and Noah A. Smith. Learning latent personas of film characters. In *ACL*, 2013.

David Bamman, Ted Underwood, and Noah A. Smith. A bayesian mixed effects model of literary character. In *ACL*, Baltimore, Maryland, June 2014. URL `http://www.aclweb.org/anthology/P14-1035`.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Martha Palmer, and Nathan Schneider. Abstract meaning

representation for sembanking. In *In Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*. Citeseer, 2013.

Mohit Bansal and Dan Klein. Simple, accurate parsing with an all-fragments grammar. In *Proceedings of ACL*, pages 1098–1107. Association for Computational Linguistics, 2010.

David Barber and Pi erre de van Laar. Variational cumulant expansions for intractable distributions. *Journal of Artificial Intelligence Research*, pages 435–455, 1999.

Frederic Charles Bartlett. *Remembering: A study in experimental and social psychology*, volume 14. Cambridge University Press, 1933.

Jon Barwise. Some computational aspects of situation semantics. In *Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics*, pages 109–111, Stanford, California, USA, June 1981. Association for Computational Linguistics. doi: 10.3115/981923.981955. URL `http://www.aclweb.org/anthology/P81-1026`.

Jon Barwise and John Perry. Situations and attitudes. *The Journal of Philosophy*, 78(11):668–691, 1981.

Cosmin Bejan and Sanda Harabagiu. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Associa-*

tion for Computational Linguistics, pages 1412–1422, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

Cosmin Adrian Bejan. Unsupervised discovery of event scenarios from texts. In *FLAIRS*, 2008.

Cosmin Adrian Bejan. *Learning Event Structures from Text*. PhD thesis, University of Texas, Dallas, 2009.

Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

Peter J. Bickel and Kjell A. Doksum. *Mathematical Statistics, Basic Ideas and Selected Topics, Vol. 1, (2nd Edition)*. Pearson, 2nd edition, May 2006.

Christopher M. Bishop. *Pattern recognition and machine learning*. springer New York, 2006.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet Allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

Phil Blunsom and Trevor Cohn. Unsupervised induction of tree substitution grammars for dependency parsing. In *Proceedings of EMNLP*, pages 1204–1213, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Rens Bod. Using an annotated corpus as a stochastic grammar. In *Proceedings of EACL*, pages 37–44. Association for Computational Linguistics, 1993.

BIBLIOGRAPHY

Rens Bod. What is the minimal set of fragments that achieves maximal parse accuracy? In *Proceedings of ACL*, pages 66–73. Association for Computational Linguistics, 2001.

Claire Bonial, William Corvey, Martha Palmer, Volha V. Petukhova, and Harry Bunt. A hierarchical unification of LIRICS and VerbNet semantic roles. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 483–489, September 2011.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge university press, 2004.

Amanda Christy Brown and Katherine Schluten. Writing rules! advice from the times on writing well. *The Learning Network*, Sep 2012. URL `https://learning.blogs.nytimes.com/2012/09/20/writing-rules-advice-from-the-new-york-times-on-writing`.

Luana Bulat, Douwe Kiela, and Stephen Clark. Vision and feature norms: Improving automatic feature norm learning through cross-modal maps. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 579–588, San Diego, California, June 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N16-1071`.

Luana Bulat, Stephen Clark, and Ekaterina Shutova. Modelling metaphor with

attribute-based semantics. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 523–528, Valencia, Spain, April 2017. Association for Computational Linguistics.

Kevin Burton, Akshay Java, and Ian Soboroff. The ICWSM 2009 Spinn3r Dataset. In *Third Annual Conference on Weblogs and Social Media (ICWSM 2009)*. AAAI, 2009.

Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr., and T.M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 1306–1313. AAAI Press, 2010.

Greg N Carlson. Thematic roles and their role in semantic interpretation. *Linguistics*, 22(3):259–280, 1984.

Xavier Carreras and Lluís Màrquez. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In Hwee Tou Ng and Ellen Riloff, editors, *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 89–97, Boston, Massachusetts, USA, May 6 - May 7 2004. Association for Computational Linguistics.

BIBLIOGRAPHY

Xavier Carreras and Lluís Màrquez. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, pages 152–164. Association for Computational Linguistics, 2005.

Hector-Neri Castañeda. Comments on donald davidson's 'the logical form of action sentences'. In Nicholas Rescher, editor, *The Logic of Decision and Action*, pages 104–112. University of Pittsburgh Press, 1967.

Nathanael Chambers. Event schema induction with a probabilistic entity-driven model. In *EMNLP*, 2013.

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*, pages 602–610. Association for Computational Linguistics, 2009.

Nathanael Chambers and Dan Jurafsky. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 976–986, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P11-1098`.

BIBLIOGRAPHY

Nathanael Chambers and Daniel Jurafsky. Unsupervised learning of narrative event chains. In *ACL*, 2008.

Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L Boyd-graber, and David M Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

Eugene Charniak. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of AAAI*, pages 598–603, 1997.

Snigdha Chaturvedi. *Structured Approaches for Exploring Interpersonal Relationships in Natural Language Text*. PhD thesis, University of Maryland, College Park, 2016.

Harr Chen, Edward Benson, Tahira Naseem, and Regina Barzilay. In-domain relation discovery with meta-constraints via posterior regularization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 530–540, Portland, Oregon, USA, June 2011a. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P11-1054`.

Jianshu Chen, Ji He, Yelong Shen, Lin Xiao, Xiaodong He, Jianfeng Gao, Xinying Song, and Li Deng. End-to-end learning of lda by mirror-descent back propagation over a deep architecture. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1765–1773. Curran Associates, Inc., 2015.

BIBLIOGRAPHY

Liyin Chen, Siaw-Fong Chung, and Chao-Lin Liu. A construction grammar approach to prepositional phrase attachment: Semantic feature analysis of v np1 into np2 construction. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, pages 607–614, Singapore, December 2011b. Institute of Digital Enhancement of Cognitive Processing, Waseda University. URL `http://www.aclweb.org/anthology/Y11-1065`.

Yun-Nung Chen, William Yang Wang, and Alexander I Rudnicky. Leveraging frame semantics and distributional semantics for unsupervised semantic slot induction in spoken dialogue systems. In *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pages 584–589. IEEE, 2014.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. Probabilistic frame induction. In *NAACL*, 2013.

Jen-Tzung Chien and Meng-Sung Wu. Adaptive bayesian latent semantic analysis. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(1):198–207, 2008.

Noam Chomsky. Lectures on government and binding. *Foris, Dordrecht*, 1981.

Tagyoung Chung, Matt Post, and Daniel Gildea. Factors affecting the accuracy of korean parsing. In *Proceedings of the NAACL HLT Workshop on Statistical Parsing of Morphologically-Rich Languages (SPMRL)*, pages 49–57, Los Angeles, California, USA, June 2010.

BIBLIOGRAPHY

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.

Trevor Cohn, Sharon Goldwater, and Phil Blunsom. Inducing compact but accurate tree-substitution grammars. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 548–556, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-41-1. URL `http://dl.acm.org/citation.cfm?id=1620754.1620834`.

Trevor Cohn, Phil Blunsom, and Sharon Goldwater. Inducing tree-substitution grammars. *Journal of Machine Learning Research*, 11:3053–3096, December 2010. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=1756006.1953031`.

Alain Colmerauer and Philippe Roussel. The birth of prolog. In *History of programming languages—II*, pages 331–367. ACM, 1996.

Ryan Cotterell, Adam Poliak, Benjamin Van Durme, and Jason Eisner. Explaining and generalizing skip-gram through exponential family principal component analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, April 2017.

et al. Cunningham. Developing language processing components with GATE version 8. Technical report, University of Sheffield Department of Computer Science, November 2014.

BIBLIOGRAPHY

Hamish Cunningham, Robert J Gaizauskas, and Yorick Wilks. A general architecture for text engineering (GATE): A new approach to language engineering r & d. Technical report, University of Sheffield Department of Computer Science, 1995.

Agata Cybulska and Piek Vossen. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 2014.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith. Probabilistic frame-semantic parsing. In *NAACL*, 2010.

Dipanjan Das, Desai Chen, André FT Martins, Nathan Schneider, and Noah A Smith. Frame-semantic parsing. *Computational linguistics*, 40(1):9–56, 2014.

Rajarshi Das, Manzil Zaheer, and Chris Dyer. Gaussian LDA for topic models with word embeddings. In *ACL (1)*, pages 795–804, 2015.

Donald Davidson. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*. University of Pittsburgh Pressyear = 1967, 1967.

Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Stanford University, 2008.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

BIBLIOGRAPHY

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Qiming Diao and Jing Jiang. A unified model for topics, events and users on Twitter. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1869–1879, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D13-1192`.

Quang Do, Yee Seng Chan, and Dan Roth. Minimally supervised event causality identification. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 294–303, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D11-1027`.

Ellen K Dodge and Miriam R L Petruck. Representing caused motion in embodied construction grammar. In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 39–44, Baltimore, MD, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W14-2408`.

David Dowty. Thematic proto-roles and argument selection. *language*, pages 547–619, 1991.

Markus Dreyer and Jason Eisner. Better informed training of latent syntactic features.

In *Proceedings of EMNLP*, pages 317–326, Sydney, Australia, July 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W06/W06-1638`.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12 (Jul):2121–2159, 2011.

Richard Eckart de Castilho and Iryna Gurevych. A broad-coverage collection of portable NLP components for building shareable analysis pipelines. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 1–11, Dublin, Ireland, August 2014. Association for Computational Linguistics and Dublin City University. URL `http://www.aclweb.org/anthology/W14-5201`.

Jacob Eisenstein, Amr Ahmed, and Eric P Xing. Sparse additive generative models of text. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1041–1048, 2011.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, pages 1606–1615, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1184.

Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998.

Francis Ferraro. Toward improving the automated classification of metonymy in text corpora. http://hdl.handle.net/1802/14985, 2011. URL http://cs.jhu.edu/~ferraro/papers.html#ferraro-ur-thesis. Undergraduate Honors Thesis.

Francis Ferraro and Benjamin Van Durme. A Unified Bayesian Model of Scripts, Frames and Language. In *AAAI*, 2016.

Francis Ferraro, Matt Post, and Benjamin Van Durme. Judging grammaticality with count-induced tree substitution grammars. In *The Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications*, 2012a.

Francis Ferraro, Benjamin Van Durme, and Matt Post. Toward tree substitution grammars with latent annotations. In *Proceedings of the NAACL-HLT Workshop on the Induction of Linguistic Structure*, pages 23–30, Montréal, Canada, June 2012b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W12-1904.

Francis Ferraro, Max Thomas, Matthew R. Gormley, Travis Wolfe, Craig Harman, and Benjamin Van Durme. Concretely Annotated Corpora. In *AKBC*, 2014.

BIBLIOGRAPHY

Francis Ferraro, Adam Poliak, Ryan Cotterell, and Benjamin Van Durme. Frame-based continuous lexical semantics through exponential family tensor factorization and semantic proto-roles. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 97–103, Vancouver, Canada, August 2017. Association for Computational Linguistics.

Charles Fillmore. Frame semantics. *Linguistics in the morning calm*, pages 111–137, 1982.

Charles J Fillmore. The case for case. In *Proceedings of the Texas Symposium on Language Universals*. ERIC, 1967.

Charles J. Fillmore. An alternative to checklist theories of meaning. In *Proceedings of the First Annual Meeting of the Berkeley Linguistics Society*, 1975.

Charles J Fillmore. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences*, 280(1):20–32, 1976.

Charles J. Fillmore and Collin Baker. A frames approach to semantic analysis. In Bernd Heine and Heiko Narrog, editors, *The Oxford Handbook of Linguistic Analysis*, chapter 13. Oxford University Press, 1 edition, 2009.

Charles J Fillmore, Paul Kay, and Mary Catherine O'connor. Regularity and idiomaticity in grammatical constructions: The case of let alone. *Language*, pages 501–538, 1988.

BIBLIOGRAPHY

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. Semantic role labeling with neural network factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 960–970, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

Lea Frermann, Ivan Titov, and Manfred Pinkal. A hierarchical bayesian model for unsupervised induction of script knowledge. In *EACL*, 2014.

Luana Făgărăşan, Eva Maria Vecchi, and Stephen Clark. From distributional semantics to feature norms: grounding semantic models in human perceptual data. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 52–57, London, UK, April 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W15-0107`.

N. Fuhr. Probabilistic datalog—a logic for powerful retrieval methods. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 282–290, 1995.

H. Gallaire, J. Minker, and J.M. Nicolas. Logic and databases: A deductive approach. *ACM Computing Surveys (CSUR)*, 16(2):153–185, 1984.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB: The Paraphrase Database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia, June 2013. Association for Computational Linguistics.

BIBLIOGRAPHY

Yang Gao, Jianfei Chen, and Jun Zhu. Streaming Gibbs Sampling for LDA Model. *arXiv preprint arXiv:1601.01142*, 2016.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

George Giannakopoulos, Elena Lloret, John M. Conroy, Josef Steinberger, Marina Litvak, Peter Rankel, and Benoit Favre, editors. *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*. Association for Computational Linguistics, Valencia, Spain, April 2017.

Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.

Erving Goffman. *Frame Analysis: An Essay on the Organization of Experience*. Harvard University Press, 1974.

Adele E Goldberg. *Constructions at work: The nature of generalization in language*. Oxford University Press on Demand, 2006.

Yoav Goldberg and Omer Levy. word2vec explained: Deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

Joshua Goodman. Efficient algorithms for parsing the dop model. In *Proceedings of EMNLP*, pages 143–152, 1996a.

BIBLIOGRAPHY

Joshua Goodman. Parsing algorithms and metrics. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 177–183, Stroudsburg, PA, USA, 1996b. Association for Computational Linguistics.

Matthew R. Gormley, Margaret Mitchell, Benjamin Van Durme, and Mark Dredze. Low-resource semantic role labeling. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1177–1187, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P14-1111`.

Swapna Gottipati, Minghui Qiu, Yanchuan Sim, Jing Jiang, and Noah A. Smith. Learning topics and positions from Debatepedia. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1858–1868, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D13-1191`.

Amit Goyal, Ellen Riloff, and Hal Daume III. Automatically producing plot unit representations for narrative text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 77–86, Cambridge, MA, October 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D10-1008`.

Amit Goyal, Ellen Riloff, and Hal Daumë III. A computational model for plot units. *Computational Intelligence*, 29(3):466–488, 2013.

BIBLIOGRAPHY

Arthur C Graesser, Keith K Millis, and Rolf A Zwaan. Discourse comprehension. *Annual Review of Psychology*, 48(1):163–189, 1997.

Arthur C Graesser, Brent Olde, and Bianca Klettke. How does the mind construct and represent stories. *Narrative Impact: Social and Cognitive Foundations*, pages 229–262, 2002.

Mark Granroth-Wilding and Stephen Clark. What happens next? event prediction using a compositional neural network model. In *AAAI*, 2016.

Clayton Greenberg, Asad Sayeed, and Vera Demberg. Improving unsupervised vector-space thematic fit evaluation via role-filler prototype clustering. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 21–31, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N15-1003`.

Larry J Griffin. Narrative, event-structure analysis, and causal interpretation in historical sociology. *American Journal of Sociology*, 98(5):1094–1133, 1993.

Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *PNAS*, 101(Suppl. 1):5228–5235, 2004.

Olga Gurevich, Richard Crouch, Tracy Holloway King, and Valeria De Paiva. Dever-

bal nouns in knowledge representation. *Journal of Logic and Computation*, 18(3): 385–404, 2007.

Aria Haghighi and Dan Klein. An entity-level approach to information extraction. In *Proceedings of the ACL 2010 Conference Short Papers*. Association for Computational Linguistics, 2010.

John Hale. A probabilistic earley parser as a psycholinguistic model. In *NAACL*, 2001.

Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. *Cohesion in English*. Pearson, 1976.

Chung-hye Han, Na-Rae Han, and Eon-Suk Ko. Bracketing guidelines for penn korean treebank. Technical report, IRCS, University of Pennsylvania, 2001.

Na-Rae Han and Shijong Ryu. Guidelines for Penn Korean Treebank. Technical report, University of Pennsylvania, 2005.

Patrick Hanks. *Lexical Analysis: Norms and Exploitations*. MIT Press, 2013.

Mary Hare, Michael Jones, Caroline Thomson, Sarah Kelly, and Ken McRae. Activating Event Knowledge. *Cognition*, 111(2):151–167, 2009.

Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

Joshua K. Hartshorne, Claire Bonial, and Martha Palmer. The VerbCorner Project: Findings from Phase 1 of crowd-sourcing a semantic decomposition of verbs. In

Proceedings of the Annual Meeting of the Association for Computational Linguistics, pages 397–402. Association for Computational Linguistics, 2014. URL http://www.aclweb.org/anthology/P14-2065.

Joshua Hartstone, Claire Bonial, and Martha Palmer. The VerbCorner project: Toward an empirically-based semantic decomposition of verbs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1438–1442, 2013.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Irene Heim. *The Semantics of Definite and Indefinite Noun Phrases*. PhD thesis, UMass Amherst, 1982.

David R Heise. Modeling event structures. *Journal of Mathematical Sociology*, 14 (2-3):139–169, 1989.

James Henderson, Paola Merlo, Ivan Titov, and Gabriele Musillo. Multi-lingual joint parsing of syntactic and semantic dependencies with a latent variable model. *Computational Linguistics*, 39(4), 2013.

Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 2016.

BIBLIOGRAPHY

Stephen Hiltner. How to write a new york times headline. *Times Insider*, April 2017. URL `https://www.nytimes.com/2017/04/09/insider/how-to-write-a-new-york-times-headline.html`.

Jerry R Hobbs. Ontological promiscuity. In *ACL*, 1985.

Jerry R Hobbs. Toward a useful concept of causality for lexical semantics. *Journal of Semantics*, 22(2):181–209, 2005.

Matt Hoffman, David M Blei, and David M Mimno. Sparse stochastic inference for latent dirichlet allocation. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, 2012.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Thomas Hofmann. Probabilistic Latent Semantic Analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

Antti Honkela, Tapani Raiko, Mikael Kuusela, Matti Tornio, and Juha Karhunen. Approximate riemannian conjugate gradient learning for fixed-form variational bayes. *Journal of Machine Learning Research*, 11(Nov):3235–3268, 2010.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. Voss, Jiawei Han, and

Avirup Sil. Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268, Berlin, Germany, August 2016a. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1025`.

Ruihong Huang and Ellen Riloff. Multi-faceted event recognition with bootstrapped dictionaries. In *NAACL*, Atlanta, Georgia, June 2013. URL `http://www.aclweb.org/anthology/N13-1005`.

Ting-Hao (Kenneth) Huang, Francis Ferraro, Nasrin Mostafazadeh, Misra Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh, Lucy Vanderwende, Michel Galley, and Margaret Mitchell. Visual Storytelling. In *NAACL*, 2016b. Equal contribution: TH, FF.

Chung Hee Hwang and Lenhart K Schubert. Episodic logic: A situational logic for natural language processing. *Situation Theory and its Applications*, 3:303–338, 1993.

Jena D. Hwang, Rodney D. Nielsen, and Martha Palmer. Towards a domain independent semantics: Enhancing semantic representation with construction grammar. In *Proceedings of the NAACL HLT Workshop on Extracting and Using Constructions in Computational Linguistics*, pages 1–8, Los Angeles, California, June 2010. Association for Computational Linguistics.

BIBLIOGRAPHY

Nancy Ide and Jens Grivolla, editors. *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*. Association for Computational Linguistics and Dublin City University, Dublin, Ireland, August 2014. URL `http://www.aclweb.org/anthology/W14-52`.

Joseph Irwin, Mamoru Komachi, and Yuji Matsumoto. Narrative schema as world knowledge for coreference resolution. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 86–92, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W11-1913`.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. Feuding families and former friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1534–1544, San Diego, California, June 2016. Association for Computational Linguistics.

Tommi Jaakkola and Michael Jordan. A variational approach to bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics*, volume 82, page 4, 1997.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017.

BIBLIOGRAPHY

Bram Jans, Steven Bethard, Ivan Vulić, and Marie Francine Moens. Skip n-grams and ranking functions for predicting script events. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 336–344. Association for Computational Linguistics, 2012.

Mark Johnson. PCFG models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632, 1998.

Mark Johnson, Thomas L Griffiths, and Sharon Goldwater. Adaptor grammars: A framework for specifying compositional nonparametric bayesian models. In *Advances in Neural Information Processing Systems*, pages 641–648, 2007.

Aravind K. Joshi and Yves Schabes. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages: Beyond Words*, volume 3, pages 71–122. Springer, 1997.

Dan Jurafsky and James H Martin. *Speech and Language Processing*. Pearson Prentice Hall, Upper Saddle River, New Jersey, 2 edition, 2008.

Hans Kamp. A theory of truth and semantic representation. In P. Portner and B. H. Partee, editors, *Formal Semantics - the Essential Readings*, pages 189–222. Blackwell, 1981.

Niels Kasch. *Mining Commonsense Knowledge from the Web: Towards Inducing*

Script-like Structures From Large-scale Text Sources. PhD thesis, University of Maryland, Baltimore County, 2012.

Paul Kay. Construction grammar. In Jeff Verschueren, Jan-Ola Ostman, and Jan Blommaert, editors, *Handbook of Pragmatics: manual*. J. Benjamins, 1995.

Dimitar Kazakov. Combining LAPIS and WordNet for the learning of LR parsers with optimal semantic constraints. In *International Conference on Inductive Logic Programming*, pages 140–151. Springer, 1999.

S. Sathiya Keerthi, Tobias Schnabel, and Rajiv Khanna. Towards a better understanding of predict and count models. *CoRR*, abs/1511.02024, 2015. URL `http://arxiv.org/abs/1511.02024`.

Anthony Kenny. *Action, emotion and will*. Routledge, 1963.

Saman Khalkhali, Jeffrey Wammes, and Ken McRae. Integrating Words that Refer to Typical Sequences of Events. *Canadian Journal of Experimental Psychology/Revue Canadienne de Psychologie Expérimentale*, 66(2):106, 2012.

Hyuk Kim. *Organizational Strategy Development for the Pharmaceutical Industry: Event Structure Analysis, Comparative Boolean Analysis, and Analogical Reasoning Model*. PhD thesis, Rutgers, The State University Of New Jersey, 2010.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*

(EMNLP), pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D14-1181.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *he International Conference on Learning Representations (ICLR)*, 2014.

Dan Klein and Christopher D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 423–430, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1075096.1075150. URL http://dx.doi.org/10.3115/1075096.1075150.

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

Angelika Kratzer. Situations in natural language semantics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2016 edition, 2016.

Alex Lascarides and Nicholas Asher. Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and philosophy*, 16(5):437–493, 1993.

Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1536–

1545, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Jey Han Lau, David Newman, and Timothy Baldwin. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 530–539, Gothenburg, Sweden, April 2014. Association for Computational Linguistics.

Wendy G Lehnert. Plot units and narrative summarization. *Cognitive Science*, 5(4): 293–331, 1981.

Beth Levin. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, 1993.

Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, June 2014a. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P14-2050`.

Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014b.

BIBLIOGRAPHY

Roger Levy. Expectation-based syntactic comprehension. *Cognition*, 106(3):1126–1177, 2008.

Roger Levy. Integrating surprisal and uncertain-input models in online sentence comprehension: formal techniques and empirical results. In *ACL*, 2011.

Roger Levy and T. Florian Jaeger. Speakers optimize information density through syntactic reduction. In *NIPS*, 2006.

Qi Li, Heng Ji, and Liang Huang. Joint event extraction via structured prediction with global features. In *ACL*, Sofia, Bulgaria, August 2013. URL `http://www.aclweb.org/anthology/P13-1008`.

Shasha Liao and Ralph Grishman. Using document level cross-event inference to improve event extraction. In *ACL*, 2010.

Dekang Lin and Patrick Pantel. DIRT — Discovery of Inference Rules from Text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pages 323–328. ACM, 2001.

Ken Litkowski. Senseval-3 task: Automatic labeling of semantic roles. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 141–146, 2004.

Jun S Liu. The collapsed gibbs sampler in bayesian computations with applications

to a gene regulation problem. *Journal of the American Statistical Association*, 89 (427):958–966, 1994.

Edward Loper, Szu-Ting Yi, and Martha Palmer. Combining lexical resources: Mapping between propbank and verbnet. In *Proceedings of the 7th International Workshop on Computational Linguistics*, 2007.

Alejandra Lorenzo and Christophe Cerisara. Unsupervised frame based semantic role induction: application to french and english. In *ACL Workshop on Statistical Parsing and Semantic Processing of Morphologically Rich Languages*, Jeju, Republic of Korea, July 12 2012. URL `http://www.aclweb.org/anthology/W12-3404`.

Jing Lu and Vincent Ng. Joint learning for coreference resolution. In *Proceedings of the Association for Computational Linguistics*, 2017.

David J. MacKay. *Information Theory, Learning and Inference*. Cambridge University Press, Cambridge, 2003.

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. NOMLEX: A lexicon of nominalizations. In *Proceedings of EURALEX*, volume 98, pages 187–193, 1998.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

BIBLIOGRAPHY

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL Demos*, pages 55–60, 2014. URL `http://www.aclweb.org/anthology/P/P14/P14-5010`.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19 (2):313–330, 1993.

Tania Marques and Katrien Beuls. Evaluation strategies for computational construction grammars. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, 2016.

Mstislav Maslennikov and Tat-Seng Chua. A multi-resolution framework for information extraction from free text. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 592–599, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P07-1075`.

Jiří Materna. LDA-Frames: an unsupervised approach to generating semantic frames. In *Computational Linguistics and Intelligent Text Processing*, pages 376–387. Springer, 2012.

BIBLIOGRAPHY

Jirí Materna. Parameter estimation for lda-frames. In *HLT-NAACL*, pages 482–486, 2013.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. Probabilistic cfg with latent annotations. In *Proceedings of ACL*, pages 75–82, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. URL `http://dx.doi.org/10.3115/1219840.1219850`.

Chandler May, Francis Ferraro, Alan McCree, Jonathan Wintrode, Daniel Garcia-Romero, and Benjamin Van Durme. Topic identification and discovery on text and speech. In *EMNLP*, 2015.

Jon D McAuliffe and David M Blei. Supervised topic models. In *Advances in neural information processing systems*, pages 121–128, 2008.

John McCarthy, Marvin L. Minsky, Nathaniel Rochester, and Claude E. Shannon. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4):12, 2006.

Ken McRae, Virginia R De Sa, and Mark S Seidenberg. On the nature and scope of featural representations of word meaning. *Journal of Experimental Psychology: General*, 126(2):99, 1997a.

Ken McRae, Todd R. Ferretti, and Liane Amyote. Thematic roles as verb-specific concepts. *Language and cognitive processes*, 12(2-3):137–176, 1997b.

BIBLIOGRAPHY

Ken McRae, George S Cree, Mark S Seidenberg, and Chris McNorgan. Semantic feature production norms for a large set of living and nonliving things. *Behavior research methods*, 37(4):547–559, 2005.

Ken McRae, Saman Khalkhali, and Mary Hare. Semantic and associative relations in adolescents and young adults: Examining a tenuous dichotomy. *The Adolescent Brain: Learning, Reasoning, and Decision Making*, 2012.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Adam Meyers, Ruth Reeves, Catherine Macleod, Rachel Szekely, Veronika Zielinska, Brian Young, and Ralph Grishman. Annotating noun argument structure for nombank. In *Proceedings of LREC-2004*, 2004.

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

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed Representations of Words and Phrases and Their Compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013b.

David Mimno and Andrew McCallum. Topic models conditioned on arbitrary features

with dirichlet multinomial regression. In *In Uncertainty in Artificial Intelligence*. Citeseer, 2008.

David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *EMNLP*, 2011.

John Paul Minda and J David Smith. Comparing prototype-based and exemplar-based accounts of category learning and attentional allocation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28(2):275, 2002.

Tom Minka. Discriminative models, not discriminative training, 2005.

Einat Minkov and Luke Zettlemoyer. Discriminative learning for joint template filling. In *ACL*, 2012.

Marvin Minsky. A framework for representing knowledge. MIT-AI Laboratory Memo 306, June 1974.

Ashutosh Modi. Event embeddings for semantic script modeling. In *CoNLL*, pages 75–83, 2016.

Ashutosh Modi and Ivan Titov. Inducing neural models of script knowledge. In *CoNLL*, Ann Arbor, Michigan, June 2014. URL `http://www.aclweb.org/anthology/W14-1606`.

Ashutosh Modi, Ivan Titov, and Alexandre Klementiev. Unsupervised induction of

frame-semantic representations. In *NAACL Workshop on the Induction of Linguistic Structure*, 2012.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California, June 2016. Association for Computational Linguistics.

Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Chapman & Hall/CRC, 2010.

Hatem Mousselly-Sergieh and Iryna Gurevych. Enriching wikidata with frame semantics. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 29–34, San Diego, CA, June 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W16-1306`.

Stephen Muggleton and Luc de Raedt. Inductive logic programming: Theory and methods. *The Journal of Logic Programming*, 19.20, Supplement 1(0):629 – 679, 1994.

Jason Naradowsky, Sebastian Riedel, and David A Smith. Improving NLP through Marginalization of Hidden Syntactic Structure. In *Proceedings of the 2012 Joint*

Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 810–820. Association for Computational Linguistics, 2012.

Srini Narayanan. Bridging text and knowledge with frames. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, pages 22–25, Baltimore, MD, USA, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W14-3006`.

Radford M Neal. Slice sampling. *Annals of statistics*, pages 705–741, 2003.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108, Los Angeles, California, June 2010. Association for Computational Linguistics.

Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. Generative event schema induction with entity disambiguation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 188–197, Beijing, China, July 2015. Association for Computational Linguistics.

BIBLIOGRAPHY

Siegfried Nijssen and Joost N. Kok. Efficient frequent query discovery in FARMER. In *PKDD*, pages 350–362, 2003.

Benjamin Nye and Ani Nenkova. Identification and characterization of newsworthy verbs in world news. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1440–1445, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N15-1166`.

John Walker Orr, Prasad Tadepalli, Janardhan Rao Doppa, Xiaoli Fern, and Thomas G Dietterich. Learning scripts as hidden markov models. In *AAAI*, 2014.

Paul Over and James Yen. Introduction to DUC-2004: An intrinsic evaluation of generic news text summarization systems. In *Proceedings of DUC 2004 Document Understanding Workshop, Boston*, 2004.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106, 2005.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English Gigaword Fifth Edition LDC2011T07. Web Download, Philadelphia: Linguistic Data Consortium, 2011.

Terence Parsons. *Events in the Semantics of English*, volume 5. Cambridge, MA: MIT Press, 1990.

Siddharth Patwardhan and Ellen Riloff. A unified model of phrasal and sentential evidence for information extraction. In *EMNLP*, 2009.

Michael Paul and Mark Dredze. Factorial lda: Sparse multi-dimensional text models. In *Advances in Neural Information Processing Systems*, pages 2582–2590, 2012.

Michael John Paul. *Topic Modeling with Structured Priors for Text-Driven Science*. PhD thesis, Johns Hopkins University, 2015.

Ellie Pavlick, Travis Wolfe, Pushpendre Rastogi, Chris Callison-Burch, Mark Dredze, and Benjamin Van Durme. Framenet+: Fast paraphrastic tripling of framenet. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 408–413, Beijing, China, July 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-2067`.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

Haoruo Peng and Dan Roth. Two discourse driven language models for semantics. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 290–300, Berlin, Germany, August 2016. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1028`.

BIBLIOGRAPHY

Nanyun Peng, Francis Ferraro, Mo Yu, Nicholas Andrews, Jay DeYoung, Max Thomas, Matthew R. Gormley, Travis Wolfe, Craig Harman, Benjamin Van Durme, and Mark Dredze. A Concrete Chinese NLP Pipeline. In *NAACL: Demo Session*, 2015.

Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D14-1162`.

Slav Petrov. *Coarse-to-fine natural language processing*. Springer Science & Business Media, 2011.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of ACL-ICCL*, pages 433–440, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. doi: http://dx.doi.org/10.3115/1220175.1220230. URL `http://dx.doi.org/10.3115/1220175.1220230`.

Slav Petrov, Dipanjan Das, and Ryan McDonald. A universal part-of-speech tagset. In *ArXiv*, April 2011.

Miriam R. L. Petruck and Gerard de Melo, editors. *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*. Association for

Computational Linguistics, Baltimore, MD, USA, June 2014. URL http://www.aclweb.org/anthology/W14-30.

Karl Pichotta and Raymond J Mooney. Statistical script learning with multi-argument events. In *EACL*, 2014.

Karl Pichotta and Raymond J Mooney. Learning statistical scripts with LSTM recurrent neural networks. In *AAAI*, 2016.

Carl Pollard and Ivan A Sag. *Head-driven phrase structure grammar*. University of Chicago Press, 1994.

Hoifung Poon, Chris Quirk, Charlie DeZiel, and David Heckerman. Literome: Pubmed-scale genomic knowledge base in the cloud. *Bioinformatics*, 30(19):2840–2842, 2014.

Matt Post and Daniel Gildea. Bayesian learning of a tree substitution grammar. In *Proceedings of ACL-IJCNLP (short papers)*, pages 45–48, Stroudsburg, PA, USA, 2009a. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1667583.1667599.

Matt Post and Daniel Gildea. Language modeling with tree substitution grammars. In *NIPS Workshop on Grammar Induction, Representation of Language, and Language Learning*, 2009b.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. The Penn Discourse TreeBank 2.0. In *LREC*, 2008.

James Pustejovsky, Patrick Hanks, and Anna Rumshisky. Automated induction of sense in context. In *Proceedings of the International Conference on Computational Linguistics*, 2004.

Altaf Rahman and Vincent Ng. Supervised models for coreference resolution. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 968–977, Singapore, August 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D/D09/D09-1101`.

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*, pages 248–256, Singapore, August 2009. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D/D09/D09-1026`.

Rajesh Ranganath, Sean Gerrish, and David Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

Pushpendre Rastogi and Benjamin Van Durme. Augmenting framenet via PPDB. In *Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 1–5, Baltimore, Maryland, USA, June 2014. Asso-

ciation for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-2901.

Pushpendre Rastogi, Benjamin Van Durme, and Raman Arora. Multiview LSA: Representation Learning via Generalized CCA. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 556–566, Denver, Colorado, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1058.

Michaela Regneri, Alexander Koller, and Manfred Pinkal. Learning script knowledge with web experiments. In *ACL*, 2010.

Roi Reichart and Regina Barzilay. Multi event extraction guided by global constraints. In *NAACL*, 2012.

Drew Reisinger, Rachel Rudinger, Francis Ferraro, Craig Harman, Kyle Rawlins, and Benjamin Van Durme. Semantic proto-roles. *Transactions of the Association for Computational Linguistics (TACL)*, 3:475–488, 2015. ISSN 2307-387X.

Danilo J Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1278–1286, 2014.

BIBLIOGRAPHY

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.

Ronald Rosenfeld. *Adaptive statistical language modeling: A maximum entropy approach*. PhD thesis, Computer Science Department, Carnegie Mellon University, 1994.

Ronald Rosenfeld. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech & Language*, 10(3):187–228, 1996.

Sascha Rothe and Hinrich Schütze. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1793–1803, Beijing, China, July 2015. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-1173`.

Rachel Rudinger and Benjamin Van Durme. Is the Stanford Dependency Representation Semantic? In *ACL Workshop on EVENTS*, Baltimore, Maryland, USA, June 2014. URL `http://www.aclweb.org/anthology/W14-2908`.

BIBLIOGRAPHY

Rachel Rudinger, Pushpendre Rastogi, Francis Ferraro, and Benjamin Van Durme. Script induction as language modeling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1681–1686, Lisbon, Portugal, September 2015. Association for Computational Linguistics. URL `http://aclweb.org/anthology/D15-1195`.

Donald Rumelhart. Schemata: The building blocks of cognition. In *Theoretical Issues in Reading Comprehension*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1980.

Josef Ruppenhofer, Michael Ellsworth, Miriam R.L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. *FrameNet II: Extended Theory and Practice*. ICSI, Berkeley, California, 2006.

Stuart Russell and Peter Norvig. *Artificial Intelligence: a Modern Approach*. Prentice hall Upper Saddle River, NJ, 2010.

Evan Sandhaus. The New York Times Annotated Corpus LDC2008T19. Web Download, Philadelphia: Linguistic Data Consortium, 2008.

Federico Sangati and Willem Zuidema. Accurate parsing with compact tree-substitution grammars: Double-dop. In *EMNLP*, 2011.

Beatrice Santorini. Part-of-speech tagging guidelines for the Penn Treebank Project (3rd revision). Technical report, University of Pennsylvania, 1990.

BIBLIOGRAPHY

Masa-Aki Sato. Online model selection based on the variational bayes. *Neural Computation*, 13(7):1649–1681, 2001.

Roger C. Schank. Using knowledge to understand. In *Proceedings of the 1975 workshop on Theoretical issues in natural language processing (TINLAP)*, 1975.

Roger C Schank and RP Abelson. Scripts. *Plans, Goals, and Understanding. Lawrence, Erlbaum, Hillsdale*, 1977.

Jason Schlachter, David Van Brackle, Luis Asencios Reynoso, James Starz, and Nathanael Chambers. Evaluating automatic learning of structure for event extraction. In *Advances in Cross-Cultural Decision Making*, pages 145–158. Springer, 2017.

L. Schubert. Can we derive general world knowledge from texts? In *Proceedings of the second international conference on Human Language Technology Research*, pages 94–97, 2002.

Lenhart K Schubert. The situations we talk about. In *Logic-based artificial intelligence*. Springer, 2000.

Lenhart K Schubert and Chung Hee Hwang. Episodic logic meets little red riding hood: A comprehensive, natural representation for language understanding. *NLP & KR*, 2000.

BIBLIOGRAPHY

Karin Kipper Schuler. *VerbNet: A broad-coverage, comprehensive verb lexicon*. PhD thesis, University of Pennsylvania, 2005.

Lei Sha, Sujian Li, Baobao Chang, and Zhifang Sui. Joint learning templates and slots for event schema induction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 428–434, San Diego, California, June 2016. Association for Computational Linguistics.

Hiroyuki Shindo, Yusuke Miyao, Akinori Fujino, and Masaaki Nagata. Bayesian symbol-refined tree substitution grammars for syntactic parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–448, Jeju Island, Korea, July 2012. Association for Computational Linguistics.

David A. Smith and Jason Eisner. Minimum Risk Annealing for Training Log-Linear Models. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 787–794, Sydney, Australia, July 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P06/P06-2101`.

Mehdi Soufifar, Marcel Kockmann, Lukáš Burget, Oldřich Plchot, Ondřej Glembek, and Torbjørn Svendsen. iVector approach to phonotactic language recognition. In *Interspeech*, pages 2913–2916, 2011.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou,

and Jun'ichi Tsujii. BRAT: a Web-Based Ttool for NLP-assisted Text Annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics, 2012.

Stephanie M. Strassel, Ann Bies, and Jennifer Tracey. Situational awareness for low resource languages: the LORELEI situation frame annotation task. In *Proceedings of the First workshop on: Exploitation of Social Media for Emergency Relief and Preparedness (SMERP)*, 2017.

Beth Sundheim. Proceedings of the fourth message understanding conference (MUC-4), 1992.

Beth Sundheim. Overview of results of the MUC-6 evaluation. In *Proceedings of a Workshop held at Vienna, Virginia: May 6-8, 1996*, 1996.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. The conll 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, Manchester, England, August 2008. Coling 2008 Organizing Committee. URL `http://www.aclweb.org/anthology/W08-2121`.

Matt Taddy. On estimation and selection for topic models. In Neil D. Lawrence and Mark Girolami, editors, *Proceedings of the Fifteenth International Conference on*

Artificial Intelligence and Statistics, volume 22 of *Proceedings of Machine Learning Research*, pages 1184–1193, La Palma, Canary Islands, 21–23 Apr 2012. PMLR. URL `http://proceedings.mlr.press/v22/taddy12.html`.

Adam Teichert, Adam Poliak, Benjamin Van Durme, and Matthew Gormley. Semantic proto-role labeling, 2017.

Ivan Titov and Ehsan Khoddam. Unsupervised induction of semantic roles within a reconstruction-error minimization framework. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1–10, Denver, Colorado, May–June 2015. Association for Computational Linguistics.

Ivan Titov and Alexandre Klementiev. A bayesian model for unsupervised semantic parsing. In *ACL*, 2011.

Tom Trabasso and Linda L Sperry. Causal relatedness and importance of story events. *Journal of Memory and Language*, 24(5):595–611, 1985.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. Evaluation of word vector representations by subspace alignment. In *Proc. of EMNLP*, 2015.

Peter D Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, 37:141–188, 2010.

BIBLIOGRAPHY

Laurens van der Maaten and Geoffrey Hinton. Visualizing Data Using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Benjamin Van Durme and Daniel Gildea. Topic models for corpus-centric knowledge generalization. Technical report, University of Rochester, 2009.

Benjamin Van Durme and Ashwin Lall. Streaming pointwise mutual information. In *NIPS*, 2009.

Benjamin Van Durme and Lenhart Schubert. Open knowledge extraction through compositional language processing. In *Proceedings of the 2008 Conference on Semantics in Text Processing (STEP)*, pages 239–254. Association for Computational Linguistics, 2008.

Remi van Trijp, Luc Steels, Katrien Beuls, and Pieter Wellens. Fluid construction grammar: The new kid on the block. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 63–68, Avignon, France, April 2012. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/E12-2013`.

David P Vinson and Gabriella Vigliocco. Semantic feature production norms for a large set of objects and events. *Behavior Research Methods*, 40(1):183–190, 2008.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. ACE

BIBLIOGRAPHY

2005 Multilingual Training Corpus LDC2006T06. DVD. Philadelphia: Linguistic Data Consortium, 2006.

Hanna M Wallach. *Structured topic models for language*. PhD thesis, University of Cambridge, 2008.

Chong Wang and David M Blei. Variational Inference in Nonconjugate Models. *The Journal of Machine Learning Research*, 14(1):1005–1031, 2013.

William Yang Wang and Diyi Yang. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *EMNLP*, 2015.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. Ontonotes release 5.0 LDC2013T19. Linguistic Data Consortium, Philadelphia, PA, 2013.

Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. Universal decompositional semantics on universal dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1713–1723, Austin, Texas, November 2016. Association for Computational Linguistics. URL `https://aclweb.org/anthology/D16-1177`.

BIBLIOGRAPHY

Mann William and Sandra Thompson. Rhetorical Structure Theory: Towards a Functional Theory of Text Organization. *Text*, 8(3):243–281, 1988.

Philip Wolfe. Convergence conditions for ascent methods. *SIAM review*, 11(2):226–235, 1969.

Philip Wolfe. Convergence conditions for ascent methods. ii: Some corrections. *SIAM review*, 13(2):185–188, 1971.

Travis Wolfe. Personal communication, April 2017.

Travis Wolfe, Mark Dredze, and Benjamin Van Durme. A study of imitation learning methods for semantic role labeling. In *Proceedings of the Workshop on Structured Prediction for NLP*, pages 44–53, Austin, TX, November 2016. Association for Computational Linguistics. URL `http://aclweb.org/anthology/W16-5905`.

Mo Yu and Mark Dredze. Improving Lexical Embeddings with Semantic Knowledge. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 545–550, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P14-2089`.

John M Zelle and Raymond J Mooney. Inducing deterministic prolog parsers from treebanks: A machine learning approach. In *AAAI*, pages 748–753, 1994.

BIBLIOGRAPHY

Ke Zhai and Jordan L Boyd-Graber. Online latent dirichlet allocation with infinite vocabulary. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 561–569, 2013.

Andreas Zollmann and Khalil Sima'an. A consistent and efficient estimator for data-oriented parsing. *Journal of Automata Languages and Combinatorics*, 10(2/3):367, 2005.

Willem Zuidema. Parsimonious data-oriented parsing. In *Proceedings of EMNLP-CoNLL*, pages 551–560, 2007.

Vita

Francis Ferraro, born in Schenectady, New York, USA in 1989, earned an honors B. S. degree in computer science, a B. S. degree in mathematics, and a minor in linguistics from the University of Rochester in 2011. He was inducted into Phi Beta Kappa in 2010, was named a finalist for the Computing Research Association's Outstanding Undergraduate Research award (2010-2011), and received offers for a National Defense Science and Engineering Fellowship and a National Science Foundation Graduate Research Fellowship, accepting the latter. Starting in Fall 2017, Frank will join the faculty of University of Maryland Baltimore County (UMBC), as an assistant professor of computer science.