Learning Frames from Text with an Unsupervised Latent Variable Model

Brendan O'Connor
Machine Learning Department, Carnegie Mellon University
Data Analysis Project

October 25, 2012

Abstract

We develop a probabilistic latent-variable model to discover semantic frames—types of events or relations and their participants—from corpora. Our key contribution is a model in which (1) frames are latent categories that explain the linking of verb-subject-object triples in a given document context; and (2) cross-cutting semantic word classes are learned, shared across frames. We also introduce an evaluation methodology that compares to FrameNet, interpreting the learned model as an alternative frame lexicon.

1 Introduction

Semantic frames—types of events or relations and their participants—are a key development in linguistic theories of semantics (Fillmore, 1982) and, sometimes called "scripts" or "schemata," have figured heavily in natural language understanding research (Schank and Abelson, 1977; Lehnert and Ringle, 1982). There has been a recent surge in interest in finding frames in text data, including: frame-semantic parsing (Das et al., 2010; Gildea and Jurafsky, 2002) following the conventions of FrameNet (Fillmore and Baker, 2001), and discovery of narrative structure (Chambers and Jurafsky, 2009).

In this paper, we seek to discover semantic frames—types of events or relations and their participants—from corpora, using probabilistic latent-variable models. This approach focuses on verbs with their subjects and objects and is inspired by models of selectional preferences and argument structure. Building on the framework of topic models (Blei et al., 2003), we further leverage document context, exploiting an assumption that relatively few frames are expected to be present in a single document. Our key contributions are in a new model in which (1) frames are latent categories that explain the linking of verb-subject-object triples in a given document context; and (2) cross-cutting semantic word classes are learned, shared across frames.

Because there are many ways to define frames, we believe a data-driven approach that does not require human annotation is attractive, especially when considering a new domain of text, or exploratory data analysis of a corpus. We explore models built on a range of datasets, highlighting differences in what kinds of frames are discovered.

We also seek to evaluate what is learned by comparing to existing lexical resources, introducing a novel evaluation methodology that compares to FrameNet, interpreting the model posterior as an alternative lexicon.

We begin by describing our models and relating them to models in the literature (§2). We discuss inference in §3 and experiments in §4, concluding with example results (§5) and FrameNet comparison (§6).

2 Models

Verbs, subjects, and objects constitute a basic syntactic encoding of actions, events, and their participants. We are interested in a modeling a dataset of document-VSO tuples,

$$(\text{DocID}, w^{(verb)}, w^{(subj)}, w^{(obj)})$$

We present two models to capture document and syntactic contextual information in the generation of text.

2.1 "Model 0": Independent tuples

Previous work in model-based syntactic distributional clustering, usually aimed at modeling selection preferences, has modeled syntactic tuples as independent (or rather, conditionally independent given the model parameters). Pereira et al. (1993) and Rooth et al. (1999) model a corpus of (verb, object) pairs with a latent variable for each tuple, and different word distributions for for each argument and class. (Rooth et al. experiment with different syntactic relations, but always use pairs; e.g. (verb, subject).)

To situate our model, we slightly generalize these approaches to (verb, subject, object) triples, and add symmetric Dirichlet priors, as follows. $\phi_f^{(arg)}$ denotes a word multinomial for argument type arg and frame f; and there are three argument types (verb, subject, object), treated completely separately.

- **Frame lexicon:** Dirichlet prior β. For each $f = 1..F$, sample three word multinomials: $\phi_f^{(v)}, \phi_f^{(s)}, \phi_f^{(o)} \sim Dir(\beta)$
- **Tuple data:** For each tuple $i = 1..N$,
  - Draw its frame indicator f_i (from a fixed prior),
  - Draw the three words from their repsective multinomials: $w_i^{(v)} \sim \phi_{f_i}^{(v)};\ w_i^{(s)} \sim \phi_{f_i}^{(s)};\ w_i^{(o)} \sim \phi_{f_i}^{(s)}$

This approach models every tuple independently, with no document or other context; the only thing shared across tuples are the per-class argument word distributions. More recent work extends the Rooth approach to model other syntactic relation pairs (Séaghdha, 2010) and web-extracted triple relations (Ritter and Etzioni, 2010). These works make several variations to the probabilistic directed graph, giving the verb a more central role; we instead stick with Rooth's symmetric setup.

2.2 Model 1: Document-tuples

We would like to use document context to constrain the selection of frames. Following the intuition of latent Dirichlet allocation (Blei et al., 2003)—that each particular document tends to use a small subset of available latent semantic factors ("topics")—we propose that a document's frames are similarly drawn from a sparsity-inducing Dirichlet prior. Our document-tuple model uses the same frame lexicon setup as above, but enriches the document generation:

- F frames, and Dirichlet priors α, β
- **Frame lexicon:** For each frame $f \in 1..F$, and argument position $a \in \{1, 2, 3\}$,
  - Draw word multinomial $\phi_f^{(a)} \sim Dir(\beta)$
- **Document-tuple data:** For each document $d \in 1..D$,
  - Draw frame multinomial $\theta_d \sim Dir(\alpha)$

- For each tuple i in the document,
  - Draw frame indicator $f_i \sim \theta_d$
  - Draw word triple: for each argument position $a \in \{1, 2, 3\}$,
    - Draw $w_i^{(a)} \sim \phi_{f_i}^{(a)}$

Note that in the limiting case as $\alpha \to \infty$, Model 1 collapses into the independent tuple model, where document context is irrelevant. In our experiments, we fit α with posterior inference, and it prefers to have relatively low values, giving orders of magnitude better likelihood than a high α—implying that the document-level sparsity assumption better explains the data than an independent tuple hypothesis.

From one perspective, LDA's "topics" have been renamed "frames." Computationally, this is a minor difference, but from an NLP persepctive, is very important, since we are asking the latent variable to do something else than LDA has it do—it now models syntactic argument selection as well as document-level effects.

The document-level mixing is potentially useful for applications, because it gives a hook into a vast literature of topic models that jointly model text with many types of document-level metadata such as time, space, arbitrary metadata, etc. (e.g. Blei and Lafferty (2006); Eisenstein et al. (2010); Mimno and McCallum (2008)) In this respect, this model could be seen as a "semantic" or "syntactic tuple" topic model, along the lines of previous work that has added various forms of sentence-internal structure to LDA's document generation process, such as bigrams (Wallach, 2006), HMM's (Griffiths et al., 2005), or certain forms of syntax (Boyd-Graber and Blei, 2008).

2.3 Model 2: Cross-cutting semantic classes

The frames in Model 1 share no word statistics with one another, so each has to relearn lexical classes for its arguments. We address this by introducing a latent word class variables c for every token. Every frame has preferences for different classes at different argument positions, so word classes can be shared across frames.

The Model 2 generative process is:

- C word classes, F frames, and Dirichlet priors $\alpha, \beta, \gamma_1, \gamma_2, \gamma_3$
- **Frame lexicon:**
  - For each class $c \in 1..C$,
    - Draw word multinomial $\phi_c \sim Dir(\beta)$
  - For each frame $f \in 1..F$, and argument position $a \in \{1, 2, 3\}$,
    - Draw the "linker" $L_{f,a} \sim Dir(\gamma)$, a multinomial over word classes: $L_{f,a} \in Simplex(C)$.
- **Document-tuple data:**
  - For each document $d \in 1..D$, draw frame multinomial $\theta_d \sim Dir(\alpha)$
  - For each tuple i for document d,
    - Draw frame indicator $f_i \sim \theta_d$
    - Draw word triple: for each argument $a \in \{1, 2, 3\}$,
      - (if a is null in this tuple, skip)
      - Draw class $c^{(a)} \sim L_{f_i,a}$
      - Draw word $w^{(a)} \sim \phi_{c^{(a)}}$

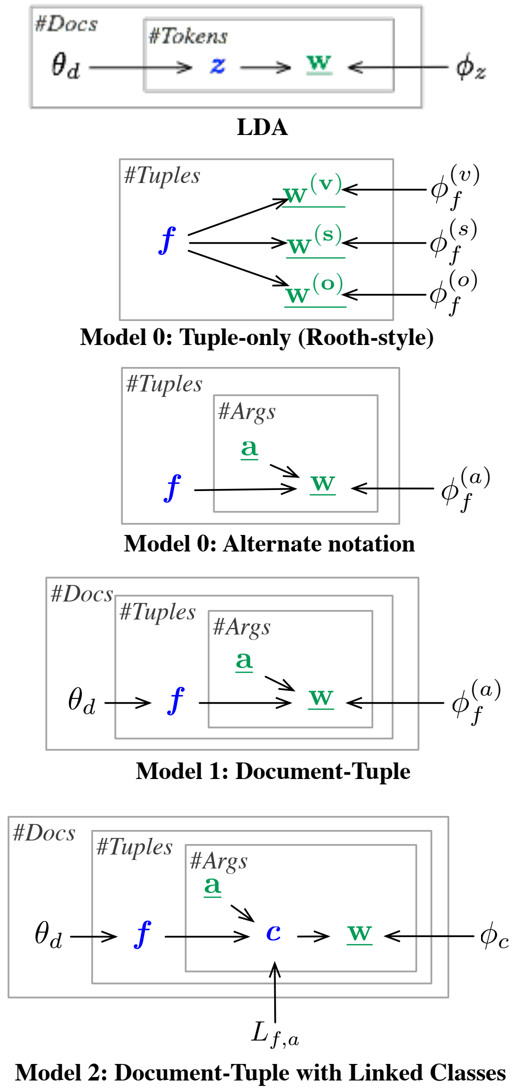


Figure 1: Probabilistic directed graphs for prior work and our models. Dirichlet priors are omitted for brevity. **Blue** variables are latent, and resampled through collapsed Gibbs sampling; **green** variables are observed.

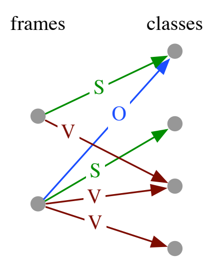


Figure 2: Example fragment of the sparse linking array $L_{f,a,c}$ as a labeled bipartite graph. For example, the S-labeled edge connecting frame 1 and class 1 indicates $L_{1,2,1} > 0$ (assuming subject corresponds to $a = 2$).

Central to this model is the "linker" $L_{f,a,c}$, a multidimensional array of dimensions $(F, 3, C)$, which says which word classes are likely for a given frame-argument combination. We show a schematic diagram in Figure 2, where edges represent high (or just significantly nonzero) probabilities. A word class that is a subject for one frame may be an object for another. A frame may have multiple classes for an argument position, though in practice the number is relatively small, due to the Dirichlet prior (and, as we will see, it naturally turns out sparse when the prior is fit to the data).

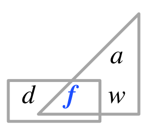
Note that Model 1 can be seen as a version of Model 2 with a deterministic linker: every (f, a) pair is bound to one single, unique class; i.e. a point mass at $c = 3(f-1) + a - 1$. This further corresponds to a perfectly sparse stick-breaking prior on $L_{f,a}$, suggesting non-parametric extensions of Model 2 for future work.

Titov and Klementiev (2011)'s model of semantic frames also uses cross-cutting word classes, embedded in a more complex model that also learns clusters of syntactic relations, and recursively generates dependency trees. Grenager and Manning (2006) and Lang and Lapata (2010) present related models for unsupervised PropBank-style semantic role labeling, where a major focus is grouping or clustering syntactic argument patterns.

Finally, while we do not enforce any relationship between syntactic argument position and the classes, in practice, most classes are exclusively either verbs or nouns, since words that can be verbs often cannot appear as nouns.

3 Inference

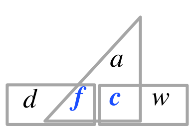
Through Dirichlet-multinomial conjugacy, we can use collapsed Gibbs sampling for inference (Griffiths and Steyvers, 2004; Neal, 1992). The Gibbs sampling equations are instructive. For Model 1, it is:



$$p(\mathbf{f}|dwa) \propto p(\mathbf{f}|d) \prod_a p(w^{(a)}|\mathbf{f}a)$$

This diagram of CPT factors shows the two soft constraints the Gibbs sampler works to satisfy: the left term $p(f|d)$ tries to ensure document-frame coherency—it exerts pressure to select a frame used elsewhere in the document. The second term $p(w|fa)$ exerts pressure for syntactic coherency—to choose a frame that has compatibility with all the syntactic arguments. Thus Model 1 combines selectional preferences with document modeling.

Model 2's Gibbs sampling equations are



$$p(\mathbf{f}|dca) \propto p(\mathbf{f}|d) \prod_a p(c^{(a)}|\mathbf{f}a)$$
$$p(\mathbf{c^{(a)}}|faw^{(a)}) \propto p(\mathbf{c^{(a)}}|fa)p(w^{(a)}|\mathbf{c^{(a)}})$$

In these factor diagrams, the boxes correspond to the maximal dimensional count tables the sampler has to maintain; for example, the tables $C(d, f)$ and $C(f, a, w)$ for Model 1. They are only the maximal, not all, count tables, since some rollups also have to be maintained for CGS denominators; e.g. $C(d)$ and $C(f, a)$ here.

We resample the symmetric Dirichlet priors $\alpha, \beta, \gamma_1, \gamma_2, \gamma_3$ with slice sampling (Neal, 2003) every 100 iterations, using a vague hyperprior. We found the choice of hyperprior (either vague gamma or improper uniform) made little difference.

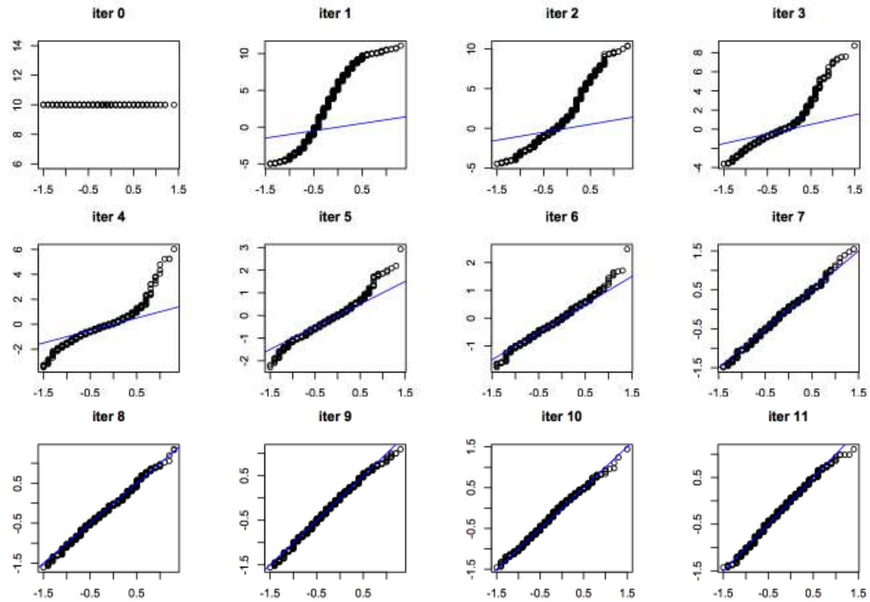


Figure 3: QQ-plots of 500 slice sampling chains, showing their distribution converging to the true posterior. Each plot is the QQ-plot of 500 different chain states, all at the same iteration, against the exhaustively calculated posterior.

The slice sampler was run for 10 iterations for each parameter, a number chosen based on a simulation experiment: from a fixed α, we generated a Dirichlet-multinomial dataset (10 groups, 10 observations each), then ran 500 independent slice sampling chains for the variable $\log(\alpha)$, assessing the cross-chain distribution of states at a single timestep against the true posterior (the latter computed with near-exact grid approximation on the DM likelihood) via QQ-plots shown in Figure 3. MCMC theory says that after burn-in you can stop the chain to have a single independent draw from the posterior, which implies these QQ-plots will become flat; it appears 7 iterations was enough in this experiment. We chose here a very bad initializer $\alpha = e^{10}$, compared to the true MAP $e^{0.9}$; a good initialization at $\alpha = e^{1}$ had posterior convergence after just 2 iterations. (This analysis technique was inspired by Cook et al. (2006)'s Bayesian software validation method.)

The hyperparameter sampling substantially improves likelihood. Interestingly, most of the movement tends to happen early in the MCMC chain, then the hyperparameter stabilizes as the rest of the model is still moving. For one experimental setting (a subset of CRIMENYT, described below), we checked if the outcome was initializer dependent by starting three different MCMC chains that were identical except for three different α initializers: Figure 4. Reassuringly, they all converged on the same region of values. This robustness to initialization was exactly what we wanted.

For larger datasets, we implemented a parallelized sampling scheme for f and c similar to Newman et al. (2009) where individual processors use stale counts and synchronize once per iteration by sending count update messages to all other processors.[1]

In experiments, we ran the Gibbs sampler for at least 5,000 iterations, and up to 20,000 as time permitted.

[1] We use from 12 to 64 CPU cores in some cases. Implementation is in Python/C/MPI.

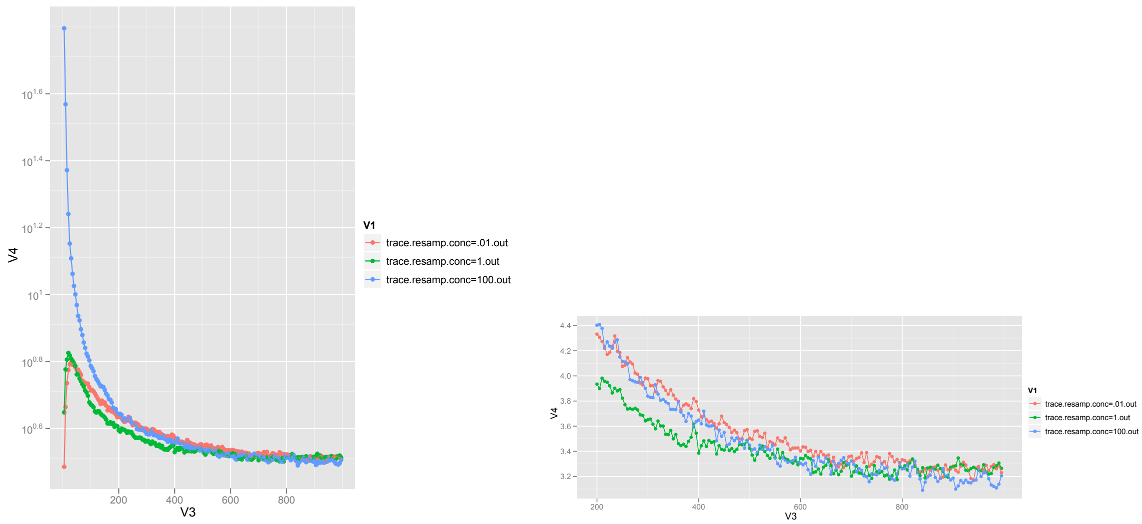


Figure 4: α parameter values over time, when being resampled, from three very different initial positions $\alpha = 0.01,\ 1,\ 100$. The left plot shows the entire history; the right shows from iteration 200 until the end. This is Model 1 on a subset of the CRIMENYT corpus.

4 Experiments

4.1 Datasets

We use datasets from the New York Times, Wall Street Journal, and the Brown corpus for experiments. The New York Times Annotated Corpus (Sandhaus, 2008) contains articles from 1987 to 2007 that have been extensively categorized by hand, each having multiple labels. We use it to extract two subcorpora. First, inspired by examples in the crime reporting domain of extracting narrative event structures (Chambers and Jurafsky, 2009) and work on the FrameNet corpus (Fillmore and Baker, 2001), we select news articles for which any category labels contain any of the words *crime*, *crimes*, or *criminal*, resulting in a targeted corpus of articles (CRIMENYT). We wanted to see if the model can learn specific types of actions and noun classes for that domain. Second, we take a uniformly drawn subsample of all articles (UNIFNYT).

The NYT article texts are processed with the Stanford CoreNLP software[2] using default settings; we use its sentence segmentation, tokenization, part-of-speech tagging, lemmatization, parsing, and dependency conversion (the *CCprocessed* version).

Finally, we also performed experiments on two pre-parsed corpora from the Penn Treebank (Marcus et al., 1994), cointaining tokenizations, part-of-speech tags, and parses: The Wall Street Journal (all sections; WSJPTB), and the PTB's subset of the Brown corpus—consisting mainly of literature and essays (BROWNPTB). We used the Stanford software to convert PTB constituent structures to dependencies and produce lemmatizations. These are substantially smaller than the NYT datasets; see Table 1.

From the dependencies, we extract active voice verb-subject-object tuples of form $(w^{(v)}, w^{(s)}, w^{(o)})$, $(w^{(v)}, w^{(s)}, null)$, or $(w^{(v)}, null, w^{(o)})$, by finding instances of a verb word $w^{(v)}$ with POS tag beginning with VB, and having a noun dependent with relation *nsubj* or *dobj*. If both an *nsubj* and *dobj* child exist, take them both to form a full VSO triple; otherwise make a VS_ or V_O pair. (Most are incomplete: in CRIMENYT, 19% of tuples are VSO, while 43% are VS_ and 38% are V_O.) If multiple *nsubj*'s exist, we arbitrarily take only one; similarly with *dobj*. (It may be better to take all possible tuples in this situation.)

[2] http://nlp.stanford.edu/software/corenlp.shtml

Corpus	#Docs	#Sent	#Tok	#VSO
CRIMENYT	27k	789k	20M	1.3M
UNIFNYT	60k	2.4M	14M	5.3M
WSJPTB	2,312	49k	1.2M	78k
BROWNPTB	192	24k	459k	27k

Table 1: Datasets used in experiments. The number of documents and VSO tuples are relevant to the model; the number of sentences and original tokens are not. The tuple count includes partial tuples (missing either subject or object).

correct?	text and VSO tuple
RIGHT: for "workers," only use single head word of the noun phrase	In less than an hour , the police and rescue unit (workers $)_{subj}$ [found $]_{verb}$ the (organ $)_{obj}$ in the tall grass of the field , packed it in ice and took it to the hospital .
RIGHT	Mrs. (Bissell $)_{subj}$, she said , never [learned $]_{verb}$ what her husband did with the money .
INCOMPLETE: "he" should be subject of "defrauded" since SD uses content verb as head	Asked why he had [defrauded $]_{verb}$ the insurance (company $)_{obj}$ and been partners with the mob . . .
WRONG: lists are a known problem case for current parsers	Guilty on Five Charges Mr. Garcia was found guilty on all five charges against him : (theft $)_{subj}$ of heroin , possession with intent to [distribute $]_{verb}$ heroin , narcotics conspiracy and two (counts $)_{obj}$ of money laundering .

Table 2: Example extracted tuples and our annotations in their original sentences (in tokenized form).

We performed error analysis of the parser and syntactic extraction, by selecting a random sample of extracted V-S-O tuples from the CRIMENYT corpus to manually assess for accuracy. A subset are shown in Table 2. We annotated 40 tuples, in context in their respective sentences, consulting the Stanford Dependency papers (De Marneffe et al., 2006; de Marneffe and Manning, 2008), which have clear linguistic definitions of the grammatical relations, their conventions for analyzing compound verbs, etc. Out of 40 tuples, we found 30 had the subject and/or object arguments correct; 6 had one or both wrong; and 4 were incomplete (missing either subject or object)—75% precision.

5 Example Results

Here we focus on inspecting the posterior results from Model 2 on the CRIMENYT dataset, with 400 frames and classes.

Table 3 shows one particular interesting frame, which we intepret as "legislation." We show its argument linking patterns by listing, for each argument position, the most common classes appearing there. These linking probabilities ($L_{f,a,c} = p(c|fa)$) are show in the left column—the linkers are quite sparse, with most of the probability mass contained in the shown classes.

For each class, we show the most common words. Interestingly, the frame distinguishes different types of actions and actors where laws or procedures are the object. In fact, it nearly describes a sequential "script"

f=286: *Legislation*		
$a, c,$ $p(c\|fa)$	*top words*	*interp.*
(v) 242 (0.45)	pass have enact impose adopt extend eliminate increase toughen abolish amend need use establish change fail strike consider restore ease	passage/enactment
(v) 34 (0.21)	violate change break enforce follow use challenge practice write obey adopt impose revise draft apply teach reform amend ignore uphold	enforce-ment and changes
(v) 163 (0.11)	pass support approve oppose vote sign introduce propose veto consider reach include block sponsor favor want reject back push get	political considera-tion
(v) 10 (0.04)	apply permit challenge give interpret bar strike impose limit involve recognize uphold OOV adopt justify regulate seek place define hold	judicial review
(v) 241 (0.04)	have OOV do find take expect see allow use create begin view produce place suffer add like start study face	generic/light verbs
(s) 71 (0.60)	state Congress government system OOV court York city judge law county Jersey Government Legislature State official California legislator country States	jurisdictions
(s) 188 (0.14)	Senate Legislature House Congress Republicans lawmaker Pataki governor Clinton Bush assembly Democrats OOV leader legislator Administration Cuomo Council administration group	legislative and executive branches
(o) 42 (0.80)	law statute rule guideline order system regulation ordinance Constitution curfew Act ban restriction policy code provision legislation requirement agreement limit	kinds/aspects of laws, regulations
(o) 332 (0.09)	penalty discretion parole use number limit period punishment system power OOV approval sale provision test type time judge release protection	(same as above)
(o) 395 (0.06)	bill measure legislation amendment provision proposal law ban version vote package penalty veto Senate agreement issue abortion language passage action	procedural terms

Table 3: Example frame ("Legislation") learned from CRIMENYT.

of actions of the process of creating, enforcing, and modifying laws—but the model has no knowledge of anything sequential. This happens simply because there are several sets of actors that perform actions upon laws, and the model can picks up on this fact; the model can see individual events in such a process, but not the structure of the process itself.

Every word class, of course, can be used multiple times by several different frames. We were curious if the model could find word classes that took the subject position in some frames, but the object position in others—Chambers and Jurafsky (2009) demonstrate interesting examples of this in their learned schemas. Our model does find such instances. Consider class $c = 99$, a "victim" class: {*girl boy student teen-ager daughter victim sister child*}. It appears as an object for a "violence/abuse against victim" frame where the most common verb class has top words {*rape assault attack meet force identify see ask*}, while it is a subject in a frame with a wider variety of generic and communication verbs {*say be go come try admit ask work agree continue refuse*}.

There are many more interesting examples. A sampling of other word classes, with our interpretations, include:

- c=3 (v) stages in a process, esp. criminal process: *serve face receive await stand enter spend complete accept get violate avoid post deny reduce give finish begin grant draw revoke jump*
- c=4 (v) argumentation: *prove demonstrate avoid reflect reach undermine carry mean affect force satisfy*; with the message/argument as subject: (c=67) {*case decision arrest ruling action*} and (c=120) {*issue*

view race theme record kind promise}

- c=16 (v) physical pursuit and apprehension/confrontation: *force approach try flee follow threaten know drag tell escape take find walk hold knock grab order push admit break describe leave climb*
- c=17 (obj) effects: *effect impact role consequence anything implication case value connection importance link interest root basis significance bearing*
- c=19 (obj): verdicts/sentences: *sentence penalty punishment execution term order conviction leniency life fine clemency verdict sentencing factor date death circumstance*
- c=44 (obj) games & pastimes: *role game part OOV basketball card ball football music song baseball sport host piano soccer golf tape politics guitar tennis bottom season*
- c=46 (subj,obj) societal ills: *problem violence crime issue situation activity abuse behavior flow criminal kind fear tide cause corruption spread threat case root type crisis*

To guard against a cherry-picking bias, we include an easy-to-read report of all frames, classes, and linkings online.[3]

Some classes are very general, and some are very specific. One interesting thing to note is that some classes have a more topical, and less syntactically coherent, flavor. For example, c=18: {*film viewer program network movie show producer station audience CBS television camera actor fan*}. It appears often for only one frame, and is split 2:1 between subject and object position. Essentially, the syntactic positioning is being ignored: $p(c|f, a = subj)$ is only twice as likely as $p(c|f, a = obj)$, whereas for most noun classes this ratio is in the hundreds or thousands. This word class functions more like an LDA topic. Is it appropriate to interpret it as a "topic," or does it correspond to entities active in a Fillmorean-style frame of "Television Show"? By leveraging both document and syntactic context, we believe our model uncovers semantics for situations and events.

6 Comparison to FrameNet

We are primarily interested in the quality of our induced frame lexicon. Evaluation is difficult; one automatic measure, held-out likelihood, may not always correlate to subjective semantic coherency (Chang et al., 2009). And while subjective coherency judgments are often collected to evaluate word clusters or argument compatibilities, it is unclear to us exactly what task setup would directly support analyzing frame quality. Our primary goal is to achieve a better understanding of what our model is and is not learning.

We propose a method to compare the similarities and differences between a learned frame model and a pre-existing lexicon. Chambers and Jurafsky (2011b) compare their learned frames to MUC templates in the domain of news reports about terrorist activities. Seeking a resource that is more general, more lexicon-focused, can be used to compare different corpus domains we turn to FrameNet (Fillmore and Baker, 2001; Ruppenhofer et al., 2006), a well-documented lexical resource of actions/situations and their typical participant types. In this section we present an analysis of wordset-to-wordset similarity alignments, that we use to analyze verb clusters.

The downloadable FrameNet 1.5 dataset consists of 1,020 *frames*, each of which is associated with a number of *lexical units*, that, when used, can evoke their respective frames. A lexical unit is essentially a word sense; it is associated with a single frame, and also a *lexeme*, which consists of a word *lemma* and part-of-speech category. In the FrameNet annotations, the frame-evoking word is known as the *target*; some of the frame's roles (called *frame elements*) then are bound to words and phrases in the annotated sentence.

Most of FN's lexical units are verbs, nouns, or adjectives. In this work we focus on the verbs, which in FN's annotated data often take their grammatical subject and object as frame arguments. FN may not

[3] http://brenocon.com/dap/materials/

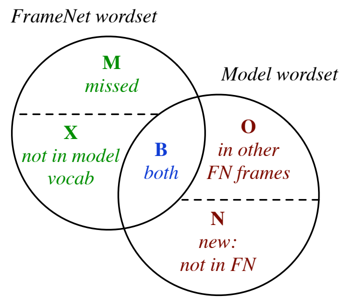


Figure 5: Quantities in a Venn comparison of two wordsets, as used in Table 4. Note "N" actually means it appears fewer than 5 times in the FrameNet annotated data.

have been constructed with this purpose in mind, but by viewing FN as a database of verb clusters and typical argument types, it has a structure comparable to our Model 1, and, to a lesser extent, Model 2. And while FrameNet was constructed to go beyond just verbs—in contrast to another similar frame resource, VerbNet—we find that extracting a verb and argument clusters from FN in this way yields a reasonable-looking dataset.

FN has 11,830 lexical units, 4,605 of which are verbs. We group lexical units by frame, and filter to frames that have 5 or more verbs as lexical units. This leaves 171 frames with 1829 unique verb lemmas. (We ignore multiword lexemes as well as non-verb lexemes.)

6.1 Comparing verb wordsets

How can we analyze the similarity and differences of two different word clusterings? Our notion of "clustering" need not be a proper partition: it merely consists of some number of *wordsets*, which may overlap and not necessarily cover the entire vocabulary. (These are often referred to as "word clusters," but for clarity we always use "wordset.") Many lexical resources and unsupervised models can be discretized and converted into this representation.

We perform a basic analysis, comparing the verb wordsets implied by FrameNet to our model. Verb wordsets (verbsets) are extracted from FrameNet by taking the set of verbs for each frame, so there are 171 verbsets. We discretize our model by taking, for every word class, the words having a count of at least 5 in the Gibbs sample being analyzed. We observed that words with smaller counts than this tended to be unrelated or marginally related to the others—their presence may be due to the randomness inherent in any single Gibbs sample.

The similarity measures are as follows. For two wordsets A and B, the Dice coefficient is

$$DiceSim(A, B) = \frac{2|A \cap B|}{|A| + |B|}$$

This is in fact equivalent to the F1-measure, and monotonic in Jaccard similarity; see the appendix for details. Let A_i be the FrameNet verbset for FN frame i, and B_j be a model's verbset. We compute all $DiceSim(A_i, B_j)$ similarities, and for each FN frame, find the best model match j,

$$\arg\max_j DiceSim(A_i, B_j)$$

and show the best matches in Table 4. This is done with CRIMENYT with 400 frames and classes.

The best match, for FrameNet's "Change position on a scale," clearly has a common semantic theme shared between the FrameNet and model verbsets. The model fails to get several words such as "decline," but does have several other words that seem plausible candidates to add to FrameNet here: "plunge," "sink," "surge."

We show in Table 5 the matches against the different corpora. Interestingly, different corpora are better at recovering different types of frames.

One big issue with this metric is that FrameNet is not designed to have complete lexical coverage for a particular role, so it is unfair to penalize our model for learning novel words not in FrameNet for a particular role. On the other hand, some of the learned new words are sometimes clearly not semantically related. When a model's wordset scores are low, we don't know whether it's because it's actually semantically incoherent, or if FrameNet had poor coverage in its area. It is important to analyze specific instances that make up the quality measure, as in Table 4.

Our method may be more useful as part of a semi-automated system to suggest new additions to a resource like FrameNet; to do this well, it may be interesting to explore building supervision in to an earlier stage of the model, rather than in posthoc analysis as we develop it here. One possibility is to use the data as infomed priors: have assymetric Dirichlet priors with higher values for roles seen in the FrameNet data.

7 Verbs and MUC

Besides FrameNet, it may be worth comparing to a verb clustering more like the Levin (1993) classes; for example, Sun et al. (2008) and Sun and Korhonen (2009) construct a set of 204 verbs in Levin-style clusters and evaluate clustering methods against them.

It would be useful to conduct a thorough evaluation comparing to Chambers and Jurafsky (2011a), which induces frames with several stages of ad-hoc clustering on unlabeled newswire data, and compares its learned frames to frames and extractions from MUC-4, a domain of newswire reports of terrorist and political events. We took initial steps in this direction, including a clean-up of the original data that makes it easier to use, and we have made freely available for futher research.[4] Unfortunately, due to the complexity of MUC data, there is a large amount of ambiguity on how to conduct an evaluation. We uncovered a number of discrepancies between the evaluation done by (Chambers and Jurafsky, 2011a) versus the previous work they compare to (Patwardhan and Riloff, 2007); after a number of email exchanges with all previous co-authors, Chambers modified his evaluation implementation and reports minor changes to their evaluation numbers. We have assembled a document with evaluation methodology clarifications from Chambers, Patwardhan, and Riloff, and posted it online.[5]

We did perform a headroom test for MUC extraction accuracy, by looking at all role-filler instances in the text, and checking how often they corresponded to a nominal subject or object according to the Stanford dependency parser plus our syntactic extraction rules. This was 42% of the time (on the DEV data, an upper bound on recall. Many of the MUC instances (as well as FrameNet annotations) use noun-noun, implicit, and other syntactic indicators of semantic relations. A full generative model would have to accomodate a number of other syntactic paths (and perhaps surface patterns), which could be expanded as an expanded set of a argument type variables—this could be seen as analogous to the clustering over syntactic paths and tags in Chambers and Jurafsky (2011a), or the syntactic path generation in Titov and Klementiev (2011).

8 Conclusion

We have illustrated a probabilistic model that learns frames from text, combining document and syntactic contexts in a Dirichlet-multinomial latent variable model. Many further extensions are possible. First, document context could be enriched with various metadata—a document's context in time, space, and author

[4] http://brenocon.com/muc4_proc/ and http://github.com/brendano/muc4_proc

[5] https://docs.google.com/document/pub?id=1erEEsWI9V0SapEecbn1AMy69Fy6TgSIdYVTsKRaF8vM

FN frame	Dice	In both	Only in FN	Only in model
Change position on a scale	0.320	plummet, skyrocket, tumble, dwindle, double, rise, triple, fall *[B=12]*	decline, rocket, mushroom, advance, drop, reach *[M=16]* *[X=0]*	shoot, represent, plunge, return, appear, show *[O=16]* be, exceed, hover, sink, stabilize, surge *[N=19]*
Statement	0.265	assert, suggest, explain, add, note, say, caution, report *[B=9]*	comment, attest, relate, address, insist, allege *[M=29]* aver, pout, conjecture, avow, gloat *[X=5]*	respond, decline, appear, describe, testify, indicate *[O=10]* be, deny, refuse, continue, emphasize, refer *[N=6]*
Text creation	0.227	write, type, pen, compose, draft *[B=5]*	utter, say, chronicle, author *[M=4]* jot *[X=1]*	translate, prepare, get, read, study, contribute *[O=14]* promote, edit, censor, deliver, submit, research *[N=15]*
Request	0.218	urge, beg, summon, implore, ask, tell *[B=6]*	command, demand, request, order, plead *[M=5]* entreat, beseech *[X=2]*	give, visit, telephone, notify, thank, lead *[O=22]* warn, convince, invite, quote, defy, cross-examine *[N=14]*
Killing	0.206	starve, murder, slaughter, drown, kill, lynch, assassinate *[B=7]*	slay, liquidate, massacre, smother, butcher, dispatch *[M=8]* crucify, asphyxiate, annihilate, garrotte, behead, decapitate *[X=6]*	blow, shoot, hit, torture, injure, intimidate *[O=18]* harm, cheat, guard, strangle, kidnap, dismember *[N=22]*
Evidence	0.200	reveal, show, contradict, support, prove, confirm, indicate, demonstrate *[B=8]*	attest, verify, testify, evidence, corroborate, disprove *[M=9]* evince *[X=1]*	emerge, conclude, relate, describe, discover, examine *[O=25]* point, focus, imply, exist, result, determine *[N=29]*
Compliance	0.185	violate, flout, break, observe, follow, obey *[B=6]*	conform, breach, comply, adhere *[M=4]* contravene *[X=1]*	use, set, contradict, ease, evade, soften *[O=14]* loosen, adopt, rewrite, strengthen, revamp, administer *[N=34]*
Getting	0.182	win, acquire, obtain, gain *[B=4]*	get *[M=1]* *[X=0]*	secure, await, terminate, want, demand, seek *[O=6]* owe, dole, trade, need, withhold, guarantee *[N=29]*
Experiencer obj	0.176	satisfy, shock, offend, infuriate, puzzle, reassure, scare, enrage *[B=17]*	unsettle, distress, rattle, frighten, confuse, sting *[M=48]* mortify, displease, exhilarate, disconcert, astound, hearten *[X=48]*	despise, love, divide, back, bring, want *[O=24]* force, transfix, owe, haunt, involve, persuade *[N=39]*
Hindering	0.170	hinder, obstruct, impede, hamper *[B=4]*	inhibit, interfere *[M=2]* trammel, constrain, encumber *[X=3]*	head, thwart, lead, avoid, pass, harass *[O=16]* insure, overstep, harm, suspend, monitor, intercept *[N=18]*
Awareness	0.167	conceive, suspect, know *[B=3]*	comprehend, understand, presume, imagine, believe, think *[M=6]* reckon *[X=1]*	notice, calm, stereotype, rattle, recognize, like *[O=14]* dig, remember, figure, spell, prefer, suppose *[N=9]*
Telling	0.163	inform, tell, notify, assure *[B=4]*	confide, advise *[M=2]* apprise *[X=1]*	give, question, telephone, thank, lead, visit *[O=24]* warn, convince, invite, remind, quote, defy *[N=14]*
Activity start	0.162	start, begin, enter *[B=3]*	initiate, launch, commence, swing *[M=4]* *[X=0]*	quit, cut, run, attend, skip, fix *[O=11]* finish, complete, reform, disrupt, rock, offer *[N=16]*
Path shape	0.160	dip, drop, reach, edge *[B=4]*	emerge, swerve, angle, veer, crisscross, snake *[M=17]* traverse, ascend, undulate, slant, zigzag, ford *[X=7]*	jump, accord, move, begin, stay, explode *[O=10]* slow, figure, rebound, tend, constitute, range *[N=8]*
Cause harm	0.157	hit, torture, injure, bludgeon, hurt, stab, strike, batter *[B=8]*	cane, pummel, bruise, clout, hammer, whip *[M=36]* electrocute, spear, pelt, cudgel, squash, horsewhip *[X=11]*	blow, starve, murder, intimidate, impress, ambush *[O=17]* harm, cheat, guard, strangle, kidnap, dismember *[N=22]*

Table 4: FN verbset single best matches to Model 2 on CRIMENYT, $F = C = 400$, having a best-Dice score at least 0.15. We break down the set comparison as per Figure 5, showing up to several words words from each subset. The full set size is given in square brackets. B = in both wordsets. M = "missed": not in this model wordset. X = not in model's vocabulary. O = in other frames. N = "new": not in our FN verb extraction, i.e. appears fewer than 5 times in the FrameNet annotated data.

FN frame	BrownPTB (100)	BrownPTB (400)	WSJPTB (100)	WSJPTB (400)	CrimeNYT (100)	CrimeNYT (400)	UnifNYT (400)
Change pos. on a scale	10	10	**57**	**49**	18	**32**	**37**
Statement	11		**31**	**26**	**34**	**26**	**28**
Cause chg. of pos. on scl.	14	13	18	**26**		11	12
Body movement	18	15	10		12	14	**21**
Awareness		**27**		11		16	15
Motion directional			**23**	18		14	13
Appearance	14	17	10			14	14
Becoming	15	12		10		13	16
Evidence				12		20	19
Arriving		13		11		12	**21**
Causation		14		**22**		12	10
Using	18	20		16			
Path shape			15	16		16	12
Getting				13		18	15
Cause harm	14				11	15	16
Self motion	10				13	12	15
Motion			13	16			14
Request						**21**	18
Change posture		12	10				12
Becoming aware	10	11		12		11	12
Coming to be	18			12		12	
Cause change		10		14		11	15
Manipulation		10			13	12	13
Ingest substance	11	12					11
Compliance						18	17
Coming to believe				16		11	10
Perception experience	13			12		11	
Departing	14					11	
Removing						13	12
Contingency		16					15
Cotheme				16		10	13
Bringing				12		13	10
Categorization		14		11			
Topic				10		14	15
Placing					10	12	10
Communicate categ’n				13		12	10
Reveal secret	15					12	12
Activity start				11		16	
Text creation						**22**	
Expectation		11		10			
Experience bodily harm	11					10	
Cause motion	11						
Leadership		14		10			10
Experiencer obj					11	17	12
Cause expansion	13			15			
Posture	10					10	11
Perception active						10	10
Birth			11	10			10
Cooking creation				14		10	
Activity stop	16						11
Traversing							11
Giving						12	11
Ride vehicle							18
Building			13	12			
Contacting						14	11
Filling						10	
Killing						**20**	15
Telling						16	16
Cause impact							
Reasoning				12		11	

Table 5: For a FrameNet frame’s verbset, its single-best-match Dice scores (multiplied by 100) against several different models and datasets. Scores less than 0.1 are not shown; greater than 0.2 are bolded. 60 frames are shown, in decreasing order of average match score across datasets/models. All runs are Model 2, with $F = C =$ the number in parentheses.

attributes can easily be incorporated in the graphical models framework. Second, the restrictions to verb-subject-object syntactic constructions must be relaxed in order to capture the types of arguments seen in semantic role labeling and information extraction.

9 Appendix

9.1 Dice, F-measure, and set similarity

[6] Let A be the set of found items, and B the set of wanted items. $Prec = |AB|/|A|$, $Rec = |AB|/|B|$. Their harmonic mean, the $F1$-measure, is the same as the Dice coefficient:

$$\begin{aligned} F1(A,B) &= \frac{2}{1/P + 1/R} = \frac{2}{|A|/|AB| + |B|/|AB|} \\ Dice(A,B) &= \frac{2|AB|}{|A| + |B|} \\ &= \frac{2|AB|}{(|AB| + |A \setminus B|) + (|AB| + |B \setminus A|)} \\ &= \frac{|AB|}{|AB| + \frac{1}{2}|A \setminus B| + \frac{1}{2}|B \setminus A|} \end{aligned}$$

This illustrates Dice's close relationship to the Jaccard metric,

$$\begin{aligned} Jacc(A,B) &= \frac{|AB|}{|A \cup B|} \\ &= \frac{|AB|}{|AB| + |A \setminus B| + |B \setminus A|} \end{aligned}$$

And in fact $J = D/(2 - D)$ and $D = 2J/(1 + J)$ for any input, so they are monotonic in one another. The Tversky index (1977) generalizes them both,

$$Tversky(A,B;\alpha,\beta) = \frac{|AB|}{|AB| + \alpha|A \setminus B| + \beta|B \setminus A|}$$

where α and β control the magnitude of penalties of false positive versus false negative errors. All weighted F-measures correspond to when $\alpha + \beta = 1$.

9.2 Appendix: Dirichlet-multinomial conjugacy and the DM

Consider the two-stage model

$$\theta \sim Dir(\alpha), \;\; x \sim Multinom(\theta)$$

where α is a real-valued vector Dirichlet parameter, and x is a vector of outcome counts. Let $A = \sum_k \alpha_k$; this is the concentration parameter.

$p(x|\alpha) = \int p(x|\theta)p(\theta|\alpha)d\theta$ is the Dirichlet-multinomial, a.k.a. Multivariate Polya distribution or Dirichlet-compound multinomial. It is a distribution over count vectors, just like the multinomial, except it has the capacity to prefer different levels of sparseness vs. non-sparseness.

[6] A version of this appendix was published at http://brenocon.com/blog/2012/04/f-scores-dice-and-jaccard-set-similarity/.

First note the Dirichlet density

$$p(\theta|\alpha) = \frac{\Gamma A}{\prod \Gamma \alpha_k} \prod \theta_k^{\alpha_k - 1} \quad = \quad \frac{1}{B(\alpha)} \prod \theta_k^{\alpha_k - 1} \tag{1}$$

where $B(\alpha)$ is the multivariate Beta function

$$B(\alpha) = \int \prod \theta_k^{\alpha_k - 1} d\theta = \frac{\prod \Gamma \alpha_k}{\Gamma A}$$

Now derive the DM PMF:

$$p(x|\alpha) = \int p(x|\theta)\ p(\theta|\alpha)\ d\theta \tag{2}$$

$$= \int Multinom(x;\theta)\ Dir(\theta;\alpha)\ d\theta \tag{3}$$

$$= \int \left(\frac{N!}{\prod x_k!} \prod \theta_k^{x_k} \right) \left(\frac{1}{B(\alpha)} \prod \theta_k^{\alpha_k - 1} \right) d\theta \tag{4}$$

Because of conjugacy (i.e., the densities play nicely under multiplication), we can combine them into the integral of a new unnormalized Dirichlet, and rewrite into closed form.

$$p(x|\alpha) = \frac{N!}{\prod x_k!} \frac{1}{B(\alpha)} \int \prod \theta_k^{x_k + \alpha_k - 1} d\theta \tag{5}$$

$$= \frac{N!}{\prod x_k!} \frac{B(\alpha + x)}{B(\alpha)} \tag{6}$$

$$DM(x;\alpha) \equiv \underbrace{\frac{N!}{\prod x_k!}}_{\text{n. seq}} \underbrace{\frac{\Gamma(A)}{\Gamma(A+N)} \prod \frac{\Gamma(\alpha_k + x_k)}{\Gamma(\alpha_k)}}_{\text{prob of a seq having counts } \vec{x}} \tag{7}$$

where $A = \sum \alpha_k$ and $N = \sum x_k$. To calculate this, one would rewrite the "number of sequences term" using $N! = \Gamma(N+1)$ and $x_k! = \Gamma(x_k + 1)$ then use the log-gamma function for everything. To calculate the log-probability of only a single sequence, omit the initial term $N!/\prod x_k!$; we call this a "single path DM" or "DM1":

$$DM1(x;\alpha) = \frac{\Gamma(A)}{\Gamma(A+N)} \prod \frac{\Gamma(\alpha_k + x_k)}{\Gamma(\alpha_k)} \tag{8}$$

Note the DM gives, for small α priors, a bowed-out preference to count vectors that lie on the extremes of the N-simplex—just like the Dirichlet with small α prefers bowed-out points on the 1-simplex. This can be seen as a preference for "bursty" behavior. A multinomial cannot do this: you only can specify the mean, then the variance is fixed. In the DM, you control both mean and variance ($\approx$ inverse concentration), allowing burstiness a.k.a. over/under-dispersion. See Figure 6 for an illustration.

For a single DM draw where $N = 1$, instead of representing the draw as a sparse count vector x, we can represent it instead as an integer ID, $z \in \{1..K\}$. In the PMF, all the combinatorics drop away, leaving:

$$DM(z;\alpha) = \frac{\alpha_z}{A}$$

which is simply the Dirichlet mean parameter. When there's only one draw there's no such thing as burstiness or not.

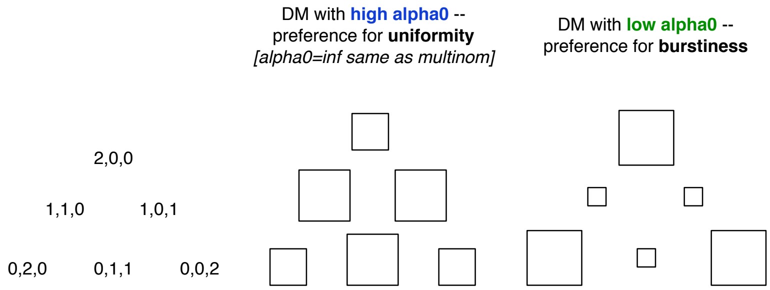


Figure 6: For two throws of a three-sided die, the six possible outcomes are laid out as shown on **left** (reminiscent of a simplex). **Center:** The PMF of a two-draw multinomial, mean parameter $p = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. Area of box corresponds to probability of the outcome. **Right:** The PMF of a Dirichlet-multinomial, with the same mean, but a low concentration A. No multinomial can represent this distributiom.

It is also easy to derive the posterior predictive distribution for a Dirichlet-Multinomial hierarchical model. We want to know the probability of the next draw z given the previous draws z_{-i}, using our entire posterior beliefs about θ. Represent z_{-i} as count vector $\vec{n} = (n_1..n_K)$, i.e. $n_k = \sum_{j \neq i} 1\{z_j = k\}$:

$$p(z|z_{-i}, \vec{\alpha}) = \int p(z|\theta)\ p(\theta|z_{-i}, \alpha)\ d\theta \tag{9}$$

$$= \int Multinom(z;\theta)\ Dir(\theta; \vec{\alpha} + \vec{n})\ d\theta \tag{10}$$

The second step used Dirichlet-multinomial conjugacy. Now this is just the 1-draw DM (i.e. the mean of the conjugately-updated Dirichlet),

$$p(z|z_{-i}, \vec{\alpha}) = DM(z; \vec{\alpha} + \vec{n}) \tag{11}$$

$$= \frac{\alpha_z + n_z}{A + N} \tag{12}$$

9.3 DM PMF in LDA hyperparameter sampling

Going through the full DM is not necessary to derive the collapsed Gibbs sampling equations, but it *is* necessary for hyperparameter sampling, which requires evaluating the likelihood of the entire dataset under different hyperparamters α. LDA under collapsing can be viewed as a series of DM draws:

- For each d, sample vector $z_{\{i:\ d_i = d\}} \sim DMPath(\alpha)$
- For each k, sample vector $w_{\{i:\ z_i = k\}} \sim DMPath(\beta)$

where "DMPath" indicates choosing one random sequence having the counts of one DM draw; its PMF is the $DM1$ function of its count vector. (This could be computed by proceeding through a Polya urn process a.k.a. (finite) Chinese restaurant process.) Therefore, for their Gibbs update steps, the hyperparameter likelihoods are:

$$p(z \mid \alpha) = \prod_d DMPath(z_{\{i:\ d_i = d\}}; \alpha) \tag{13}$$

$$p(w \mid z, \beta) = \prod_k DMPath(w_{\{i:\ z_i = k\}}; \beta) \tag{14}$$

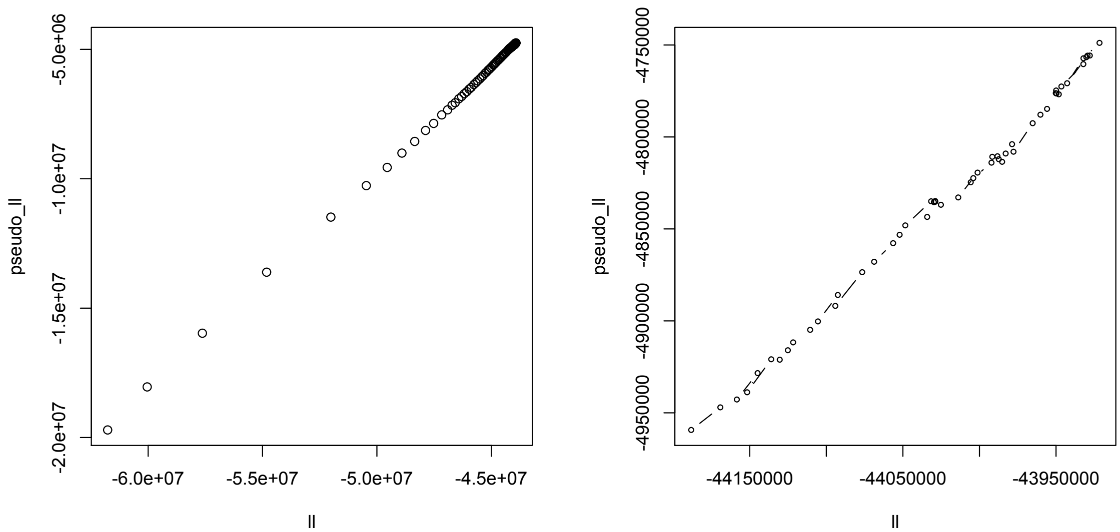


Figure 7: Correlation of running pseudolikelihood (evaluated during Gibbs sampling) to actual likelihood (evaluated exactly via the DM PMF (section 9.2)), for one MCMC run (small dataset, Model 1). Left: shown for all iterations where likelihood was evaluated. Right: shown for iterations 500 and later, with lines drawn between successive iterations.

For the other models, analogous formulations are available as well. We were initially tempted to try to compute the likehoods with per-token local conditionals similar to what is used for the z Gibbs updates,

$$\prod_i p(w_i|z_i, \beta)\, p(z_i|z_{-i}; \alpha)$$

which is easy to compute, but unfortunately wrong: it is actually a pseudolikelihood approximation to the likelihood (Besag 1975). Since it is possible to compute the actual likelihood closed-form log-gammas, we do so.

However, it does turn out the running-sum pseudolikelihood is a good approximation, as shown in Figure 7, for correlating to likelihood across MCMC samples, thus could be used to aid MCMC convergence diagnosis in situations where full likelihood evaluation is computationally expensive or difficult to implement.

References

David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:9931022, 2003.

Jordan Boyd-Graber and David M. Blei. Syntactic topic models. In *Neural Information Processing Systems*, 2008.

Nathanael. Chambers and D. 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 ACL*, 2011a.

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

J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. M Blei. Reading tea leaves: How humans interpret topic models. In *Proceedings of NIPS*, 2009.

S.R. Cook, A. Gelman, and D.B. Rubin. Validation of software for bayesian models using posterior quantiles. *Journal of Computational and Graphical Statistics*, 15(3):675–692, 2006.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A. Smith. Probabilistic frame-semantic parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 948–956. Association for Computational Linguistics, 2010.

M. C de Marneffe and C. D Manning. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, page 18, 2008.

M. C De Marneffe, B. MacCartney, and C. D Manning. Generating typed dependency parses from phrase structure parses. In *LREC 2006*, 2006.

Jacob Eisenstein, Brendan O'Connor, Noah A. Smith, and Eric P. Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1277—1287, 2010.

Charles Fillmore. Frame semantics. In *Linguistics in the Morning Calm*. Hanshin Publishing Company, Seoul, South Korea, 1982.

Charles J. Fillmore and Collin F. Baker. Frame semantics for text understanding. In *Proceedings of WordNet and Other Lexical Resources Workshop*, pages 3–5. Pittsburgh, NAACL, 2001.

Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28 (3):245–288, 2002.

T. Grenager and C. D Manning. Unsupervised discovery of a statistical verb lexicon. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, page 18, 2006. ISBN 1932432736.

T.L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228, 2004.

T.L. Griffiths, M. Steyvers, D.M. Blei, and J.B. Tenenbaum. Integrating topics and syntax. *Advances in neural information processing systems*, 17:537–544, 2005.

Joel Lang and Mirella Lapata. Unsupervised induction of semantic roles. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 939–947. Association for Computational Linguistics, 2010.

W. G. Lehnert and M. H. Ringle, editors. *Strategies for Natural Language Processing*. Psychology Press, August 1982. ISBN 0898592666.

Beth Levin. *English verb classes and alternations: A preliminary investigation*, volume 348. University of Chicago press Chicago, IL, 1993.

M. P Marcus, B. Santorini, and M. A Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313330, 1994.

David. Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. In *Uncertainty in Artificial Intelligence*, pages 411–418, 2008.

Radford M. Neal. Slice sampling. *Annals of Statistics*, pages 705–741, 2003.

R.M. Neal. Bayesian mixture modeling. In *Maximum Entropy and Bayesian Methods: Proceedings of the 11th International Workshop on Maximum Entropy and Bayesian Methods of Statistical Analysis, Seattle*, volume 1991, page 197211, 1992.

David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed algorithms for topic models. *The Journal of Machine Learning Research*, 10:1801–1828, 2009.

S. Patwardhan and E. Riloff. Effective information extraction with semantic affinity patterns and relevant regions. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 717–727, 2007.

F. Pereira, N. Tishby, and L. Lee. Distributional clustering of english words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, page 183190, 1993.

Alan Ritter and Oren Etzioni. A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 424–434. Association for Computational Linguistics, 2010.

M. Rooth, S. Riezler, D. Prescher, G. Carroll, and F. Beil. Inducing a semantically annotated lexicon via EM-based clustering. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, page 104111, 1999.

J. Ruppenhofer, M. Ellsworth, M.R.L. Petruck, C.R. Johnson, and J. Scheffczyk. Framenet ii: Extended theory and practice. *International Computer Science Institute*, 2006.

Evan Sandhaus. The New York Times Annotated Corpus, 2008.

Roger C. Schank and Robert P. Abelson. *Scripts, Plans, Goals, Understanding: An Inquiry Into Human Knowledge Structures*. Psychology Press, 1977.

Diarmuid O. Séaghdha. Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 435–444. Association for Computational Linguistics, 2010.

Lin Sun and Anna Korhonen. Improving verb clustering with automatically acquired selectional preferences. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 638–647. Association for Computational Linguistics, 2009.

Lin Sun, Anna Korhonen, and Yuval Krymolowski. Verb class discovery from rich syntactic data. *Computational Linguistics and Intelligent Text Processing*, pages 16–27, 2008.

Ivan Titov and Alexandre Klementiev. A bayesian model for unsupervised semantic parsing. In *Proceedings of ACL*, 2011.

H.M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984. ACM, 2006.