# SPRITE: Generalizing Topic Models with Structured Priors

**Michael J. Paul** and **Mark Dredze**
Department of Computer Science
Human Language Technology Center of Excellence
Johns Hopkins University, Baltimore, MD 21218
mpaul@cs.jhu.edu, mdredze@cs.jhu.edu

## Abstract

We introduce SPRITE, a family of topic models that incorporates structure into model priors as a function of underlying components. The structured priors can be constrained to model topic hierarchies, factorizations, correlations, and supervision, allowing SPRITE to be tailored to particular settings. We demonstrate this flexibility by constructing a SPRITE-based model to jointly infer topic hierarchies and author perspective, which we apply to corpora of political debates and online reviews. We show that the model learns intuitive topics, outperforming several other topic models at predictive tasks.

## 1 Introduction

Topic models can be a powerful aid for analyzing large collections of text by uncovering latent interpretable structures without manual supervision. Yet people often have expectations about topics in a given corpus and how they should be structured for a particular task. It is crucial for the user experience that topics meet these expectations (Mimno et al., 2011; Talley et al., 2011) yet black box topic models provide no control over the desired output.

This paper presents SPRITE, a family of topic models that provide a flexible framework for encoding preferences as priors for how topics should be structured. SPRITE can incorporate many types of structure that have been considered in prior work, including hierarchies (Blei et al., 2003a; Mimno et al., 2007), factorizations (Paul and Dredze, 2012; Eisenstein et al., 2011), sparsity (Wang and Blei, 2009; Balasubramanyan and Cohen, 2013), correlations between topics (Blei and Lafferty, 2007; Li and McCallum, 2006), preferences over word choices (Andrzejewski et al., 2009; Paul and Dredze, 2013), and associations between topics and document attributes (Ramage et al., 2009; Mimno and McCallum, 2008).

SPRITE builds on a standard topic model, adding structure to the *priors* over the model parameters. The priors are given by log-linear functions of underlying *components* (§2), which provide additional latent structure that we will show can enrich the model in many ways. By applying particular constraints and priors to the component hyperparameters, a variety of structures can be induced such as hierarchies and factorizations (§3), and we will show that this framework captures many existing topic models (§4).

After describing the general form of the model, we show how SPRITE can be tailored to particular settings by describing a specific model for the applied task of jointly inferring topic hierarchies and perspective (§6). We experiment with this topic+perspective model on sets of political debates and online reviews (§7), and demonstrate that SPRITE learns desired structures while outperforming many baselines at predictive tasks.

## 2 Topic Modeling with Structured Priors

Our model family generalizes latent Dirichlet allocation (LDA) (Blei et al., 2003b). Under LDA, there are $K$ topics, where a topic is a categorical distribution over $V$ words parameterized by $\phi_k$. Each document has a categorical distribution over topics, parameterized by $\theta_m$ for the $m$th document. Each observed word in a document is generated by drawing a topic $z$ from $\theta_m$, then drawing the word from $\phi_z$. $\theta$ and $\phi$ have priors given by Dirichlet distributions.

Our generalization adds structure to the generation of the Dirichlet parameters. The priors for these parameters are modeled as log-linear combinations of underlying *components*. Components are real-valued vectors of length equal to the vocabulary size $V$ (for priors over word distributions) or length equal to the number of topics $K$



Transactions of the Association for Computational Linguistics, vol. 3, pp. 43–57, 2015. Action Editor: Janyce Wiebe.
Submission batch: 7/2014; Revision batch 12/2014; Published 1/2015. 

(for priors over topic distributions).

For example, we might assume that topics about sports like baseball and football share a common prior – given by a component – with general words about sports. A fine-grained topic about steroid use in sports might be created by combining components about broader topics like sports, medicine, and crime. By modeling the priors as combinations of components that are shared across all topics, we can learn interesting connections between topics, where components provide an additional latent layer for corpus understanding.

As we'll show in the next section, by imposing certain requirements on which components feed into which topics (or documents), we can induce a variety of model structures. For example, if we want to model a topic hierarchy, we require that each topic depend on exactly one parent component. If we want to jointly model topic and ideology in a corpus of political documents (§6), we make topic priors a combination of one component from each of two groups: a topical component and an ideological component, resulting in ideology-specific topics like "conservative economics".

Components construct priors as follows. For the topic-specific word distributions $\phi$, there are $C^{(\phi)}$ *topic components*. The $k$th topic's prior over $\phi_k$ is a weighted combination (with coefficient vector $\beta_k$) of the $C^{(\phi)}$ components (where component $c$ is denoted $\omega_c$). For the document-specific topic distributions $\theta$, there are $C^{(\theta)}$ *document components*. The $m$th document's prior over $\theta_m$ is a weighted combination (coefficients $\alpha_m$) of the $C^{(\theta)}$ components (where component $c$ is denoted $\delta_c$).

Once conditioned on these priors, the model is identical to LDA. The generative story is described in Figure 1. We call this family of models **SPRITE**: **S**tructured **PRI**or **T**opic mod**E**ls.

To illustrate the role that components can play, consider an example in which we are modeling research topics in a corpus of NLP abstracts (as we do in §7.3). Consider three speech-related topics: signal processing, automatic speech recognition, and dialog systems. Conceptualized as a hierarchy, these topics might belong to a higher level category of spoken language processing. SPRITE allows the relationship between these three topics to be defined in two ways. One, we can model that these topics will all have words in common. This is handled by the topic components – these three topics could all draw from a common "spoken language" topic component, with high-weight words such as *speech* and *spoken*, which informs the prior of all three topics. Second, we can model that these topics are likely to occur together in documents. For example, articles about dialog systems are likely to discuss automatic speech recognition as a subroutine. This is handled by the document components – there could be a "spoken language" document component that gives high weight to all three topics, so that if a document draw its prior from this component, then it is more likely to give probability to these topics together.

- Generate hyperparameters: $\alpha$, $\beta$, $\delta$, $\omega$ (§3)
- For each document $m$, generate parameters:
  1. $\tilde{\theta}_{mk} = \exp(\sum_{c=1}^{C^{(\theta)}} \alpha_{mc}\, \delta_{ck})$, $1 \leq k \leq K$
  2. $\theta_m \sim \text{Dirichlet}(\tilde{\theta}_m)$
- For each topic $k$, generate parameters:
  1. $\tilde{\phi}_{kv} = \exp(\sum_{c=1}^{C^{(\phi)}} \beta_{kc}\, \omega_{cv})$, $1 \leq v \leq V$
  2. $\phi_k \sim \text{Dirichlet}(\tilde{\phi}_k)$
- For each token $(m, n)$, generate data:
  1. Topic (unobserved): $z_{m,n} \sim \theta_m$
  2. Word (observed): $w_{m,n} \sim \phi_{z_{m,n}}$

Figure 1: The generative story of SPRITE. The difference from latent Dirichlet allocation (Blei et al., 2003b) is the generation of the Dirichlet parameters.

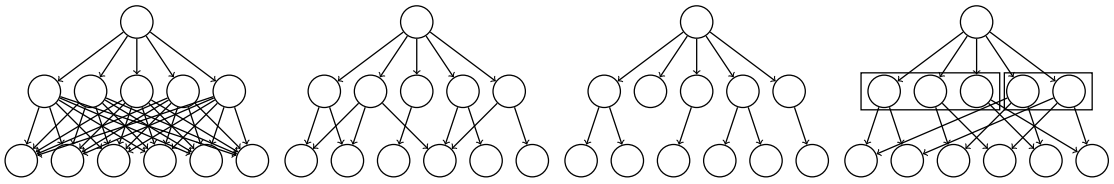
The next section will describe how particular priors over the coefficients can induce various structures such as hierarchies and factorizations, and components and coefficients can also be provided as input to incorporate supervision and prior knowledge. The general prior structure used in SPRITE can be used to represent a wide array of existing topic models, outlined in Section 4.

## 3 Topic Structures

By changing the particular configuration of the hyperparameters – the component coefficients ($\alpha$ and $\beta$) and the component weights ($\delta$ and $\omega$) – we obtain a diverse range of model structures and behaviors. We now describe possible structures and the corresponding priors.

### 3.1 Component Structures

This subsection discusses various graph structures that can describe the relation between topic components and topics, and between document components and documents, illustrated in Figure 2.



(a) Dense DAG (b) Sparse DAG (c) Tree (d) Factored Forest

Figure 2: Example graph structures describing possible relations between components (middle row) and topics or documents (bottom row). Edges correspond to non-zero values for $\alpha$ or $\beta$ (the component coefficients defining priors over the document and topic distributions). The root node is a shared prior over the component weights (with other possibilities discussed in §3.3).

#### 3.1.1 Directed Acyclic Graph

The general SPRITE model can be thought of as a dense directed acyclic graph (DAG), where every document or topic is connected to every component with some weight $\alpha$ or $\beta$. When many of the $\alpha$ or $\beta$ coefficients are zero, the DAG becomes sparse. A sparse DAG has an intuitive interpretation: each document or topic depends on some subset of components.

The default prior over coefficients that we use in this study is a 0-mean Gaussian distribution, which encourages the weights to be small. We note that to induce a sparse graph, one could use a 0-mean Laplace distribution as the prior over $\alpha$ and $\beta$, which prefers parameters such that some components are zero.

#### 3.1.2 Tree

When each document or topic has exactly one parent (one nonzero coefficient) we obtain a two-level tree structure. This structure naturally arises in topic hierarchies, for example, where fine-grained topics are children of coarse-grained topics.

To create an (unweighted) tree, we require $\alpha_{mc} \in \{0, 1\}$ and $\sum_c \alpha_{mc} = 1$ for each document $m$. Similarly, $\beta_{kc} \in \{0, 1\}$ and $\sum_c \beta_{kc} = 1$ for each topic $k$. In this setting, $\alpha_m$ and $\beta_k$ are indicator vectors which select a single component.

In this study, rather than strictly requiring $\alpha_m$ and $\beta_k$ to be binary-valued indicator vectors, we create a relaxation that allows for easier parameter estimation. We let $\alpha_m$ and $\beta_k$ to real-valued variables in a simplex, but place a prior over their values to encourage sparse values, favoring vectors with a single component near 1 and others near 0. This is achieved using a Dirichlet($\rho < 1$) distribution as the prior over $\alpha$ and $\beta$, which has higher density near the boundaries of the simplex.[1]

For a weighted tree, $\alpha$ and $\beta$ could be a product of two variables: an "integer-like" indicator vector with sparse Dirichlet prior as suggested above, combined with a real-valued weight (e.g., with a Gaussian prior). We take this approach in our model of topic and perspective (§6).

#### 3.1.3 Factored Forest

By using structured sparsity over the DAG, we can obtain a structure where components are grouped into $G$ factors, and each document or topic has one parent from each group. Figure 2(d) illustrates this: the left three components belong to one group, the right two belong to another, and each bottom node has exactly one parent from each. This is a DAG that we call a "factored forest" because the subgraphs associated with each group in isolation are trees. This structure arises in "multi-dimensional" models like SAGE (Eisenstein et al., 2011) and Factorial LDA (Paul and Dredze, 2012), which allow tokens to be associated with multiple variables (e.g. a topic along with a variable denoting positive or negative sentiment). This allows word distributions to depend on both factors.

The "exactly one parent" indicator constraint is the same as in the tree structure but enforces a tree only within each group. This can therefore be (softly) modeled using a sparse Dirichlet prior as described in the previous subsection. In this case, the subsets of components belonging to each factor have separate sparse Dirichlet priors. Using the example from Figure 2(d), the first three component indicators would come from one Dirichlet, while the latter two component indicators would come from a second.

### 3.2 Tying Topic and Document Components

A desirable property for many situations is for the topic and document components to correspond to

[1] This generalizes the technique used in Paul and Dredze (2012), who approximated binary variables with real-valued variables in $(0, 1)$, by using a "U-shaped" Beta($\rho < 1$) distribution as the prior to encourage sparsity. The Dirichlet distribution is the multivariate extension of the Beta distribution.

each other. For example, if we think of the components as coarse-grained topics in a hierarchy, then the coefficients $\beta$ enforce that topic word distributions share a prior defined by their parent $\omega$ component, while the coefficients $\alpha$ represent a document's proportions of coarse-grained topics, which effects the document's prior over child topics (through the $\delta$ vectors). Consider the example with spoken language topics in §2: these three topics (signal processing, speech recognition, and dialog systems) are *a priori* likely both to share the same words and to occur together in documents. By tying these together, we ensure that the patterns are consistent across the two types of components, and the patterns from both types can reinforce each other during inference.

In this case, the number of topic components is the same as the number of document components ($C^{(\phi)} = C^{(\theta)}$), and the coefficients ($\beta_{cz}$) of the topic components should correlate with the weights of the document components ($\delta_{zc}$). The approach we take (§6) is to define $\delta$ and $\beta$ as a product of two variables (suggested in §3.1.2): a binary mask variable (with sparse Dirichlet prior), which we let be identical for both $\delta$ and $\beta$, and a real-valued positive weight.

### 3.3 Deep Components

As for priors over the component weights $\delta$ and $\omega$, we assume they are generated from a 0-mean Gaussian. While not experimented with in this study, it is also possible to allow the components themselves to have rich priors which are functions of higher level components. For example, rather than assuming a mean of zero, the mean could be a weighted combination of higher level weight vectors. This approach was used by Paul and Dredze (2013) in Factorial LDA, in which each $\omega$ component had its own Gaussian prior provided as input to guide the parameters.

## 4 Special Cases and Extensions

We now describe several existing Dirichlet prior topic models and show how they are special cases of SPRITE. Table 1 summarizes these models and their relation to SPRITE. In almost every case, we also describe how the SPRITE representation of the model offers improvements over the original model or can lead to novel extensions.

| Model | Sec. | Document priors | Topic priors |
|---|---|---|---|
| LDA | 4.1 | Single component | Single component |
| SCTM | 4.2 | Single component | Sparse binary $\beta$ |
| SAGE | 4.3 | Single component | Sparse $\omega$ |
| FLDA | 4.3 | Binary $\delta$ is transpose of $\beta$ | Factored binary $\beta$ |
| PAM | 4.4 | $\alpha$ are supertopic weights | Single component |
| DMR | 4.5 | $\alpha$ are feature values | Single component |

Table 1: Topic models with Dirichlet priors that are generalized by SPRITE. The description of each model can be found in the noted section number. PAM is not equivalent, but captures very similar behavior. The described component formulations of SCTM and SAGE are equivalent, but these differ from SPRITE in that the components directly define the parameters, rather than priors over the parameters.

### 4.1 Latent Dirichlet Allocation

In LDA (Blei et al., 2003b), all $\theta$ vectors are drawn from the same prior, as are all $\phi$ vectors. This is a basic instance of our model with only one component at the topic and document levels, $C^{(\theta)} = C^{(\phi)} = 1$, with coefficients $\alpha = \beta = 1$.

### 4.2 Shared Components Topic Models

Shared components topic models (SCTM) (Gormley et al., 2010) define topics as products of "components", where components are word distributions. To use the notation of our paper, the $k$th topic's word distribution in SCTM is parameterized by $\phi_{kv} \propto \prod_c \omega_{cv}^{\beta_{kc}}$, where the $\omega$ vectors are word distributions (rather than vectors in $\mathbb{R}^V$), and the $\beta_{kc} \in \{0, 1\}$ variables are indicators denoting whether component $c$ is in topic $k$.

This is closely related to SPRITE, where topics also depend on products of underlying components. A major difference is that in SCTM, the topic-specific word distributions are exactly defined as a product of components, whereas in SPRITE, it is only the prior that is a product of components.[2] Another difference is that SCTM has an unweighted product of components ($\beta$ is binary), whereas SPRITE allows for weighted products. The log-linear parameterization leads to simpler optimization procedures than the product parameterization. Finally, the components in SCTM only apply to the word distributions, and not the topic distributions in documents.

### 4.3 Factored Topic Models

Factored topic models combine multiple aspects of the text to generate the document (instead of just topics). One such topic model is Factorial LDA (FLDA) (Paul and Dredze, 2012). In FLDA,

[2]The posterior becomes concentrated around the prior when the Dirichlet variance is low, in which case SPRITE behaves like SCTM. SPRITE is therefore more general.

"topics" are actually tuples of potentially multiple variables, such as aspect and sentiment in online reviews (Paul et al., 2013). Each document distribution $\theta_m$ is a distribution over pairs (or higher-dimensional tuples if there are more than two factors), and each pair $(j, k)$ has a word distribution $\phi_{(j,k)}$. FLDA uses a similar log-linear parameterization of the Dirichlet priors as SPRITE. Using our notation, the Dirichlet($\tilde{\phi}_{(j,k)}$) prior for $\phi_{(j,k)}$ is defined as $\tilde{\phi}_{(j,k),v}$=exp($\omega_{jv}$+$\omega_{kv}$), where $\omega_j$ is a weight vector over the vocabulary for the $j$th component of the first factor, and $\omega_k$ encodes the weights for the $k$th component of the second factor. (Some bias terms are omitted for simplicity.) The prior over $\theta_m$ has a similar form: $\tilde{\theta}_{m,(j,k)}$=exp($\alpha_{mj} + \alpha_{mk}$), where $\alpha_{mj}$ is document $m$'s preference for component $j$ of the first factor (and likewise for $k$ of the second).

This corresponds to an instantiation of SPRITE using an unweighted factored forest (§3.1.3), where $\beta_{zc} = \delta_{cz}$ (§3.2, recall that $\delta$ are document components while $\beta$ are the topic coefficients). Each subtopic $z$ (which is a pair of variables in the two-factor model) has one parent component from each factor, indicated by $\beta_z$ which is binary-valued. At the document level in the two-factor example, $\delta_j$ is an indicator vector with values of 1 for all pairs with $j$ as the first component, and thus the coefficient $\alpha_{mj}$ controls the prior for all such pairs of the form $(j, \cdot)$, and likewise $\delta_k$ indicates pairs with $k$ as the second component, controlling the prior over $(\cdot, k)$.

The SPRITE representation offers a benefit over the original FLDA model. FLDA assumes that the entire Cartesian product of the different factors is represented in the model (e.g. $\phi$ parameters for every possible tuple), which leads to issues with efficiency and overparameterization with higher numbers of factors. With SPRITE, we can simply fix the number of "topics" to a number smaller than the size of the Cartesian product, and the model will *learn* which subset of tuples are included, through the values of $\beta$ and $\delta$.

Finally, another existing model family that allows for topic factorization is the sparse additive generative model (SAGE) (Eisenstein et al., 2011). SAGE uses a log-linear parameterization to define word distributions. SAGE is a general family of models that need not be factored, but is presented as an efficient solution for including multiple factors, such as topic and geography or topic and author ideology. Like SCTM, $\phi$ is exactly defined as a product of $\omega$ weights, rather than our approach of using the product to define a prior over $\phi$.

### 4.4 Topic Hierarchies and Correlations

While the two previous subsections primarily focused on word distributions (with FLDA being an exception that focused on both), SPRITE's priors over topic distributions also have useful characteristics. The component-specific $\delta$ vectors can be interpreted as common topic distribution patterns, where each component is likely to give high weight to groups of topics that tend to occur together. Each document's $\alpha$ weights encode which of the topic groups are present in that document.

Similar properties are captured by the Pachinko allocation model (PAM) (Li and McCallum, 2006). Under PAM, each document has a distribution over *supertopics*. Each supertopic is associated with a Dirichlet prior over *subtopic* distributions, where subtopics are the low level topics that are associated with word parameters $\phi$. Documents also have supertopic-specific distributions over subtopics (drawn from each supertopic-specific Dirichlet prior). Each topic in a document is drawn by first drawing a supertopic from the document's distribution, then drawing a subtopic from that supertopic's document distribution.

While not equivalent, this is quite similar to SPRITE where document components correspond to supertopics. Each document's $\alpha$ weights can be interpreted to be similar to a distribution over supertopics, and each $\delta$ vector is that supertopic's contribution to the prior over subtopics. The prior over the document's topic distribution is thus affected by the document's supertopic weights $\alpha$.

The SPRITE formulation naturally allows for powerful extensions to PAM. One possibility is to include topic components for the word distributions, in addition to document components, and to tie together $\delta_{cz}$ and $\beta_{zc}$ (§3.2). This models the intuitive characteristic that subtopics belonging to similar supertopics (encoded by $\delta$) should come from similar priors over their word distributions (since they will have similar $\beta$ values). That is, children of a supertopic are topically related – they are likely to share words. This is a richer alternative to the hierarchical variant of PAM proposed by Mimno et al. (2007), which modeled separate word distributions for supertopics and subtopics, but the subtopics were not dependent on the super-

topic word distributions. Another extension is to form a strict tree structure, making each subtopic belong to exactly one supertopic: a true hierarchy.

### 4.5 Conditioning on Document Attributes

SPRITE also naturally provides the ability to condition document topic distributions on features of the document, such as a user rating in a review. To do this, let the number of document components be the number of features, and the value of $\alpha_{mc}$ is the $m$th document's value of the $c$th feature. The $\delta$ vectors then influence the document's topic prior based on the feature values. For example, increasing $\alpha_{mc}$ will increase the prior for topic $z$ if $\delta_{cz}$ is positive and decrease the prior if $\delta_{cz}$ is negative. This is similar to the structure used for PAM (§4.4), but here the $\alpha$ weights are fixed and provided as input, rather than learned and interpreted as supertopic weights. This is identical to the Dirichlet-multinomial regression (DMR) topic model (Mimno and McCallum, 2008). The DMR topic model define's each document's Dirichlet prior over topics as a log-linear function of the document's feature values and regression coefficients for each topic. The $c$th feature's regression coefficients correspond to the $\delta_c$ vector in SPRITE.

## 5 Inference and Parameter Estimation

We now discuss how to infer the posterior of the latent variables $\mathbf{z}$ and parameters $\theta$ and $\phi$, and find maximum *a posteriori* (MAP) estimates of the hyperparameters $\alpha$, $\beta$, $\delta$, and $\omega$, given their hyperpriors. We take a Monte Carlo EM approach, using a collapsed Gibbs sampler to sample from the posterior of the topic assignments $\mathbf{z}$ conditioned on the hyperparameters, then optimizing the hyperparameters using gradient-based optimization conditioned on the samples.

Given the hyperparameters, the sampling equations are identical to the standard LDA sampler (Griffiths and Steyvers, 2004). The partial derivative of the collapsed log likelihood $\mathcal{L}$ of the corpus with respect to each hyperparameter $\beta_{kc}$ is:

$$\frac{\partial \mathcal{L}}{\partial \beta_{kc}} = \frac{\partial P(\beta)}{\partial \beta_{kc}} + \sum_v \omega_{cv} \tilde{\phi}_{kv} \quad \times \quad \left(\Psi(n_v^k + \tilde{\phi}_{kv}) - \Psi(\tilde{\phi}_{kv}) + \Psi(\textstyle\sum_{k'} \tilde{\phi}_{k'v}) - \Psi(\sum_{k'} n_v^{k'} + \tilde{\phi}_{k'v})\right) \tag{1}$$

where $\tilde{\phi}_{kv}$=$\exp(\sum_{c'} \beta_{kc'}\omega_{c'v})$, $n_v^k$ is the number of times word $v$ is assigned to topic $k$ (in the samples from the E-step), and $\Psi$ is the digamma function, the derivative of the log of the gamma function. The digamma terms arise from the Dirichlet-multinomial distribution, when integrating out the parameters $\phi$. $P(\beta)$ is the hyperprior. For a 0-mean Gaussian hyperprior with variance $\sigma^2$, $\frac{\partial P(\beta)}{\partial \beta_{kc}} = -\frac{\beta_{kc}}{\sigma^2}$. Under a Dirchlet($\rho$) hyperprior, when we want $\beta$ to represent an indicator vector (§3.1.2), $\frac{\partial P(\beta)}{\partial \beta_{kc}} = \frac{\rho - 1}{\beta_{kc}}$.

The partial derivatives for the other hyperparameters are similar. Rather than involving a sum over the vocabulary, $\frac{\partial \mathcal{L}}{\partial \delta_{ck}}$ sums over documents, while $\frac{\partial \mathcal{L}}{\partial \omega_{cv}}$ and $\frac{\partial \mathcal{L}}{\partial \alpha_{mc}}$ sum over topics.

Our inference algorithm alternates between one Gibbs iteration and one iteration of gradient ascent, so that the parameters change gradually. For unconstrained parameters, we use the update rule: $\mathbf{x}^{t+1}$=$\mathbf{x}^t + \eta_t \nabla \mathcal{L}(\mathbf{x}^t)$, for some variable $\mathbf{x}$ and a step size $\eta_t$ at iteration $t$. For parameters constrained to the simplex (such as when $\beta$ is a soft indicator vector), we use exponentiated gradient ascent (Kivinen and Warmuth, 1997) with the update rule: $x_i^{t+1} \propto x_i^t \exp(\eta_t \nabla_i \mathcal{L}(x^t))$.

### 5.1 Tightening the Constraints

For variables that we prefer to be binary but have softened to continuous variables using sparse Beta or Dirichlet priors, we can straightforwardly strengthen the preference to be binary by modifying the objective function to favor the prior more heavily. Specifically, under a Dirichlet($\rho$<1) prior we will introduce a scaling parameter $\tau_t \geq 1$ to the prior log likelihood: $\tau_t \log P(\beta)$ with partial derivative $\tau_t \frac{\rho - 1}{\beta_{kc}}$, which adds extra weight to the sparse Dirichlet prior in the objective. The algorithm used in our experiments begins with $\tau_1 = 1$ and optionally increases $\tau$ over time. This is a deterministic annealing approach, where $\tau$ corresponds to an inverse temperature (Ueda and Nakano, 1998; Smith and Eisner, 2006).

As $\tau$ approaches infinity, the prior-annealed MAP objective $\max_\beta P(\phi|\beta)P(\beta)^\tau$ approaches $\max_\beta P(\phi|\beta) \max_\beta P(\beta)$. Annealing only the prior $P(\beta)$ results in maximization of this term only, while the outer $\max$ chooses a good $\beta$ under $P(\phi|\beta)$ as a tie-breaker among all $\beta$ values that maximize the inner $\max$ (binary-valued $\beta$).[3]

We show experimentally (§7.2.2) that annealing the prior yields values that satisfy the constraints.

[3] Other modifications could be made to the objective function to induce sparsity, such as entropy regularization (Balasubramanyan and Cohen, 2013).

## 6 A Factored Hierarchical Model of Topic and Perspective

We will now describe a SPRITE model that encompasses nearly all of the structures and extensions described in §3–4, followed by experimental results using this model to jointly capture topic and "perspective" in a corpus of political debates (where perspective corresponds to ideology) and a corpus of online doctor reviews (where perspective corresponds to the review sentiment).

First, we will create a topic **hierarchy** (§4.4). The hierarchy will model both topics and documents, where $\alpha_m$ is document $m$'s supertopic proportions, $\delta_c$ is the $c$th supertopic's subtopic prior, $\omega_c$ is the $c$th supertopic's word prior, and $\beta_k$ is the weight vector that selects the $k$th topic's parent supertopic, which incorporates (soft) indicator vectors to encode a tree structure (§3.1.2).

We want a weighted tree; while each $\beta_k$ has only one nonzero element, the nonzero element can be a value other than 1. We do this by replacing the single coefficient $\beta_{kc}$ with a product of two variables: $b_{kc}\hat{\beta}_{kc}$. Here, $\hat{\beta}_k$ is a real-valued weight vector, while $b_{kc}$ is a binary indicator vector which zeroes out all but one element of $\beta_k$. We do the same with the $\delta$ vectors, replacing $\delta_{ck}$ with $b_{kc}\hat{\delta}_{ck}$. The $\mathbf{b}$ variables are shared across both topic and document components, which is how we tie these together (§3.2). We relax the binary requirement and instead allow a positive real-valued vector whose elements sum to 1, with a Dirichlet($\rho$<1) prior to encourage sparsity (§3.1.2).

To be properly interpreted as a hierarchy, we constrain the coefficients $\alpha$ and $\beta$ (and by extension, $\delta$) to be positive. To optimize these parameters in a mathematically convenient way, we write $\beta_{kc}$ as $\exp(\log \beta_{kc})$, and instead optimize $\log \beta_{kc} \in \mathbb{R}$ rather than $\beta_{kc} \in \mathbb{R}_+$.

Second, we **factorize** (§4.3) our hierarchy such that each topic depends not only on its supertopic, but also on a value indicating perspective. For example, a conservative topic about energy will appear differently from a liberal topic about energy. The prior for a topic will be a log-linear combination of both a supertopic (e.g. energy) and a perspective (e.g. liberal) weight vector. The variables associated with the perspective component are denoted with superscript $(P)$ rather than subscript $c$.

To learn meaningful perspective parameters, we include supervision in the form of document **attributes** (§4.5). Each document includes a positive or negative score denoting the perspective, which is the variable $\alpha_m^{(P)}$ for document $m$. Since $\alpha^{(P)}$ are the coefficients for $\delta^{(P)}$, positive values of $\delta_k^{(P)}$ indicate that topic $k$ is more likely if the author is conservative (which has a positive $\alpha$ score in our data), and less likely if the author is liberal (which has a negative score). There is only a single perspective component, but it represents two ends of a spectrum with positive and negative weights; $\beta^{(P)}$ and $\delta^{(P)}$ are not constrained to be positive, unlike the supertopics. We also set $\beta_k^{(P)} = \delta_k^{(P)}$. This means that topics with positive $\delta_k^{(P)}$ will also have a positive $\beta$ coefficient that is multiplied with the perspective word vector $\omega^{(P)}$.

- $\mathbf{b}_k \sim \text{Dirichlet}(\rho < 1)$ (soft indicator)
- $\alpha^{(P)}$ is given as input (perspective value)
- $\delta_k^{(P)} = \beta_k^{(P)}$
- $\tilde{\phi}_{kv} = \exp(\omega_v^{(B)} + \beta_k^{(P)}\omega_v^{(P)} + \sum_c b_{kc}\hat{\beta}_{kc}\omega_{cv})$
- $\tilde{\theta}_{mk} = \exp(\delta_k^{(B)} + \alpha_m^{(P)}\delta_k^{(P)} + \sum_c b_{kc}\alpha_{mc}\hat{\delta}_{ck})$

Figure 3: Summary of the hyperparameters in our SPRITE-based topic and perspective model (§6).

Finally, we include "bias" component vectors denoted $\omega^{(B)}$ and $\delta^{(B)}$, which act as overall weights over the vocabulary and topics, so that the component-specific $\omega$ and $\delta$ weights can be interpreted as deviations from the global bias weights.

Figure 3 summarizes the model. This includes most of the features described above (trees, factored structures, tying topic and document components, and document attributes), so we can ablate model features to measure their effect.

## 7 Experiments

### 7.1 Datasets and Experimental Setup

We applied our models to two corpora:

- *Debates:* A set of floor debates from the 109th–112th U.S. Congress, collected by Nguyen et al. (2013), who also applied a hierarchical topic model to this data. Each document is a transcript of one speaker's turn in a debate, and each document includes the first dimension of the DW-NOMINATE score (Lewis and Poole, 2004), a real-valued score indicating how conservative (positive) or liberal (negative) the speaker is. This value is $\alpha^{(P)}$. We took a sample of 5,000 documents from the House debates (850,374 tokens; 7,426 types), balanced across party affilia-

tion. We sampled from the most partisan speakers, removing scores below the median value.

- *Reviews:* Doctor reviews from RateMDs.com, previously analyzed using FLDA (Paul et al., 2013; Wallace et al., 2014). The reviews contain ratings on a 1–5 scale for multiple aspects. We centered the ratings around the middle value 3, then took reviews that had the same sign for all aspects, and averaged the scores to produce a value for $\alpha^{(P)}$. Our corpus contains 20,000 documents (476,991 tokens; 10,158 types), balanced across positive/negative scores.

Unless otherwise specified, $K$=50 topics and $C$=10 components (excluding the perspective component) for *Debates*, and $K$=20 and $C$=5 for *Reviews*. These values were chosen as a qualitative preference, not optimized for predictive performance, but we experiment with different values in §7.2.2. We set the step size $\eta_t$ according to AdaGrad (Duchi et al., 2011), where the step size is the inverse of the sum of squared historical gradients.[4] We place a sparse Dirichlet($\rho$=0.01) prior on the $\mathbf{b}$ variables, and apply weak regularization to all other hyperparameters via a $\mathcal{N}(0, 10^2)$ prior. These hyperparameters were chosen after only minimal tuning, and were selected because they showed stable and reasonable output qualitatively during preliminary development.

We ran our inference algorithm for 5000 iterations, estimating the parameters $\theta$ and $\phi$ by averaging the final 100 iterations. Our results are averaged across 10 randomly initialized samplers.[5]

## 7.2 Evaluating the Topic Perspective Model

### 7.2.1 Analysis of Output

Figure 4 shows examples of topics learned from the *Reviews* corpus. The figure includes the highest probability words in various topics as well as the highest weight words in the supertopic components and perspective component, which feed into the priors over the topic parameters. We see that one supertopic includes many words related to surgery, such as *procedure* and *performed*, and has multiple children, including a topic about dental work. Another supertopic includes words describing family members such as *kids* and *husband*. One topic has both supertopics as parents, which appears to describe surgeries that saved a family member's life, with top words including {*saved*, *life*, *husband*, *cancer*}. The figure also illustrates which topics are associated more with positive or negative reviews, as indicated by the value of $\delta^{(P)}$.

Interpretable parameters were also learned from the *Debates* corpus. Consider two topics about energy that have polar values of $\delta^{(P)}$. The conservative-leaning topic is about oil and gas, with top words including {*oil*, *gas*, *companies*, *prices*, *drilling*}. The liberal-leaning topic is about renewable energy, with top words including {*energy*, *new*, *technology*, *future*, *renewable*}. Both of these topics share a common parent of an industry-related supertopic whose top words are {*industry*, *companies*, *market*, *price*}. A nonpartisan topic under this same supertopic has top words {*credit*, *financial*, *loan*, *mortgage*, *loans*}.

### 7.2.2 Quantitative Evaluation

We evaluated the model on two predictive tasks as well as topic quality. The first metric is perplexity of held-out text. The held-out set is based on tokens rather than documents: we trained on even numbered tokens and tested on odd tokens. This is a type of "document completion" evaluation (Wallach et al., 2009b) which measures how well the model can predict held-out tokens of a document after observing only some.

We also evaluated how well the model can predict the attribute value (DW-NOMINATE score or user rating) of the document. We trained a linear regression model using the document topic distributions $\theta$ as features. We held out half of the documents for testing and measured the mean absolute error. When estimating document-specific SPRITE parameters for held-out documents, we fix the feature value $\alpha_m^{(P)} = 0$ for that document.

These predictive experiments do not directly measure performance at many of the particular tasks that topic models are well suited for, like data exploration, summarization, and visualization. We therefore also include a metric that more directly measures the quality and interpretability of topics. We use the topic *coherence* metric introduced by Mimno et al. (2011), which is based on co-occurrence statistics among each topic's most probable words and has been shown to correlate with human judgments of topic quality. This metric measures the quality of each topic, and we

[4] AdaGrad decayed too quickly for the $\mathbf{b}$ variables. For these, we used a variant suggested by Zeiler (2012) which uses an average of historical gradients rather than a sum.

[5] Our code and the data will be available at: `http://cs.jhu.edu/~mpaul`.

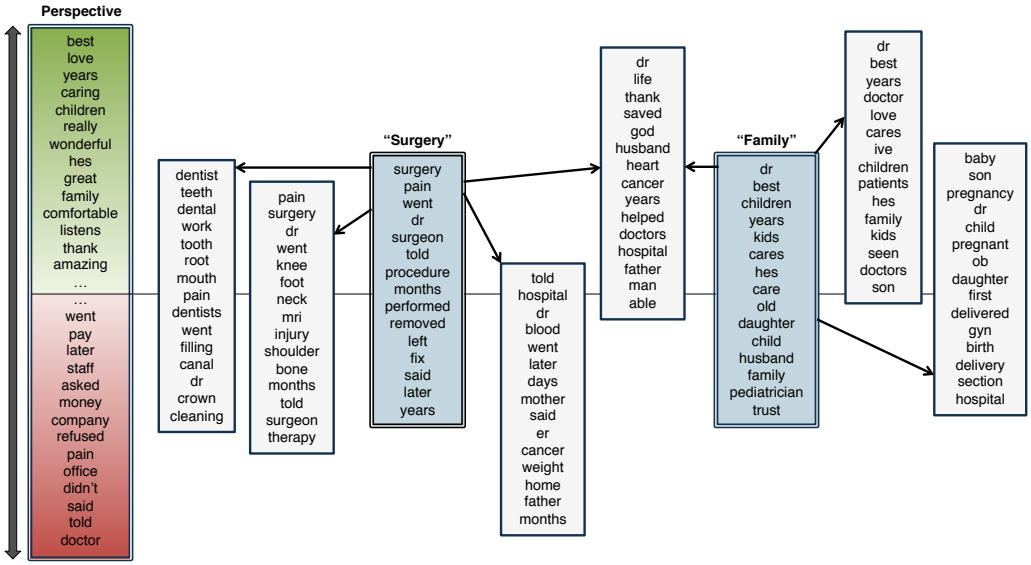



Figure 4: Examples of topics (gray boxes) and components (colored boxes) learned on the *Reviews* corpus with 20 topics and 5 components. Words with the highest and lowest values of $\omega^{(P)}$, the perspective component, are shown on the left, reflecting positive and negative sentiment words. The words with largest $\omega$ values in two supertopic components are also shown, with manually given labels. Arrows from components to topics indicate that the topic's word distribution draws from that component in its prior (with non-zero $\beta$ value). There are also implicit arrows from the perspective component to all topics (omitted for clarity). The vertical positions of topics reflect the topic's perspective value $\delta^{(P)}$. Topics centered above the middle line are more likely to occur in reviews with positive scores, while topics below the middle line are more likely in negative reviews. Note that this is a "soft" hierarchy because the tree structure is not strictly enforced, so some topics have multiple parent components. Table 3 shows how strict trees can be learned by tuning the annealing parameter.

measure the average coherence across all topics:

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{m=2}^{M}\sum_{l=1}^{m-1}\log\frac{DF(v_{km},v_{kl})+1}{DF(v_{kl})} \qquad (2)$$

where $DF(v,w)$ is the document frequency of words $v$ and $w$ (the number of documents in which they both occur), $DF(v)$ is the document frequency of word $v$, and $v_{ki}$ is the $i$th most probable word in topic $k$. We use the top $M = 20$ words. This metric is limited to measuring only the quality of word clusters, ignoring the potentially improved interpretability of organizing the data into certain structures. However, it is still useful as an alternative measure of performance and utility, independent of the models' predictive abilities.

Using these three metrics, we compared to several variants (denoted in bold) of the **full model** to understand how the different parts of the model affect performance:

- Variants that contain the hierarchy components but not the perspective component (**Hierarchy only**), and vice versa (**Perspective only**).
- The "hierarchy only" model using only document components $\delta$ and no topic components. This is a **PAM-style** model because it exhibits similar behavior to PAM (§4.4). We also compared to the original **PAM** model.
- The "hierarchy only" model using only topic components $\omega$ and no document components. This is a **SCTM-style** model because it exhibits similar behavior to SCTM (§4.2).
- The full model where $\alpha^{(P)}$ is learned rather than given as input. This is a **FLDA-style** model that has similar behavior to FLDA (§4.3). We also compared to the original **FLDA** model.
- The "perspective only" model but without the $\omega^{(P)}$ topic component, so the attribute value affects only the topic distributions and not the word distributions. This is identical to the **DMR** model of Mimno and McCallum (2008) (§4.5).
- A model with no components except for the bias vectors $\omega^{(B)}$ and $\delta^{(B)}$. This is equivalent to **LDA** with optimized hyperparameters **(learned)**. We also experimented with using **fixed** symmetric hyperparameters, using values suggested by Griffiths and Steyvers (2004): $50/K$ and 0.01 for topic and word distributions.

To put the results in context, we also compare to two types of baselines: (1) "bag of words" baselines, where we measure the perplexity of add-one smoothed unigram language models, we measure

| | *Debates* | | | *Reviews* | | |
|---|---|---|---|---|---|---|
| Model | Perplexity | Prediction error | Coherence | Perplexity | Prediction error | Coherence |
| Full model | †**1555.5** ± 2.3 | †0.615 ± 0.001 | -342.8 ± 0.9 | †1421.3 ± 8.4 | †**0.787** ± 0.006 | -512.7 ± 1.6 |
| Hierarchy only | †1561.8 ± 1.4 | 0.620 ± 0.002 | -342.6 ± 1.1 | †1457.2 ± 6.9 | †0.804 ± 0.007 | -509.1 ± 1.9 |
| Perspective only | †1567.3 ± 2.3 | †**0.613** ± 0.002 | -342.1 ± 1.2 | †**1413.7** ± 2.2 | †0.800 ± 0.002 | -512.0 ± 1.7 |
| SCTM-style | 1572.5 ± 1.6 | 0.620 ± 0.002 | †-335.8 ± 1.1 | 1504.0 ± 1.9 | †0.837 ± 0.002 | †-490.8 ± 0.9 |
| PAM-style | †1567.4 ± 1.9 | 0.620 ± 0.002 | -347.6 ± 1.4 | †1440.4 ± 2.7 | †0.835 ± 0.004 | -542.9 ± 6.7 |
| FLDA-style | †1559.5 ± 2.0 | 0.617 ± 0.002 | -340.8 ± 1.4 | †1451.1 ± 5.4 | †0.809 ± 0.006 | -505.3 ± 2.3 |
| DMR | 1578.0 ± 1.1 | 0.618 ± 0.002 | -343.1 ± 1.0 | †1416.4 ± 3.0 | †0.799 ± 0.003 | -511.6 ± 2.0 |
| PAM | 1578.9 ± 0.3 | 0.622 ± 0.003 | †-336.0 ± 1.1 | 1514.8 ± 0.9 | †0.835 ± 0.003 | †-493.3 ± 1.2 |
| FLDA | 1574.1 ± 2.2 | 0.618 ± 0.002 | -344.4 ± 1.3 | 1541.9 ± 2.3 | 0.856 ± 0.003 | -502.2 ± 3.1 |
| LDA (learned) | 1579.6 ± 1.5 | 0.620 ± 0.001 | -342.6 ± 0.6 | 1507.9 ± 2.4 | 0.846 ± 0.002 | -501.4 ± 1.2 |
| LDA (fixed) | 1659.3 ± 0.9 | 0.622 ± 0.002 | -349.5 ± 0.8 | 1517.2 ± 0.4 | 0.920 ± 0.003 | -585.2 ± 0.9 |
| Bag of words | 2521.6 ± 0.0 | 0.617 ± 0.000 | †-196.2 ± 0.0 | 1633.5 ± 0.0 | 0.813 ± 0.000 | †-408.1 ± 0.0 |
| Naive baseline | 7426.0 ± 0.0 | 0.677 ± 0.000 | -852.9 ± 7.4 | 10158.0 ± 0.0 | 1.595 ± 0.000 | -795.2 ± 13.0 |

Table 2: Perplexity of held-out tokens and mean absolute error for attribute prediction using various models (± std. error). † indicates significant improvement ($p < 0.05$) over optimized LDA under a two-sided t-test.

the prediction error using bag of words features, and we measure coherence of the unigram distribution; (2) naive baselines, where we measure the perplexity of the uniform distribution over each dataset's vocabulary, the prediction error when simply predicting each attribute as the mean value in the training set, and the coherence of 20 randomly selected words (repeated for 10 trials).

Table 2 shows that the full SPRITE model substantially outperforms the LDA baseline at both predictive tasks. Generally, model variants with more structure perform better predictively.

The difference between **SCTM-style** and **PAM-style** is that the former uses only topic components (for word distributions) and the latter uses only document components (for the topic distributions). Results show that the structured priors are more important for topic than word distributions, since PAM-style has lower perplexity on both datasets. However, models with both topic and document components generally outperform either alone, including comparing the **Perspective only** and **DMR** models. The former includes both topic and document perspective components, while DMR has only a document level component.

**PAM** does not significantly outperform optimized LDA in most measures, likely because it updates the hyperparameters using a moment-based approximation, which is less accurate than our gradient-based optimization. **FLDA** perplexity is 2.3% higher than optimized LDA on *Reviews*, comparable to the 4% reported by Paul and Dredze (2012) on a different corpus. The **FLDA-style** SPRITE variant, which is more flexible, significantly outperforms FLDA in most measures.

The results are quite different under the coherence metric. It seems that topic components (which influence the word distributions) improve coherence over LDA, while document components worsen coherence. SCTM-style (which uses only topic components) does the best in both datasets, while PAM-style (which uses only documents) does the worst. PAM also significantly improves over LDA, despite worse perplexity.

The **LDA (learned)** baseline substantially outperforms **LDA (fixed)** in all cases, highlighting the importance of optimizing hyperparameters, consistent with prior research (Wallach et al., 2009a).

Surprisingly, many SPRITE variants also outperform the bag of words regression baseline, even though the latter was tuned to optimize performance using heavy $\ell_2$ regularization, which we applied only weakly (without tuning) to the topic model features. We also point out that the "bag of words" version of the coherence metric (the coherence of the top 20 words) is higher than the average topic coherence, which is an artifact of how the metric is defined: the most probable words in the corpus also tend to co-occur together in most documents, so these words are considered to be highly coherent when grouped together.

**Parameter Sensitivity** We evaluated the full model at the two predictive tasks with varying numbers of topics ($\{12,25,50,100\}$ for *Debates* and $\{5,10,20,40\}$ for *Reviews*) and components ($\{2,5,10,20\}$). Figure 5 shows that performance is more sensitive to the number of topics than components, with generally less variance among the latter. More topics improve performance monotonically on *Debates*, while performance declines at 40 topics on *Reviews*. The middle range of components (5–10) tends to perform better than too few (2) or too many (20) components.

Regardless of quantitative differences, the

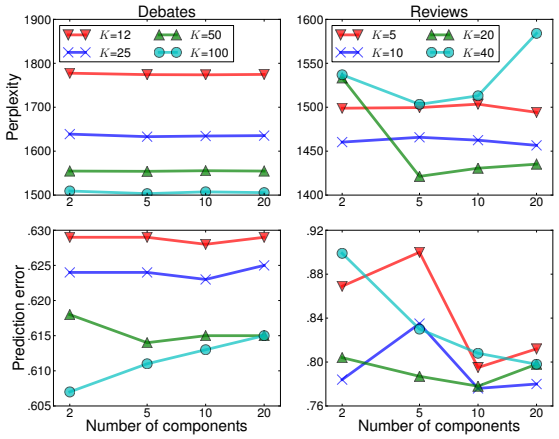



Figure 5: Predictive performance of full model with different numbers of topics $K$ across different numbers of components, represented on the x-axis (log scale).

| $\tau_t$ | | *Debates* | *Reviews* |
|---|---|---|---|
| 0.000 | (Sparse DAG) | 58.1% | 42.4% |
| 1.000 | (Soft Tree) | 93.2% | 74.6% |
| $1.001^t$ | (Hard Tree) | 99.8% | 99.4% |
| $1.003^t$ | (Hard Tree) | 100% | 100% |

Table 3: The percentage of indicator values that are sparse (near 0 or 1) when using different annealing schedules.

choice of parameters may depend on the end application and the particular structures that the user has in mind, if interpretability is important. For example, if the topic model is used as a visualization tool, then 2 components would not likely result in an interesting hierarchy to the user, even if this setting produces low perplexity.

**Structured Sparsity** We use a relaxation of the binary $\mathbf{b}$ that induces a "soft" tree structure. Table 3 shows the percentage of $\mathbf{b}$ values which are within $\epsilon = .001$ of 0 or 1 under various annealing schedules, increasing the inverse temperature $\tau$ by 0.1% after each iteration (i.e. $\tau_t = 1.001^t$) as well as 0.3% and no annealing at all ($\tau = 1$). At $\tau = 0$, we model a DAG rather than a tree, because the model has no preference that $\mathbf{b}$ is sparse. Many of the values are binary in the DAG case, but the sparse prior substantially increases the number of binary values, obtaining fully binary structures with sufficient annealing. We compare the DAG and tree structures more in the next subsection.

### 7.3 Structure Comparison

The previous subsection experimented with models that included a variety of structures, but did not provide a comparison of each structure in isolation, since most model variants were part of a complex joint model. In this section, we experiment with the basic SPRITE model for the three structures described in §3: a **DAG**, a **tree**, and a **factored** forest. For each structure, we also experiment with each type of component: **document**, **topic**, and both types (**combined**).

For this set of experiments, we included a third dataset that does not contain a perspective value:

- *Abstracts:* A set of 957 abstracts from the ACL anthology (97,168 tokens; 8,246 types). These abstracts have previously been analyzed with FLDA (Paul and Dredze, 2012), so we include it here to see if the factored structure that we explore in this section learns similar patterns.

Based on our sparsity experiments in the previous subsection, we set $\tau_t = 1.003^t$ to induce hard structures (tree and factored) and $\tau = 0$ to induce a DAG. We keep the same parameters as the previous subsection: $K$=50 and $C$=10 for *Debates* and $K$=20 and $C$=5 for *Reviews*. For the factored structures, we use two factors, with one factor having more components than the other: 3 and 7 components for *Debates*, and 2 and 3 components for *Reviews* (the total number of components across the two factors is therefore the same as for the DAG and tree experiments). The *Abstracts* experiments use the same parameters as with *Debates*.

Since the *Abstracts* dataset does not have a perspective value to predict, we do not include prediction error as a metric, instead focusing on held-out perplexity and topic coherence (Eq. 2). Table 4 shows the results of these two metrics.

Some trends are clear and consistent. Topic components always hurt perplexity, while these components typically improve coherence, as was observed in the previous subsection. It has previously been observed that perplexity and topic quality are not correlated (Chang et al., 2009). These results show that the choice of components depends on the task at hand. Combining the two components tends to produce results somewhere in between, suggesting that using both component types is a reasonable "default" setting.

Document components usually improve perplexity, likely due to the nature of the document completion setup, in which half of each document is held out. The document components capture correlations between topics, so by inferring the components that generated the first half of the document, the prior is adjusted to give more probability to topics that are likely to occur in the unseen second half. Another interesting trend is that the

| | Perplexity | | | Coherence | | |
|---|---|---|---|---|---|---|
| | DAG | Tree | Factored | DAG | Tree | Factored |
| | *Debates* | | | | | |
| Document | 1572.0 ± 0.9 | 1568.7 ± 2.0 | 1566.8 ± 2.0 | -342.9 ± 1.2 | -346.0 ± 0.9 | -343.2 ± 1.0 |
| Topic | 1575.0 ± 1.5 | 1573.4 ± 1.8 | 1559.3 ± 1.5 | -342.4 ± 0.6 | -339.2 ± 1.7 | **-333.9** ± 0.9 |
| Combined | 1566.7 ± 1.7 | 1559.9 ± 1.9 | **1552.5** ± 1.9 | -342.9 ± 1.3 | -342.6 ± 1.2 | -340.3 ± 1.0 |
| | *Reviews* | | | | | |
| Document | 1456.9 ± 3.8 | **1446.4** ± 4.0 | 1450.4 ± 5.5 | -512.2 ± 4.6 | -527.9 ± 6.5 | -535.4 ± 7.4 |
| Topic | 1508.5 ± 1.7 | 1517.9 ± 2.0 | 1502.0 ± 1.9 | -500.1 ± 1.2 | -499.0 ± 0.9 | **-486.1** ± 1.5 |
| Combined | 1464.1 ± 3.3 | 1455.1 ± 5.6 | 1448.5 ± 8.5 | -504.9 ± 1.4 | -527.8 ± 6.1 | -535.5 ± 8.2 |
| | *Abstracts* | | | | | |
| Document | 3107.7 ± 7.7 | **3089.5** ± 9.1 | 3098.7 ± 10.2 | -393.2 ± 0.8 | -390.8 ± 0.9 | -392.8 ± 1.5 |
| Topic | 3241.7 ± 2.1 | 3455.9 ± 10.2 | 3507.4 ± 9.7 | -389.0 ± 0.8 | -388.8 ± 0.7 | **-332.2** ± 1.1 |
| Combined | 3200.8 ± 3.5 | 3307.2 ± 7.8 | 3364.9 ± 19.1 | -373.1 ± 0.8 | -360.6 ± 0.9 | -342.3 ± 0.9 |

Table 4: Quantitative results for different structures (columns) and different components (rows) for two metrics (± std. error) across three datasets. The best (structure, component) pair for each dataset and metric is in bold.

factored structure tends to perform well under both metrics, with the lowest perplexity and highest coherence in a majority of the nine comparisons (i.e. each row). Perhaps the models are capturing a natural factorization present in the data.

To understand the factored structure qualitatively, Figure 6 shows examples of components from each factor along with example topics that draw from all pairs of these components, learned on *Abstracts*. We find that the factor with the smaller number of components (left of the figure) seems to decompose into components representing the major themes or disciplines found in ACL abstracts, with one component expressing computational approaches (top) and the other expressing linguistic theory (bottom). The third component (not shown) has words associated with speech, including {*spoken*, *speech*, *recognition*}.

The factor shown on the right seems to decompose into different research topics: one component represents semantics (top), another syntax (bottom), with others including morphology (top words including {*segmentation*, *chinese*, *morphology*}) and information retrieval (top words including {*documents*, *retrieval*, *ir*}).

Many of the topics intuitively follow from the components of these two factors. For example, the two topics expressing vector space models and distributional semantics (top left and right) both draw from the "computational" and "semantics" components, while the topics expressing ontologies and question answering (middle left and right) draw from "linguistics" and "semantics".

The factorization is similar to what had been previously been induced by FLDA. Figure 3 of Paul and Dredze (2012) shows components that look similar to the computational methods and linguistic theory components here, and the factor with the largest number of components also decomposes by research topic. These results show that SPRITE is capable of recovering similar structures as FLDA, a more specialized model. SPRITE is also much more flexible than FLDA. While FLDA strictly models a one-to-one mapping of topics to each pair of components, SPRITE allows multiple topics to belong to the same pair (as in the semantics examples above), and conversely SPRITE does not require that all pairs have an associated topic. This property allows SPRITE to scale to larger numbers of factors than FLDA, because the number of topics is not required to grow with the number of all possible tuples.

## 8 Related Work

Our topic and perspective model is related to supervised hierarchical LDA (SHLDA) (Nguyen et al., 2013), which learns a topic hierarchy while also learning regression parameters to associate topics with feature values such as political perspective. This model does not explicitly incorporate perspective-specific word priors into the topics (as in our factorized approach). The regression structure is also different. SHLDA is a "downstream" model, where the perspective value is a response variable conditioned on the topics. In contrast, SPRITE is an "upstream" model, where the topics are conditioned on the perspective value. We argue that the latter is more accurate as a generative story (the emitted words depend on the author's perspective, not the other way around). Moreover, in our model the perspective influences both the word and topic distributions (through the topic and document components, respectively).

Inverse regression topic models (Rabinovich and Blei, 2014) use document feature values (such as political ideology) to alter the parameters of the

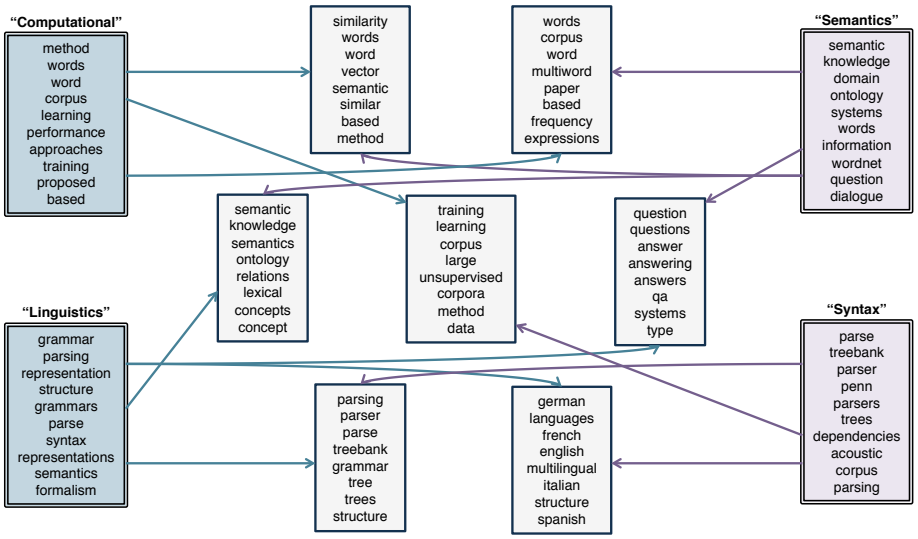



Figure 6: Examples of topics (gray boxes) and components (colored boxes) learned on the *Abstracts* corpus with 50 topics using a factored structure. The components have been grouped into two factors, one factor with 3 components (left) and one with 7 (right), with two examples shown from each. Each topic prior draws from exactly one component from each factor.

topic-specific word distributions. This is an alternative to the more common approach to regression based topic modeling, where the variables affect the topic distributions rather than the word distributions. Our SPRITE-based model does both: the document features adjust the prior over topic distributions (through $\delta$), but by tying together the document and topic components (with $\beta$), the document features also affect the prior over word distributions. To the best of our knowledge, this is the first topic model to condition both topic and word distributions on the same features.

The topic aspect model (Paul and Girju, 2010a) is also a two-dimensional factored model that has been used to jointly model topic and perspective (Paul and Girju, 2010b). However, this model does not use structured priors over the parameters, unlike most of the models discussed in §4.

An alternative approach to incorporating user preferences and expertise are interactive topic models (Hu et al., 2013), a complimentary approach to SPRITE.

## 9 Discussion and Conclusion

We have presented SPRITE, a family of topic models that utilize structured priors to induce preferred topic structures. Specific instantiations of SPRITE are similar or equivalent to several existing topic models. We demonstrated the utility of SPRITE by constructing a single model with many different characteristics, including a topic hierarchy, a factorization of topic and perspective, and supervision in the form of document attributes. These structures were incorporated into the priors of both the word and topic distributions, unlike most prior work that considered one or the other. Our experiments explored how each of these various model features affect performance, and our results showed that models with structured priors perform better than baseline LDA models.

Our framework has made clear advancements with respect to existing structured topic models. For example, SPRITE is more general and offers simpler inference than the shared components topic model (Gormley et al., 2010), and SPRITE allows for more flexible and scalable factored structures than FLDA, as described in earlier sections. Both of these models were motivated by their ability to learn interesting structures, rather than their performance at any predictive task. Similarly, our goal in this study was not to provide state of the art results for a particular task, but to demonstrate a framework for learning structures that are richer than previous structured models. Therefore, our experiments focused on understanding how SPRITE compares to commonly used models with similar structures, and how the different variants compare under different metrics.

Ultimately, the model design choice depends on the application and the user needs. By unifying such a wide variety of topic models, SPRITE can serve as a common framework for enabling model exploration and bringing application-specific preferences and structure into topic models.

## Acknowledgments

We thank Jason Eisner and Hanna Wallach for helpful discussions, and Viet-An Nguyen for providing the Congressional debates data. Michael Paul is supported by a Microsoft Research PhD fellowship.

## References

D. Andrzejewski, X. Zhu, and M. Craven. 2009. Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In *ICML*.

R. Balasubramanyan and W. Cohen. 2013. Regularization of latent variable models to obtain sparsity. In *SIAM Conference on Data Mining*.

D. Blei and J. Lafferty. 2007. A correlated topic model of Science. *Annals of Applied Statistics*, 1(1):17–35.

D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. 2003a. Hierarchical topic models and the nested Chinese restaurant process. In *NIPS*.

D. Blei, A. Ng, and M. Jordan. 2003b. Latent Dirichlet allocation. *JMLR*.

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

J. Duchi, E. Hazan, and Y. Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12:2121–2159.

J. Eisenstein, A. Ahmed, and E. P. Xing. 2011. Sparse additive generative models of text. In *ICML*.

M.R. Gormley, M. Dredze, B. Van Durme, and J. Eisner. 2010. Shared components topic models. In *NAACL*.

T. Griffiths and M. Steyvers. 2004. Finding scientific topics. In *Proceedings of the National Academy of Sciences of the United States of America*.

Y. Hu, J. Boyd-Graber, B. Satinoff, and A. Smith. 2013. Interactive topic modeling. *Machine Learning*, 95:423–469.

J. Kivinen and M.K. Warmuth. 1997. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132:1–63.

J.B. Lewis and K.T. Poole. 2004. Measuring bias and uncertainty in ideal point estimates via the parametric bootstrap. *Political Analysis*, 12(2):105–127.

W. Li and A. McCallum. 2006. Pachinko allocation: DAG-structured mixture models of topic correlations. In *International Conference on Machine Learning*.

D. Mimno and A. McCallum. 2008. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *UAI*.

D. Mimno, W. Li, and A. McCallum. 2007. Mixtures of hierarchical topics with Pachinko allocation. In *International Conference on Machine Learning*.

D. Mimno, H.M. Wallach, E. Talley, M. Leenders, and A. McCallum. 2011. Optimizing semantic coherence in topic models. In *EMNLP*.

V. Nguyen, J. Boyd-Graber, and P. Resnik. 2013. Lexical and hierarchical topic regression. In *Neural Information Processing Systems*.

M.J. Paul and M. Dredze. 2012. Factorial LDA: Sparse multi-dimensional text models. In *Neural Information Processing Systems (NIPS)*.

M.J. Paul and M. Dredze. 2013. Drug extraction from the web: Summarizing drug experiences with multi-dimensional topic models. In *NAACL*.

M. Paul and R. Girju. 2010a. A two-dimensional topic-aspect model for discovering multi-faceted topics. In *AAAI*.

M.J. Paul and R. Girju. 2010b. Summarizing contrastive viewpoints in opinionated text. In *Empirical Methods in Natural Language Processing*.

M.J. Paul, B.C. Wallace, and M. Dredze. 2013. What affects patient (dis)satisfaction? Analyzing online doctor ratings with a joint topic-sentiment model. In *AAAI Workshop on Expanding the Boundaries of Health Informatics Using AI*.

M. Rabinovich and D. Blei. 2014. The inverse regression topic model. In *International Conference on Machine Learning*.

D. Ramage, D. Hall, R. Nallapati, and C.D. Manning. 2009. Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP*.

N.A. Smith and J. Eisner. 2006. Annealing structural bias in multilingual weighted grammar induction. In *COLING-ACL*.

E.M. Talley, D. Newman, D. Mimno, B.W. Herr II, H.M. Wallach, G.A.P.C. Burns, M. Leenders, and A. McCallum. 2011. Database of NIH grants using machine-learned categories and graphical clustering. *Nature Methods*, 8(6):443–444.

N. Ueda and R. Nakano. 1998. Deterministic annealing EM algorithm. *Neural Networks*, 11(2):271–282.

B.C. Wallace, M.J. Paul, U. Sarkar, T.A. Trikalinos, and M. Dredze. 2014. A large-scale quantitative analysis of latent factors and sentiment in online doctor reviews. *Journal of the American Medical Informatics Association*, 21(6):1098–1103.

H.M. Wallach, D. Mimno, and A. McCallum. 2009a. Rethinking LDA: Why priors matter. In *NIPS*.

H.M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. 2009b. Evaluation methods for topic models. In *ICML*.

C. Wang and D. Blei. 2009. Decoupling sparsity and smoothness in the discrete hierarchical Dirichlet process. In *NIPS*.

M.D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. *CoRR*, abs/1212.5701.