Hierarchical Text Segmentation from Multi-Scale Lexical Cohesion

Jacob Eisenstein
Beckman Institute for Advanced Science and Technology
University of Illinois
Urbana, IL 61801
jacobe@illinois.edu

Abstract

This paper presents a novel unsupervised method for hierarchical topic segmentation. Lexical cohesion – the workhorse of unsupervised linear segmentation – is treated as a multi-scale phenomenon, and formalized in a Bayesian setting. Each word token is modeled as a draw from a pyramid of latent topic models, where the structure of the pyramid is constrained to induce a hierarchical segmentation. Inference takes the form of a coordinate-ascent algorithm, iterating between two steps: a novel dynamic program for obtaining the globally-optimal hierarchical segmentation, and collapsed variational Bayesian inference over the hidden variables. The resulting system is fast and accurate, and compares well against heuristic alternatives.

1 Introduction

Recovering structural organization from unformatted texts or transcripts is a fundamental problem in natural language processing, with applications to classroom lectures, meeting transcripts, and chatroom logs. In the unsupervised setting, a variety of successful systems have leveraged *lexical cohesion* (Halliday and Hasan, 1976) – the idea that topically-coherent segments display consistent lexical distributions (Hearst, 1994; Utiyama and Isahara, 2001; Eisenstein and Barzilay, 2008). However, such systems almost invariably focus on *linear* segmentation, while it is widely believed that discourse displays a hierarchical structure (Grosz and Sidner, 1986). This paper introduces the concept of *multi-scale lexical cohesion*, and leverages this idea in a Bayesian generative model for hierarchical topic segmentation.

The idea of multi-scale cohesion is illustrated by the following two examples, drawn from the Wikipedia entry for the city of Buenos Aires.

> There are over 150 city **bus** *lines* called **Colectivos** ... **Colectivos** in Buenos Aires do not have a fixed *timetable*, but *run* from 4 to several per *hour*, depending on the **bus** *line* and *time* of the day.

> The Buenos Aires **metro** has six *lines*, 74 **stations**, and 52.3 km of **track.** An expansion program is underway to extend existing *lines* into the outer *neighborhoods*. **Track** length is expected to reach 89 km...

The two sections are both part of a high-level segment on transportation. Words in bold are characteristic of the subsections (buses and trains, respectively), and do not occur elsewhere in the transportation section; words in italics occur throughout the high-level section, but not elsewhere in the article. This paper shows how multi-scale cohesion can be captured in a Bayesian generative model and exploited for unsupervised hierarchical topic segmentation.

Latent topic models (Blei et al., 2003) provide a powerful statistical apparatus with which to study discourse structure. A consistent theme is the treatment of individual words as draws from multinomial language models indexed by a hidden "topic" associated with the word. In latent Dirichlet allocation (LDA) and related models, the hidden topic for each word is unconstrained and unrelated to the hidden topic of neighboring words (given the parameters). In this paper, the latent topics are constrained to produce a hierarchical segmentation structure, as shown in Figure 1.

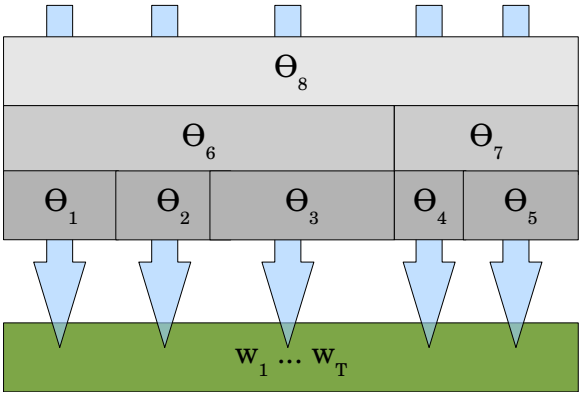

Figure 1: Each word w_t is drawn from a mixture of the language models located above t in the pyramid.

These structural requirements simplify inference, allowing the language models to be analytically marginalized. The remaining hidden variables are the scale-level assignments for each word token. Given marginal distributions over these variables, it is possible to search the entire space of hierarchical segmentations in polynomial time, using a novel dynamic program. Collapsed variational Bayesian inference is then used to update the marginals. This approach achieves high quality segmentation on multiple levels of the topic hierarchy.

Source code is available at `http://people.csail.mit.edu/jacobe/naacl09.html`.

2 Related Work

The use of lexical cohesion (Halliday and Hasan, 1976) in unsupervised topic segmentation dates back to Hearst's seminal TEXTTILING system (1994). Lexical cohesion was placed in a probabilistic (though not Bayesian) framework by Utiyama and Isahara (2001). The application of Bayesian topic models to text segmentation was investigated first by Blei and Moreno (2001) and later by Purver et al. (2006), using HMM-like graphical models for linear segmentation. Eisenstein and Barzilay (2008) extend this work by marginalizing the language models using the Dirichlet compound multinomial distribution; this permits efficient inference to be performed directly in the space of segmentations. All of these papers consider only linear topic segmentation; we introduce multi-scale lexical cohesion, which posits that the distribution of some words changes slowly with high-level topics, while others change rapidly with lower-level subtopics. This gives a principled mechanism to model hierarchical topic segmentation.

The literature on hierarchical topic segmentation is relatively sparse. Hsueh et al. (2006) describe a supervised approach that trains separate classifiers for topic and sub-topic segmentation; more relevant for the current work is the unsupervised method of Yaari (1997). As in TEXTTILING, cohesion is measured using cosine similarity, and agglomerative clustering is used to induce a dendrogram over paragraphs; the dendrogram is transformed into a hierarchical segmentation using a heuristic algorithm. Such heuristic approaches are typically brittle, as they include a number of parameters that must be hand-tuned. These problems can be avoided by working in a Bayesian probabilistic framework.

We note two orthogonal but related approaches to extracting nonlinear discourse structures from text. Rhetorical structure theory posits a hierarchical structure of discourse relations between spans of text (Mann and Thompson, 1988). This structure is richer than hierarchical topic segmentation, and the base level of analysis is typically more fine-grained – at the level of individual clauses. Unsupervised approaches based purely on cohesion are unlikely to succeed at this level of granularity.

Elsner and Charniak (2008) propose the task of conversation disentanglement from internet chatroom logs. Unlike hierarchical topic segmentation, conversational threads may be disjoint, with unrelated threads interposed between two utterances from the same thread. Elsner and Charniak present a supervised approach to this problem, but the development of cohesion-based unsupervised methods is an interesting possibility for future work.

3 Model

Topic modeling is premised on a generative framework in which each word w_t is drawn from a multinomial θ_{y_t}, where y_t is a *hidden topic* indexing the language model that generates w_t. From a modeling standpoint, linear topic segmentation merely adds the constraint that $y_t \in \{y_{t-1}, y_{t-1} + 1\}$. Segmentations that draw boundaries so as to induce compact, low-entropy language models will achieve a

high likelihood. Thus topic models situate lexical cohesion in a probabilistic setting.

For hierarchical segmentation, we take the hypothesis that lexical cohesion is a multi-scale phenomenon. This is represented with a pyramid of language models, shown in Figure 1. Each word may be drawn from any language model above it in the pyramid. Thus, the high-level language models will be required to explain words throughout large parts of the document, while the low-level language models will be required to explain only a local set of words. A hidden variable z_t indicates which level is responsible for generating the word w_t.

Ideally we would like to choose the segmentation $\hat{\mathbf{y}} = \text{argmax}_{\mathbf{y}} p(\mathbf{w}|\mathbf{y})p(\mathbf{y})$. However, we must deal with the hidden language models Θ and scale-level assignments $\mathbf{z}$. The language models can be integrated out analytically (Section 3.1). Given marginal likelihoods for the hidden variables $\mathbf{z}$, the globally optimal segmentation $\hat{\mathbf{y}}$ can be found using a dynamic program (Section 4.1). Given a segmentation, we can estimate marginals for the hidden variables, using collapsed variational inference (Section 4.2). We iterate between these procedures in an EM-like coordinate-ascent algorithm (Section 4.4) until convergence.

3.1 Language models

We begin the formal presentation of the model with some notation. Each word w_t is modeled as a single draw from a multinomial language model θ_j. The language models in turn are drawn from symmetric Dirichlet distributions with parameter α. The number of language models is written K; the number of words is W; the length of the document is T; and the depth of the hierarchy is L.

For hierarchical segmentation, the vector $\mathbf{y}_t$ indicates the segment index of t at each level of the topic hierarchy; the specific level of the hierarchy responsible for w_t is given by the hidden variable z_t. Thus, $y_t^{(z_t)}$ is the index of the language model that generates w_t.

With these pieces in place, we can write the observation likelihood,

$$p(\mathbf{w}|\mathbf{y}, \mathbf{z}, \Theta) = \prod_t^T p(w_t|\theta_{y_t^{(z_t)}}) = \prod_j^K \prod_{\{t:\{y_t^{(z_t)}=j\}} p(w_t|\theta_j),$$

where we have merely rearranged the product to group terms that are drawn from the same language model. As the goal is to obtain the hierarchical segmentation and not the language models, the search space can be reduced by marginalizing Θ. The derivation is facilitated by a notational convenience: $\mathbf{x}_j$ represents the lexical counts induced by the set of words $\{w_t : y_t^{(z_t)} = j\}$.

$$\begin{aligned} p(\mathbf{w}|\mathbf{y}, \mathbf{z}, \alpha) &= \prod_j^K \int d\theta_j p(\theta_j|\alpha) p(\mathbf{x}_j|\theta_j) \\ &= \prod_j^K p_{dcm}(\mathbf{x}_j; \alpha) \\ &= \prod_j^K \frac{\Gamma(W\alpha)}{\Gamma(\sum_i^W x_{ji} + \alpha)} \prod_i^W \frac{\Gamma(x_{ji} + \alpha)}{\Gamma(\alpha)}. \end{aligned} \quad (1)$$

Here, p_{dcm} indicates the Dirichlet compound multinomial distribution (Madsen et al., 2005), which is the closed form solution to the integral over language models. Also known as the multivariate Polya distribution, the probability density function can be computed exactly as a ratio of gamma functions. Here we use a symmetric Dirichlet prior α, though asymmetric priors can easily be applied.

Thus far we have treated the hidden variables $\mathbf{z}$ as observed. In fact we will compute approximate marginal probabilities $Q_{z_t}(z_t)$, written $\gamma_{t\ell} \equiv Q_{z_t}(z_t = \ell)$. Writing $\langle x \rangle_{Q_z}$ for the expectation of x under distribution Q_z, we approximate,

$$\langle p_{dcm}(\mathbf{x}_j; \alpha) \rangle_{Q_z} \approx p_{dcm}(\langle \mathbf{x}_j \rangle_{Q_z}; \alpha)$$

$$\langle x_j(i) \rangle_{Q_z} = \sum_{\{t:j \in \mathbf{y}_t\}} \sum_\ell^L \delta(w_t = i)\delta(y_t^{(\ell)} = j)\gamma_{t\ell},$$

where $x_j(i)$ indicates the count for word type i generated from segment j. In the outer sum, we consider all t for possibly drawn from segment j. The inner sum goes over all levels of the pyramid. The delta functions take the value one if the enclosed Boolean expression is true and zero otherwise, so we are adding the fractional counts $\gamma_{t\ell}$ only when $w_t = i$ and $y_t^{(\ell)} = j$.

3.2 Prior on segmentations

Maximizing the joint probability $p(\mathbf{w}, \mathbf{y}) = p(\mathbf{w}|\mathbf{y})p(\mathbf{y})$ leaves the term $p(\mathbf{y})$ as a prior on segmentations. This prior can be used to favor segmentations with the desired granularity. Consider a prior of the form $p(\mathbf{y}) = \prod_{\ell=1}^{L} p(\mathbf{y}^{(\ell)}|\mathbf{y}^{(\ell-1)})$; for notational convenience, we introduce a base level such that $y_t^{(0)} = t$, where every word is a segmentation point. At every level $\ell > 0$, the prior is a Markov process, $p(\mathbf{y}^{(\ell)}|\mathbf{y}^{(\ell-1)}) = \prod_t^T p(y_t^{(\ell)}|y_{t-1}^{(\ell)}, \mathbf{y}^{(\ell-1)})$.

The constraint $y_t^{(\ell)} \in \{y_{t-1}^{(\ell)}, y_{t-1}^{(\ell)} + 1\}$ ensures a linear segmentation at each level. To enforce hierarchical consistency, each $y_t^{(\ell)}$ can be a segmentation point only if t is also a segmentation point at the lower level $\ell - 1$. Zero probability is assigned to segmentations that violate these constraints.

To quantify the prior probability of legal segmentations, assume a set of parameters d_ℓ, indicating the expected segment duration at each level. If t is a valid potential segmentation point at level ℓ (i.e., $y_t^{(\ell-1)} = 1 + y_{t-1}^{(\ell-1)}$), then the prior probability of a segment transition is $r_\ell = d_{\ell-1}/d_\ell$, with $d_0 = 1$. If there are N segments in level ℓ and $M \geq N$ segments in level $\ell - 1$, then the prior $p(\mathbf{y}^{(\ell)}|\mathbf{y}^{(\ell-1)}) = r_\ell^N (1 - r_\ell)^{M-N}$, as long as the hierarchical segmentation constraint is obeyed.

For the purposes of inference it will be preferable to have a prior that decomposes over levels and segments. In particular, we do not want to have to commit to a particular segmentation at level ℓ before segmenting level $\ell + 1$. The above prior can be approximated by replacing M with its expectation $\langle M \rangle_{d_{\ell-1}} = T/d_{\ell-1}$. Then a single segment ranging from w_u to w_v (inclusive) will contribute $\log r_\ell + \frac{v-u}{d_{\ell-1}} \log(1 - r_\ell)$ to the log of the prior.

4 Inference

This section describes the inference for the segmentation $\mathbf{y}$, the approximate marginals Q_Z, and the hyperparameter α.

4.1 Dynamic programming for hierarchical segmentation

While the model structure is reminiscent of a factorial hidden Markov model (HMM), there are important differences that prevent the direct application of HMM inference. Hidden Markov models assume that the parameters of the observation likelihood distributions are available directly, while we marginalize them out. This has the effect of introducing dependencies throughout the state space: the segment assignment for each y_t contributes to lexical counts which in turn affect the observation likelihoods for many other t'. However, due to the left-to-right nature of segmentation, efficient inference of the optimal hierarchical segmentation (given the marginals Q_Z) is still possible.

Let $B^{(\ell)}[u, v]$ represent the log-likelihood of grouping together all contiguous words $w_u \dots w_{v-1}$ at level ℓ of the segmentation hierarchy. Using $\mathbf{x}_t$ to indicate a vector of zeros with one at the position w_t, we can express B more formally:

$$B^{(\ell)}[u, v] = \log p_{dcm}\left(\sum_{t=u}^{v} \mathbf{x}_t \gamma_{t\ell}\right) + \log r_\ell + \frac{v - u - 1}{d_{\ell-1}} \log(1 - r_\ell).$$

The last two terms are from the prior $p(\mathbf{y})$, as explained in Section 3.2. The value of $B^{(\ell)}[u, v]$ is computed for all u, all $v > u$, and all ℓ.

Next, we compute the log-likelihood of the optimal segmentation, which we write as $A^{(L)}[0, T]$. This matrix can be filled in recursively:

$$A^{(\ell)}[u, v] = \max_{u \leq t < v} B^{(\ell)}[t, v] + A^{(\ell-1)}[t, v] + A^{(\ell)}[u, t].$$

The first term adds in the log probability of the segment from t to v at level ℓ. The second term returns the best score for segmenting this same interval at a more detailed level of segmentation. The third term recursively segments the interval from u to t at the same level ℓ. The boundary case $A^{(\ell)}[u, u] = 0$.

4.1.1 Computational complexity

The sizes of A and B are each $\mathcal{O}(LT^2)$. The matrix A can be constructed by iterating through the layers and then iterating: u from 1 to T; v from $u+1$ to T; and t from u to $v+1$. Thus, the time cost for filling A is $\mathcal{O}(LT^3)$. For computing the observation likelihoods in B, the time complexity is $\mathcal{O}(LT^2W)$, where W is the size of the vocabulary – by keeping cumulative lexical counts, we can compute $B[u, v]$ without iterating from u to v.

Eisenstein and Barzilay (2008) describe a dynamic program for linear segmentation with a space complexity of $\mathcal{O}(T)$ and time complexities of $\mathcal{O}(T^2)$ to compute the A matrix and $\mathcal{O}(TW)$ to fill the B matrix.[1] Thus, moving to hierarchical segmentation introduces a factor of TL to the complexity of inference.

4.1.2 Discussion

Intuitively, efficient inference is possible because the location of each segment boundary affects the likelihood of only the adjoining segments at the same level of the hierarchy, and their "children" at the lower levels of the hierarchy. Thus, the observation likelihood at each level decomposes across the segments of the level. This is due to the left-to-right nature of segmentation – in general it is not possible to marginalize the language models and still perform efficient inference in HMMs. The prior (Section 3.2) was designed to decompose across segments – if, for example, $p(\mathbf{y})$ explicitly referenced the total number of segments, inference would be more difficult.

A simpler inference procedure would be a greedy approach that makes a fixed decision about the top-level segmentation, and then applies recursion to achieve segmentation at the lower levels. The greedy approach will not be optimal if the best top-level segmentation leads to unsatisfactory results at the lower levels, or if the lower levels could help to disambiguate high-level segmentation. In contrast, the algorithm presented here maximizes the overall score across all levels of the segmentation hierarchy.

[1] The use of dynamic programming for linear topic segmentation goes back at least to (Heinonen, 1998); however, we are aware of no prior work on dynamic programming for hierarchical segmentation.

4.2 Scale-level marginals

The hidden variable z_t represents the level of the segmentation hierarchy from which the word w_t is drawn. Given language models Θ, each w_t can be thought of as a draw from a Bayesian mixture model, with z_t as the index of the component that generates w_t. However, as we are marginalizing the language models, standard mixture model inference techniques do not apply. One possible solution would be to instantiate the maximum *a posteriori* language models after segmenting, but we would prefer not to have to commit to specific language models. Collapsed Gibbs sampling (Griffiths and Steyvers, 2004) is another possibility, but sampling-based solutions may not be ideal from a performance standpoint.

Recent papers by Teh et al. (2007) and Sung et al. (2008) point to an appealing alternative: collapsed variational inference (called latent-state variational Bayes by Sung et al.). Collapsed variational inference integrates over the parameters (in this case, the language models) and computes marginal distributions for the latent variables, $Q_{\mathbf{z}}$. However, due to the difficulty of computing the expectation of the normalizing term, these marginal probabilities are available only in approximation.

More formally, we wish to compute the approximate distribution $Q_{\mathbf{z}}(\mathbf{z}) = \prod_t^T Q_{z_t}(z_t)$, factorizing across all latent variables. As is typical in variational approaches, we fit this distribution by optimizing a lower bound on the data marginal likelihood $p(\mathbf{w}, \mathbf{z}|\mathbf{y})$ – we condition on the segmentation $\mathbf{y}$ because we are treating it as fixed in this part of the inference. The lower bound can be optimized by iteratively setting,

$$Q_{z_t}(z_t) \propto \exp\left\{\langle \log P(\mathbf{x}, \mathbf{z}|\mathbf{y})\rangle_{\sim Q_{z_t}}\right\},$$

indicating the expectation under $Q_{z_t'}$ for all $t' \neq t$. Due to the couplings across $\mathbf{z}$, it is not possible to compute this expectation directly, so we use the first-order approximation described in (Sung et al., 2008). In this approximation, the value $Q_{z_t}(z_t = \ell)$ – which we abbreviate as $\gamma_{t\ell}$ – takes the form of the likelihood of the observation w_t, given a modified mixture model. The parameters of the mixture model are based on the priors and the counts of $\mathbf{w}$

and γ for all $t' \neq t$:

$$\gamma_{t\ell} \propto \beta_\ell \frac{\tilde{x}_\ell^{\neg t}(w_t)}{\sum_i^W \tilde{x}_\ell^{\neg t}(i)} \quad (2)$$

$$\tilde{x}_\ell^{\neg t}(i) = \alpha_\ell(i) + \sum_{t' \neq t} \gamma_{t'\ell} \delta(w_{t'} = i). \quad (3)$$

The first term in equation 2 is the set of component weights β_ℓ, which are fixed at $1/L$ for all ℓ. The fraction represents the posterior estimate of the language models: standard Dirichlet-multinomial conjugacy gives a sum of counts plus a Dirichlet prior (equation 3). Thus, the form of the update is extremely similar to collapsed Gibbs sampling, except that we maintain the full distribution over z_t rather than sampling a specific value. The derivation of this update is beyond the scope of this paper, but is similar to the mixture of Bernoullis model presented in Section 5 of (Sung et al., 2008).

Iterative updates of this form are applied until the change in the lower bound is less than 10^{-3}. This procedure appears at step 5a of algorithm 1.

4.3 Hyperparameter estimation

The inference procedure defined here includes two parameters: α, the symmetric Dirichlet prior on the language models; and $\mathbf{d}$, the expected segment durations. The granularity of segmentation is considered to be a user-defined characteristic, so there is no "right answer" for how to set this parameter. We simply use the oracle segment durations, and provide the same oracle to the baseline methods where possible. As discussed in Section 6, this parameter had little effect on system performance.

The α parameter controls the expected sparsity of the induced language models; it will be set automatically. Given a segmentation $\mathbf{y}$ and hidden-variable marginals γ, we can maximize $p(\alpha, \mathbf{w}|\mathbf{y}, \gamma) = p_{dcm}(\mathbf{w}|\mathbf{y}, \gamma, \alpha)p(\alpha)$ through gradient descent. The Dirichlet compound multinomial has a tractable gradient, which can be computed using scaled counts, $\tilde{\mathbf{x}}_j = \sum_{t:y_t^{(z_t)}=j} \gamma_{tj}\mathbf{x}_t$ (Minka, 2003). The scaled counts are taken for each segment j across the entire segmentation hierarchy. The likelihood $p(\tilde{\mathbf{x}}|\alpha)$ then has the same form as equation 1, with the x_{ji} terms replaced by $\tilde{x}_{ji}$. The gradient of the log-likelihood is thus a sum across segments,

$$d\ell/d\alpha = \sum_j^K W(\Psi(W\alpha) - \Psi(\alpha)) + \sum_i^W \Psi(\tilde{x}_{ji} + \alpha) - \Psi(W\alpha + \sum_i^W \tilde{x}_{ji}).$$

Here, Ψ indicates the digamma function, which is the derivative of the log gamma function. The prior $p(\alpha)$ takes the form of a Gamma distribution with parameters $\mathcal{G}(1, 1)$, which has the effect of discouraging large values of α. With these parameters, the gradient of the Gamma distribution with respect to α is negative one. To optimize α, we interpose an epoch of L-BFGS (Liu and Nocedal, 1989) optimization after maximizing γ (Step 5b of algorithm 1).

Algorithm 1 Complete segmentation inference

1. **Input** text $\mathbf{w}$; expected durations d.
2. $\gamma \leftarrow$ INITIALIZE-GAMMA($\mathbf{w}$)
3. $\hat{\mathbf{y}} \leftarrow$ EQUAL-WIDTH-SEG($\mathbf{w}, d$)
4. $\alpha \leftarrow .1$
5. **Do**
   (a) $\gamma \leftarrow$ ESTIMATE-GAMMA($\hat{\mathbf{y}}, \mathbf{w}, \gamma, \alpha$)
   (b) $\alpha \leftarrow$ ESTIMATE-ALPHA($\hat{\mathbf{y}}, \mathbf{w}, \gamma$)
   (c) $\mathbf{y} \leftarrow$ SEGMENT($\mathbf{w}, \gamma, \alpha, d$)
   (d) **If** $\mathbf{y} = \hat{\mathbf{y}}$ **then return** $\mathbf{y}$
   (e) **Else** $\hat{\mathbf{y}} \leftarrow \mathbf{y}$

4.4 Combined inference procedure

The final inference procedure alternates between updating the marginals γ, the Dirichlet prior α, and the MAP segmentation $\hat{\mathbf{y}}$. Since the procedure makes hard decisions on α and the segmentations $\mathbf{y}$, it can be thought of as a form of Viterbi expectation-maximization (EM). When a repeated segmentation is encountered, the procedure terminates. Initialization involves constructing a segmentation $\hat{\mathbf{y}}$ in which each level is segmented uniformly, based on the expected segment duration d_ℓ. The hidden variable marginals γ are initialized randomly. While there is no guarantee of finding the global maximum, little sensitivity to the random initialization of γ was observed in preliminary experiments.

The dynamic program described in this section achieves polynomial time complexity, but $\mathcal{O}(LT^3)$

can still be slow when T is the number of word tokens in a large document such as a textbook. For this reason, we only permit segment boundaries to be placed at gold-standard sentence boundaries. The only change to the algorithm is that the tables A and B need contain only cells for each sentence rather than for each word token – hidden variable marginals are still computed for each word token. Implemented in Java, the algorithm runs in roughly five minutes for a document with 1000 sentences on a dual core 2.4 GHz machine.

5 Experimental Setup

Corpora The dataset for evaluation is drawn from a medical textbook (Walker et al., 1990).[2] The text contains 17083 sentences, segmented hierarchically into twelve high-level parts, 150 chapters, and 520 sub-chapter sections. Evaluation is performed separately on each of the twelve parts, with the task of correctly identifying the chapter and section boundaries. Eisenstein and Barzilay (2008) use the same dataset to evaluate linear topic segmentation, though they evaluated only at the level of sections, given gold standard chapter boundaries.

Practical applications of topic segmentation typically relate to more informal documents such as blogs or speech transcripts (Hsueh et al., 2006), as formal texts such as books already contain segmentation markings provided by the author. The premise of this evaluation is that textbook corpora provide a reasonable proxy for performance on less structured data. However, further clarification of this point is an important direction for future research.

Metrics All experiments are evaluated in terms of the commonly-used P_k and WindowDiff metrics (Pevzner and Hearst, 2002). Both metrics pass a window through the document, and assess whether the sentences on the edges of the window are properly segmented with respect to each other. WindowDiff is stricter in that it requires that the number of intervening segments between the two sentences be identical in the hypothesized and the reference segmentations, while P_k only asks whether the two sentences are in the same segment or not. This evaluation uses source code provided by Malioutov and Barzilay (2006).

Experimental system The joint hierarchical Bayesian model described in this paper is called HIERBAYES. It performs a three-level hierarchical segmentation, in which the lowest level is for sub-chapter sections, the middle level is for chapters, and the top level spans the entire part. This top-level has the effect of limiting the influence of words that are common throughout the document.

Baseline systems As noted in Section 2, there is little related work on unsupervised hierarchical segmentation. However, a straightforward baseline is a greedy approach: first segment at the top level, and then recursively feed each top-level segment to the segmenter again. Any linear segmenter can be plugged into this baseline as a "black box."

To isolate the contribution of joint inference, the greedy framework can be combined with a one-level version of the Bayesian segmentation algorithm described here. This is equivalent to BAYESSEG, which achieved the best reported performance on the linear segmentation of this same dataset (Eisenstein and Barzilay, 2008). The hierarchical segmenter built by placing BAYESSEG in a greedy algorithm is called GREEDY-BAYES.

To identify the contribution of the Bayesian segmentation framework, we can plug in alternative linear segmenters. Two frequently-cited systems are LCSEG (Galley et al., 2003) and TEXTSEG (Utiyama and Isahara, 2001). LCSEG optimizes a metric of lexical cohesion based on lexical chains. TEXTSEG employs a probabilistic segmentation objective that is similar to ours, but uses maximum *a posteriori* estimates of the language models, rather than marginalizing them out. Other key differences are that they set $\alpha = 1$, and use a minimum description length criterion to determine segmentation granularity. Both of these baselines were run using their default parametrization.

Finally, as a minimal baseline, UNIFORM produces a hierarchical segmentation with the ground truth number of segments per level and uniform duration per segment at each level.

Preprocessing The Porter (1980) stemming algorithm is applied to group equivalent lexical items. A set of stop-words is also removed, using the same

[2] The full text of this book is available for free download at http://onlinebooks.library.upenn.edu.

	chapter			section			average	
	# segs	P_k	WD	# segs	P_k	WD	P_k	WD
HIERBAYES	5.0	**.248**	**.255**	8.5	.312	.351	.280	**.303**
GREEDY-BAYES	19.0	.260	.372	19.5	**.275**	**.340**	**.268**	.356
GREEDY-LCSEG	7.8	.256	.286	52.2	.351	.455	.304	.371
GREEDY-TEXTSEG	11.5	.251	.277	88.4	.473	.630	.362	.454
UNIFORM	**12.5**	.487	.491	**43.3**	.505	.551	.496	.521

Table 1: Segmentation accuracy and granularity. Both the P_k and WindowDiff (WD) metrics are penalties, so lower scores are better. The **# segs** columns indicate the average number of segments at each level; the gold standard segmentation granularity is given in the UNIFORM row, which obtains this granularity by construction.

list originally employed by several competitive systems (Utiyama and Isahara, 2001).

6 Results

Table 1 presents performance results for the joint hierarchical segmenter and the three greedy baselines. As shown in the table, the hierarchical system achieves the top overall performance on the harsher WindowDiff metric. In general, the greedy segmenters each perform well at one of the two levels and poorly at the other level. The joint hierarchical inference of HIERBAYES enables it to achieve balanced performance at the two levels.

The GREEDY-BAYES system achieves a slightly better average P_k than HIERBAYES, but has a very large gap between its P_k and WindowDiff scores. The P_k metric requires only that the system correctly classify whether two points are in the same or different segments, while the WindowDiff metric insists that the exact number of interposing segments be identified correctly. Thus, the generation of spurious short segments may explain the gap between the metrics.

LCSEG and TEXTSEG use heuristics to determine segmentation granularity; even though these methods did not score well in terms of segmentation accuracy, they were generally closer to the correct granularity. In the Bayesian methods, granularity is enforced by the Markov prior described in Section 3.2. This prior was particularly ineffective for GREEDY-BAYES, which gave nearly the same number of segments at both levels, despite the different settings of the expected duration parameter d.

The Dirichlet prior α was selected automatically, but informal experiments with manual settings suggest that this parameter exerts a stronger influence on segmentation granularity. Low settings reflect an expectation of sparse lexical counts and thus encourage short segments, while high settings reflect an expectation of evenly-distributed counts and thus lead to long segments. Further investigation is needed on how best to control segmentation granularity in a Bayesian setting.

7 Discussion

While it is widely agreed that language often displays hierarchical topic structure (Grosz, 1977), there have been relatively few attempts to extract such structure automatically. This paper shows that the lexical features that have been successfully exploited in linear segmentation can also be used to extract a hierarchical segmentation, due to the phenomenon of multi-scale lexical cohesion. The Bayesian methodology offers a principled probabilistic formalization of multi-scale cohesion, yielding an accurate and fast segmentation algorithm with a minimal set of tunable parameters.

It is interesting to consider how multi-scale segmentation might be extended to finer-grain segments, such as paragraphs. The lexical counts at the paragraph level will be sparse, so lexical cohesion alone is unlikely to be sufficient. Rather it may be necessary to model discourse connectors and lexical semantics explicitly. The development of more comprehensive Bayesian models for discourse structure seems an exciting direction for future research.

Acknowledgments Thanks to Michel Galley, Igor Malioutov, and Masao Utiyama for making their topic segmentation systems publicly available, and to the anonymous reviewers for useful feedback. This research is supported by the Beckman Postdoctoral Fellowship.

References

David M. Blei and Pedro J. Moreno. 2001. Topic segmentation with an aspect hidden markov model. In *SIGIR*, pages 343–348.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Jacob Eisenstein and Regina Barzilay. 2008. Bayesian unsupervised topic segmentation. In *Proceedings of EMNLP*.

Micha Elsner and Eugene Charniak. 2008. You Talking to Me? A Corpus and Algorithm for Conversation Disentanglement. In *Proceedings of ACL*.

Michel Galley, Katheen McKeown, Eric Fosler-Lussier, and Hongyan Jing. 2003. Discourse segmentation of multi-party conversation. pages 562–569.

T.L. Griffiths and M. Steyvers. 2004. Finding scientific topics.

Barbara Grosz and Candace Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Barbara Grosz. 1977. The representation and use of focus in dialogue understanding. Technical Report 151, Artificial Intelligence Center, SRI International.

M. A. K. Halliday and Ruqaiya Hasan. 1976. *Cohesion in English*. Longman.

Marti A. Hearst. 1994. Multi-paragraph segmentation of expository text. In *Proceedings of ACL*, pages 9–16.

Oskari Heinonen. 1998. Optimal Multi-Paragraph Text Segmentation by Dynamic Programming. In *Proceedings of ACL*, pages 1484–1486.

P.Y. Hsueh, J. Moore, and S. Renals. 2006. Automatic segmentation of multiparty dialogue. In *Proccedings of EACL*.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45:503–528.

R.E. Madsen, D. Kauchak, and C. Elkan. 2005. Modeling word burstiness using the Dirichlet distribution. In *Proceedings of ICML*.

Igor Malioutov and Regina Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proceedings of ACL*, pages 25–32.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8:243–281.

Thomas P. Minka. 2003. Estimating a dirichlet distribution. Technical report, Massachusetts Institute of Technology.

Lev Pevzner and Marti A. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14:130–137.

M. Purver, T.L. Griffiths, K.P. Körding, and J.B. Tenenbaum. 2006. Unsupervised topic modelling for multiparty spoken discourse. In *Proceedings of ACL*, pages 17–24.

Jaemo Sung, Zoubin Ghahramani, and Sung-Yang Bang. 2008. Latent-space variational bayes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(12):2236–2242, Dec.

Y.W. Teh, D. Newman, and M. Welling. 2007. A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. In *NIPS*, volume 19, page 1353.

Masao Utiyama and Hitoshi Isahara. 2001. A statistical model for domain-independent text segmentation. In *Proceedings of ACL*, pages 491–498.

H. Kenneth Walker, W. Dallas Hall, and J. Willis Hurst, editors. 1990. *Clinical Methods : The History, Physical, and Laboratory Examinations*. Butterworths.

Y. Yaari. 1997. Segmentation of Expository Texts by Hierarchical Agglomerative Clustering. In *Recent Advances in Natural Language Processing*.