Joint Segmentation and Clustering in Text Corpuses

Sam Blasiak *  Sithu Sudarsan †  Huzefa Rangwala ‡

Abstract

In recent years, many private corporations and government organizations have digitized corpuses of legacy paper documents. Often, these organizations hope to take advantage of digital representations to transform costly manual tasks associated with paper archives into less-costly computer-assisted tasks. The most common approach toward automated information extraction is through inverted indexing systems that allow fast keyword searches. Keyword-based indexing, however, is ineffective for tasks that require information from higher-level contexts.

To allow for more effective information extraction from digital corpuses, we propose combining two common document processing tasks, (i) clustering and (ii) segmentation, into one process to simultaneously segment documents within a corpus and assign each segment to a category. We have developed a generative probabilistic model to accomplish this task, which we call the Joint Segmentation and Clustering (JSC) model.

From experiments measuring segmentation and clustering ability, we show that our model can accurately partition documents and assign meaningful categories to each partition. In addition, experiments tracking predictive perplexity show that our JSC model outperforms basic topic modeling approaches in terms of conciseness of the induced representation.

Keywords: Segmentation, Clustering, Topic-Modeling, Text-Mining

1 Introduction

A common problem faced by many organizations is in extracting useful information from large corpuses of physical documents. To avoid full manual processing, these documents are often first processed using an optical character recognition (OCR) system. Once digitized, documents can be indexed for searching, allowing fast reverse look-ups by keyword. The process of indexing alone, however, neglects important content structure across documents. For example, if an organization tracks medical devices, then many documents may share similar sections that discuss potential dangers of these devices. If accounted for, this content structure can greatly accelerate an organization's ability to extract relevant information from legacy files.

Figure 1 shows an example of a context where jointly segmenting and clustering sentences in a document could be useful. The example document is a Wikipedia article which contains a number of section headings. These headings, indicated by dashed outlines in the diagram, define boundaries between areas of text. In addition, we can view each section heading as a label applied to each block of text in the section. Text is divided into blocks (sentences or groups of sentences), indicated by solid outlines in the diagram. This document structure leads to the following inference problem: Given only the text (at block-level granularity) in the corpus, recover section boundaries and assign blocks of text into categories corresponding to their original associated section headings.

To make sense of digital documents from this domain, we propose combining two common document processing tasks, (i) clustering and (ii) segmentation, into one process to simultaneously segment documents within a corpus and assign each segment to a category. We have developed a generative probabilistic model to accomplish this task which we call the Joint Segmentation and Clustering (JSC) model.

The JSC model draws upon ideas from both segmentation models and topic models. From a segmentation model perspective, incorporating a document-specific prior, as in topic-modeling, over per-segment cluster assignments has the potential to improve segmentation by taking advantage of repeated patterns of words across disparate portions of the corpus. From a topic-modeling perspective, our model uses segments, rather than individual words, as the within-document exchangeable unit, taking advantage of higher-level document structure.
Figure 1: An excerpt from a Wikipedia article partitioned into segments based on section headings. Section headings also can be viewed as categories associated with blocks of text inside the section. Section headings are marked with dashed outlines and blocks of body text are indicated by solid outlines. The first section in the example article is marked in blue, while the second is marked in green. Given text from a set of documents at a block-level granularity, we would like to recover section boundaries and common section topics associated with each block.

An additional contribution of our work is in introducing a noise-based heuristic to the JSC inference algorithm. This heuristic allows the JSC model to escape undesirable areas in the topic space and improves in both clustering and segmentation ability.

In sets of experiments designed to measure both segmentation and clustering ability, we show that our model accurately partitions documents and assigns meaningful categories to each partition. In addition, experiments tracking predictive perplexity show that our JSC model can outperform basic topic modeling approaches in terms of conciseness of the induced representation.

2 Related Work

Document segmentation is a well-researched problem where an algorithm attempts to partition a text document or transcription of spoken discourse into semantically meaningful portions. The notion of lexical cohesion guides many unsupervised methods for text segmentation [6, 13]. Lexical cohesion postulates that subparts of a semantically meaningful text segment possess common features. These common features are typically chosen to be within-segment word counts. Unsupervised text segmentation models often employ Hidden Markov Models (HMMs) [6, 10] to either determine segment boundaries or assign units of text to a segment. The BayesSeg [6] algorithm is an example of this type of approach. BayesSeg postulates that the contents of each segment are drawn separately from Dirichlet Conditional Multinomial [6] distributions. Another approach involves using a Segmental Semi-Markov Model (SSMM) [13] to assign chains of textual units (e.g., sentences) to a segment. Unlike our JSC model, a common characteristic of these approaches is that they model the contents of each segment independently to capture cohesion within individual segments without introducing sharing between segments. Related supervised approaches include the Segmental Semi-Markov Random Field [11], a close relative of the Conditional Random Field [7].

Clustering is concerned with partitioning a dataset so that elements within each partition share common characteristics. One approach toward probabilistic clustering is to explain a collection of data by fitting a mixture of probability distributions to the dataset. A fundamental approach to mixture modeling techniques, such as Latent Dirichlet Allocation (LDA) [2], involves careful application of De Finetti’s theorem. From De Finetti’s theorem, if elements of a dataset are infinitely exchangeable, then they can be represented as a mixture that is conditionally independent on a hidden variable. LDA considers both documents within a corpus and words within a document to be infinitely exchangeable. These exchangeability assumptions lead to LDA’s characteristic model hierarchy. Like LDA, segmentation models consider documents within a corpus to be infinitely exchangeable. However, within a docu-
ment, segmentation models consider segments, i.e., cohesive chains of words or other text elements, rather than individual words, to be exchangeable.

Topic modeling is concerned with finding shared themes (topics) across documents that produce a concise representation of a corpus. Latent Dirichlet Allocation, for instance, both finds a set of topics (distributions over words in the vocabulary) and clusters the words in each document by topic. The Special Words with Background (SWB) model [3] is an extension of LDA that incorporates separate document-level and corpus-level topics. These extra topics allow the model to factor out “background topics” - topics that are either common across the every document in the corpus or are specific to individual documents. We incorporate this background model into our JSC model to allow finer granularity in assigning topics to segments.

Models that both segment a corpus and find repeated themes between segments are not as common as those that only segment documents or those that cluster text elements independently. One notable example of a model that performs both tasks is Chen et. al.’s Latent Permutations model [4]. In the Latent Permutations model, a set of topics are assigned to blocks of text through a permutation drawn from the Generalized Mallows Model. A salient different between the Latent Permutations model and the JSC model is in their respective problem settings; the Latent Permutations model is designed specifically for corporuses where segments with the same topic assignment occur only once per document while for the JSC model, we expect multiple segments to have the same topic assignment.

3 Joint Segmentation and Clustering

To jointly segment documents and cluster words within a corpus, we define the Joint Segmentation and Clustering (JSC) model. Unlike text segmentation models, which assume that each segment is generated independently, the JSC model assumes that subsets of segments share characteristics. The JSC model captures these shared characteristics by associating a cluster assignment with each segment. This cluster assignment indexes a generating distribution over the words within the associated segment.

To describe how the JSC model operates, we indicate the $b^{th}$ word in the $t^{th}$ text block from document $d$ using $x_{d,t,b}$. Each segment in document $d$ is described by its start position, $t_{d,i}$, and end position $t_{d,i+1}$, with the final segment in the document ending at text block $t_{d,S_d}$, where $S_d$ is the number of segments in document $d$. The $t^{th}$ segment in the $d^{th}$ document is associated with the set of text blocks $x_{d,t-1:t_{d,i+1}-1}$: (we use ‘,’ as in Matlab slice notation). We provide a full list of model parameters in Table 1. A plate diagram of the model is given in Figure 2.

The joint probability of a single segment and associated cluster assignment in the JSC model is given by

$$p(x_{d,t-1:t_{d,i+1}}, \pi, \Theta, S_d) = \prod_{d,t,b} \pi_{d,t,b} p(x_{d,t,b} | \pi_{d,t,b})$$

where $p(x_{d,t-1:t_{d,i+1}} | \Theta)$ can be varied according to the dataset domain. For the simplest case, we generate each segment’s contents using a multinomial distribution over words:

$$p(x_{d,t,b} | \pi_{d,t,b}) = \frac{\pi_{d,t,b}}{\sum_{c=1}^{C} \pi_{d,t,b}}$$

where the index $b$ is over words within a block of text. In this case, $\Theta = \{ \phi \}$ and $\phi_{c,m}$ is the probability of generating word $m$ given a segment cluster assignment of $c$. This multinomial distribution, $\phi_{c,:}$, is equivalent to a distribution over words given a topic in the LDA model [2].

The marginal probability of the entire corpus is given by

$$p(x; S, H, A) = \int_{\Theta} \int_{\pi} p(\Theta | H) \prod_{d} \prod_{t} p(\pi_{d,t} | A) p(x_{d,t} | \pi_{d,t})$$

where the sum is over $S_d$, all possible numbers of segmentations, $c_d$, all possible cluster assignments of segments, and $t_d$, all possible ways to segment the document into $S_d$ parts. We use a uniform prior on numbers of segmentations and placement of segment boundaries, $p(t_{d,1:S_d-1}) \sim \mathcal{U}$. We also include a Dirichlet prior over the document cluster probabilities, $p(\pi_{d,t} | A) \sim \text{Dir}(\alpha)$, and a distribution over segment-level parameters, $p(\Theta | H)$.

Table 1: Description of JSC model parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_{d,t,b}$</td>
<td>the $b^{th}$ word in the $t^{th}$ block in document $d$</td>
</tr>
<tr>
<td>$t_{d,i}$</td>
<td>the starting position of the $i^{th}$ segment in document $d$</td>
</tr>
<tr>
<td>$t_{d,i+1}$</td>
<td>the ending position of the $i^{th}$ segment in document $d$</td>
</tr>
<tr>
<td>$S_d$</td>
<td>the number of segments in document $d$</td>
</tr>
<tr>
<td>$\Pi$</td>
<td>the set of all possible segmentations of document $d$</td>
</tr>
<tr>
<td>$p_{d,i}$</td>
<td>the set of parameters used to compute the probability of the words in a segment given that the segment was assigned to cluster $c$</td>
</tr>
<tr>
<td>$H$</td>
<td>the prior distribution over $\pi$</td>
</tr>
<tr>
<td>$N_d$</td>
<td>the probability of assigning a segment to cluster $c$</td>
</tr>
<tr>
<td>$\phi_{c,m}$</td>
<td>the probability of generating word $m$ in cluster $c$</td>
</tr>
<tr>
<td>$\phi_{c,:}$</td>
<td>the probability of generating word in from global background topic</td>
</tr>
<tr>
<td>$\alpha_0$</td>
<td>a Dirichlet prior on the variable contained in the superscripted parentheses, e.g., $\alpha_{0,t}$ is the Dirichlet prior parameter for every row of $x$</td>
</tr>
</tbody>
</table>

Copyright © SIAM. Unauthorized reproduction of this article is prohibited.
3.1 Background Distributions If multiple cluster-word distributions are significantly similar, then the JSC model may assign similar probabilities to many different segmentations. Because a sampled segmentation is used to sample cluster-word distributions and vice-versa, poor initializations are self-reinforcing, and the JSC model, as initially defined, often fails to converge to either useful clusters or segmentations. To encourage distinct cluster-word distributions, we follow Chenudgunga et al. [3] and factor out both a global background topic and a document-specific background topic. Including background topics, the per-segment probability becomes

\[
p(x_{d,t,i} ; t_{d,i} = 1 ; c_{d,i}, \Theta) = \prod_{i=1}^{t_{d,i}} \prod_{t'=t_{d,i}} \xi_{d,t',b} \phi_{d,i,t',b} \eta x_{d,t',b} \psi_{d,t',b} \]

where the index \( b \) is over words within a block of text, and \( o_{d,t,b} \) is a selection variable drawn from a multinomial distribution over possible selections, \( \xi_{d} \), that indicates whether the \( b^{th} \) word in the \( t^{th} \) text block of the \( d^{th} \) document is associated with the segment’s topic (\( o_{d,t,b} = 1 \)), the global background topic (\( o_{d,t,b} = 2 \)), or the document specific background topic (\( o_{d,t,b} = 3 \)). \( \psi_{d} \) indicates the background topic distribution for document \( d \) and \( \eta \) indicates the global background topic distribution. We draw \( \xi_{d} \), \( \psi_{d} \), and \( \eta \) from Dirichlet prior distributions with uniform prior parameters of \( a(\xi), a(\psi) \), and \( a(\eta) \) respectively.

Adding background topic distributions to the model has an additional positive effect on topic granularity for the cluster-level topics. Without the background distributions, assigning a segment to a cluster assigns every word in the segment to the cluster’s topic. This segment-level granularity introduces excessive noise to the JSC model’s topic distributions, especially for small corpuses with large text blocks. Ideally, we would only like to associate words with a cluster’s topic that are significant to the meaning of that particular cluster. The background word model is an effective way of allowing finer-grained cluster-word assignments which, in turn, allows the model to isolate meaningful words in each segment. Due to the effectiveness of using background distributions, we incorporate them into the models used in all of our experiments.

3.2 Inference We conduct inference in the JSC model by interleaving two separate Gibbs sampling steps. In the first step, we work directly with the multinomial distributions, \( \pi, \xi, \phi, \psi \), and \( \eta \), and perform blocked Gibbs sampling of both segmentations and cluster assignments using the forward algorithm [13] combined with a backward sampling step. This step enables efficient sampling of segmentations at the expense of less efficient sampling of cluster assignments. The forward algorithm is a dynamic programming algorithm that employs a forward recurrence, \( a_{d,t}(c) \). This recurrence computes the sum of all segmentation probabilities for document \( d \) that start at the first text block and end at text block \( t \) with cluster assignment \( c \) as
follows:
\[ \alpha_{d,t}(c) = \sum_{1 \leq t' < t} p(x_{d,t',t}, c_{d,t'}|t, t', \pi, \Theta) \alpha_{d,t'-1} \]
\[ \alpha_{d,t} = \sum_c \alpha_{d,t}(c), \quad \alpha_{d,0} = 1 \]

For inference, we sample a segmentation in the backward direction using the previously-computed forward recurrence:
\[ p(c, t'|t) \propto \alpha_{d,t'}(c) \forall t' < t \]

In the second inference step, we fix segment boundaries in each document and run a predefined number of collapsed Gibbs sampling steps on the cluster assignments for each segment. This inference step is similar to the collapsed Gibbs sampling algorithm for the SWB model [3] and is described in Section 1 of the Supplementary Material.

After sampling cluster assignments we can then sample \( \pi_{d,c} \) from its Dirichlet-distributed conditional posterior:
\[ \pi_{d,c} \propto \prod_c \pi_{d,c,a}^{n_{d,c}+a(c)-1} \]

where \( \pi_{d,c} \) indicates the probability distribution over clusters in document \( d \) and \( n_{d,c} \) indicates the number of segments assigned to cluster \( c \) in document \( d \). In addition, because we choose \( p(x_{d,t}, t+1, c_{d,t+1}|-1, c_{d,t}) \) to be conjugate to \( p(\pi|\Theta) \) with \( \Theta = \{ \phi \} \) (Equation 3.2), we can also sample \( \phi \) from a Dirichlet distributed conditional posterior. Finally, the global and document-specific background distributions, \( \eta \) and \( \psi \), are also Dirichlet distributed with conjugate priors, allowing us to sample these from their conditional posteriors as well. Figure 3 shows an outline of the Gibbs sampling algorithm for the JSC model.

3.3 Noise-based Heuristic

A drawback of MCMC inference on the JSC model is that it tends to get stuck in potentially unfavorable areas of the parameter space. This undesirable behavior results from topic-word distributions, \( \phi_{c,:} \), converging to settings that cause inference to strongly favor assigning a block to a specific topic. We have developed a heuristic, inspired by the Hierarchical Dirichlet Process (HDP) [12], that introduces more volatility in cluster assignments during the inference process. This heuristic changes the behavior of the sampler so that at each iteration, text segments have an increased probability of being assigned to a small number of randomly selected clusters. Introducing this targeted noise thus allows the model to escape areas of the topic space that produce poor segmentations.

The noise-based heuristic operates as follows: At each iteration we draw a new set of non-uniform prior parameters on the \( \pi_{d,c} \)'s from a Dirichlet distribution. Specifically, at every Gibbs sampling iteration of the JSC inference algorithm, we set \( a^{(c)} = a^{(c)} \beta \), where \( a^{(c)} \) is a large multiplicative factor, and \( \beta \sim \text{Dirichlet} \left( a^{(d)} \right) \), with \( a^{(d)} \) set to a small positive value. This procedure is similar to the Gibbs sampling step in the auxiliary variable sampler for a JSC-like model where an HDP prior is placed on the foreground topics\(^2\). For the noise-based heuristic, however, instead of sampling auxiliary HDP dish assignment counts (which are used to sample \( \beta \) in the HDP[12]), we simply set these counts to zero. This technique has the effect of including a prior distribution on cluster assignments from a separate large, fictitious corpus, which changes at every iteration of the algorithm.

A disadvantage of applying this heuristic is that the model is no longer a proper probabilistic model. However, the noise-based heuristic leads to segmentations and clusterings that clearly outperform the JSC model in the segmentation and clustering domain, as shown in the Section 4. Although we provide no theoretical guarantees of convergence using the heuristic, we found experimentally that our heuristic converges to a stationary distribution in a similar manner as the original JSC model.

4 Experimental Results

We performed several sets of experiments to assess the performance of our model on text segmentation, recovering original segment category assignments, and finding concise representations of a corpus. For the clustering task, we compared our model against CLUTO\(^3\), a similarity-based clustering algorithm and against the Latent Permutations model [4]. CLUTO took into account only word counts within blocks of text and performed no segmentation. For the segmentation task, we compared our algorithms against BayesSeg [6], which does not associate segments with cluster assignments, and against the Latent Permutations model [4]. We assessed our model’s ability to represent a text corpus by computing the predictive perplexity on a held-out set of documents. We compared perplexities after inference with our model to perplexities from the SWB model [3]. For experiments with the JSC algorithm, cluster assignments and segmentation boundaries were computed

\(^2\)In a set of unpublished experiments, we found that placing an HDP prior on foreground topics gives similar performance to the standard JSC model.

\(^3\)http://glaros.dtc.umn.edu/gkhome/views/cluto
by taking the Viterbi path over segments and clusters given the final set of parameters from each inference algorithm.

In all experiments using the SWB and JSC model, we initialized \( a^{(v)} = .1, a^{(b)} = 0.25, a^{(e)} = 0.01, a^{(c)} = 0.3, \) and \( a^{(s)} = .1. \) For the noise-based heuristic, we set \( a^{(c)} = 10^7 \) and \( a^{(d)} = 1.0. \) Experiments on the SWB and JSC models and experiments using the noise-based heuristic were run for 100 Gibbs sampling iterations for the segmentation and clustering experiments and 500 iterations for the perplexity experiments. For the perplexity experiments, we sampled all Dirichlet prior parameters on both the JSC and SWB models using the auxiliary variable technique described by Newman et. al. [8]. For this auxiliary variable technique, we placed a Gamma distributed hypervariable with both shape and rate set to \( 10^{-4} \) on all Dirichlet prior parameters.

Experiments on the Latent Permutations model were run for 10000 iterations with parameter settings described by Chen et. al. [4]. CLUTO and BayesSeg were run with their default parameter settings until convergence. We ran only the dynamic programming portion of BayesSeg inference, as additional Metropolis-Hastings steps decreased segmentation quality.

In segmentation and clustering experiments, we set the number of clusters for each approach to be significantly smaller than the number of ground truth clusters in the datasets. For all algorithms where the number of clusters could be specified, we ran experiments with 10 and 20 possible cluster assignments. These cluster assignments were indicative of the range of the best performing numbers of clusters under these models.

4.1 Datasets We conducted experiments on four datasets. To assess segmentation and segment category assignment performance, we trained our model on the WikiCities and WikiElements datasets used to assess the Latent Permutations model of Chen et. al. [4]. These datasets consisted of sets of Wikipedia articles on cities and chemical elements respectively. Ground truth segment boundaries were set to section boundaries in the Wikipedia articles, and text blocks were labeled by ground truth cluster assignments according the heading of the section containing the block (see Figure 1). Both the text of the section headings and the boundaries between sections were discarded in the datasets presented to each inference algorithm and used only to evaluate segmentation and clustering ability.

We also constructed the WikiPeople dataset from Wikipedia articles about people. This dataset is larger than both the WikiCities and WikiElements datasets but was constructed similarly. In the WikiPeople dataset, we only retained sections whose headings appeared more than ten times in the entire corpus. Table 2 lists the number of documents and clusters in each of the Wiki datasets.

We used the Associated Press (AP) dataset to assess perplexity. AP news articles do not include reference segmentations, so the AP dataset is not appropriate for assessing segmentation ability.

We preprocessed each dataset by removing stop words, stemming words, removing very frequently or infrequently occurring words, and replacing strings of digits with the text "NUMBER".

4.2 Clustering Table 3 shows a comparison of clustering results between CLUTO, the Latent Permutations model [4], and our models. To assess the performance of each approach, we computed three types of scores using a confusion matrix, \( C. \) The value of each entry, \( C_{r,h}, \) is given by the number of sentences in reference cluster, \( r, \) which are also contained in hypothetical cluster, \( h. \) We computed the average recall as \( \frac{\sum_{r,h} \max_h C_{r,h}}{\sum_{r,h} C_{r,h}} \), the average precision as \( \frac{\sum_{r,h} \max_h C_{r,h}}{\sum_{r,h} C_{r,h}} \), and the F1-score as \( 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \). We assessed overall performance using the F1 score.

Both the Latent Permutations model and our models produce better F1 scores than CLUTO on all datasets, indicating that the joint segmentation task adds useful information toward predicting cluster assignments. The Latent Permutations model outperforms the JSC model on all datasets. However, the noise-based technique outperforms the Latent Permutations model on the WikiCities dataset at a significance level of \( p < .031 \) (comparing the best performing settings from each algorithm) and the WikiElements dataset \( (p < .320) \) but is outperformed by the Latent Permutations model on the WikiPeople dataset \( (p < .150) \). We performed significance testing using the approximate randomization technique following Chen et. al. [4]. Approximate randomization computes P-

---

\(^{4}\)http://groups.csail.mit.edu/tbg/code/mallows/

\(^{5}\)http://www.cs.gmu.edu/~sblasiak/wikipeople.tar.gz

\(^{6}\)http://www.cs.princeton.edu/~blei/lda-c/ap.tar.gz

---

Table 2: A description of datasets used to test our algorithms. Vocabulary sizes and total word counts were computed after preprocessing the raw text.
The strong performance of the noise-based heuristic compared to the basic JSC model can be explained by the difference in topics discovered by each model. Topics in the JSC model are more correlated than topics using the noise-based technique. For instance, the average log cosine distances between foreground topics from the WikiElements dataset using the JSC model ranged between -2 and -3, while average log cosine distances using the noise-based heuristic decreased to between -3 and -4. More correlation between topics causes an increase in the number of inferred segment boundaries because similar topics will split singly-themed sections into more than one segment. These extra segment boundaries are the cause of the discrepancies in segmentation scores between the basic JSC model and the noise-based approach.

Evidence of the extra segment boundaries can also be seen by comparing the $P_k$ and WD scores from the basic JSC model against scores from BayesSeg. For all datasets, the basic JSC model produces better $P_k$ scores than BayesSeg but worse WD scores. This behavior is due to the WD scoring system penalizing occurrences of multiple short segments within a window which are not penalized by the $P_k$ scoring system.

### 4.4 Representation ability

To assess the ability of the JSC model in finding compressed representations of text, we compared the perplexity of the JSC model to the Special Words with Background (SWB) model [3]. The SWB model is similar to LDA, but, like the JSC model, incorporates both global and document-specific background topics. We assessed both training and test set (predictive) perplexity on the Associated Press dataset. To do so, for each experiment, we

### Table 4: A comparison of segmentation results between our model and the Latent Permutations model.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th># Clusters</th>
<th>Recall</th>
<th>Precision</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latent Permutations</td>
<td>10</td>
<td>0.640 (0.023)</td>
<td>0.561 (0.022)</td>
<td>0.277 (0.013)</td>
</tr>
<tr>
<td>JSC</td>
<td>10</td>
<td>0.438 (0.026)</td>
<td>0.443 (0.025)</td>
<td>0.280 (0.014)</td>
</tr>
<tr>
<td>Noise-Based</td>
<td>10</td>
<td>0.381 (0.023)</td>
<td>0.416 (0.027)</td>
<td>0.331 (0.014)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.501 (0.017)</td>
<td>0.574 (0.015)</td>
<td>0.337 (0.008)</td>
</tr>
</tbody>
</table>

The strong performance of the noise-based heuristic compared to the basic JSC model can be explained by the difference in topics discovered by each model. Topics in the JSC model are more correlated than topics using the noise-based technique. For instance, the average log cosine distances between foreground topics from the WikiElements dataset using the JSC model ranged between -2 and -3, while average log cosine distances using the noise-based heuristic decreased to between -3 and -4. More correlation between topics causes an increase in the number of inferred segment boundaries because similar topics will split singly-themed sections into more than one segment. These extra segment boundaries are the cause of the discrepancies in segmentation scores between the basic JSC model and the noise-based approach.

Evidence of the extra segment boundaries can also be seen by comparing the $P_k$ and WD scores from the basic JSC model against scores from BayesSeg. For all datasets, the basic JSC model produces better $P_k$ scores than BayesSeg but worse WD scores. This behavior is due to the WD scoring system penalizing occurrences of multiple short segments within a window which are not penalized by the $P_k$ scoring system.
Table 4: A comparison of the performance of the BayesSeg algorithm, the Latent Permutations [4] model, our JSC model, and our noise-based heuristic on the text segmenting task for three Wikipedia datasets. We compare $P_k$ and WindowDiff (WD) scores (smaller scores are better) on (a) the WikiCities dataset (b) the WikiElements dataset, and (c) the WikiPeople dataset. We report the means and standard deviations over 10 experiments from the JSC and noise-based approaches.

![Table 4](image)

The JSC and SWB models on the training sets computed the perplexity:  
$$\text{perplexity} \left(X^{(\text{train})}\right) = \exp\left(\frac{-\log p\left(X^{(\text{train})}\right)}{|X|}\right),$$  
where $|X|$ indicates the number of words in corpus $X$. Then, given the topic assignments and segmentations from the training set, we trained each model on the test set and computed the test set perplexity:  
$$\text{perplexity} \left(X^{(\text{test})}\right) = \exp\left(\frac{-\log p\left(X^{(\text{test})}\right)}{|X^{(\text{test})}|}\right).$$

We performed two sets of experiments. In the first set of experiments, we assessed each model’s performance under varying amounts of training data. We created 10 training/test set splits of the corpus with the first 10%, 20%, ..., 90% of the documents selected as the training set for each split and the rest selected for the test set. Each experiment was repeated 10 times with 15 topics for both the JSC and SWB models. Figure 4a shows a comparison of the perplexity from each model at each training/test set split. For each split, the JSC model produces lower perplexity than the SWB model.

In the second set of experiments, we assessed the effect of different numbers of topics or cluster assignments in each model. For these experiments, the training set consisted of the first half of the AP documents, and the test set consisted of the second half. We then varied the number of topics in the SWB model and clusters in the JSC model from 5 to 100. In these experiments, as shown in Figure 4b, the JSC model gives lower perplexities than the SWB model for all topic settings. These experiments indicate that introducing segmentations allows the JSC model to produce more concise representations of text corpuses than topic models that assume full exchangeability of individual words within a document. We do not include perplexity results from the noise-based technique because it is not associated with a true probabilistic model.

5 Conclusions

We have developed the JSC model, which combines ideas from text segmentation and topic modeling, to extract useful information from unannotated text corpuses. To improve the model’s performance, we have introduced a noise-based heuristic that allows the sampling algorithm to escape from undesirable areas of the topic space. In experiments, we show that these models perform well at both segmentation and in inferring original segment categories. Specifically, the JSC model combined with the noise-based technique produces comparable performance to the state-of-the-art Latent Permutations model in segmentation and clustering tasks on Wikipedia datasets. The Latent Permutations model, however, operates on a restricted problem domain compared to the JSC model, as it was designed to model documents containing only single occurrences of separate section categories.

A key component in improving performance of the JSC model is in controlling which words are assigned to background topics in the model. As such, more advanced approaches toward modeling background topics [3] may lead to more powerful methods for jointly segmenting and clustering corpuses.

Acknowledgements

This project was supported in part by an appointment to the research participation program at the Center for Device and Radiological Health administered by the Oskar Ridge Institute for Science and Education through and interagency agreement between the U.S. Department of Energy and the U.S. Food and Drug Administration.
Figure 4: A comparison of average training and test set perplexities between the JSC and the SWB models on the Associated Press dataset (a) as the percentage of training set documents increases and (b) as the number of topics increases. Each experiment was run for 10 trials for each setting. For all experiments, the JSC model produces lower perplexities than the SWB model. The gray outline indicates one standard deviation.

References