The Report committee for Kayla Hope Schaefer
Certifies that this is the approved version of the following report:

Document Clustering with Nonparametric Hierarchical Topic Modeling

APPROVED BY

SUPERVISING COMMITTEE:

Sinead Williamson, Supervisor

Mingyuan Zhou

Document Clustering with Nonparametric Hierarchical Topic Modeling

by

Kayla Hope Schaefer, B.A.

Report

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in Statistics

THE UNIVERSITY OF TEXAS AT AUSTIN
May 2015

Document Clustering with Nonparametric Hierarchical Topic Modeling

by

Kayla Hope Schaefer, M.S.Stat
The University of Texas at Austin, 2015
Supervisor: Sinead Williamson

Since its introduction, topic modeling has been a fundamental tool in analyzing corpus structures. While the Relational Topic Model provides a way to link, and subsequently cluster, documents together as an extension of the original Latent Dirichlet Allocation (LDA) model, this paper seeks to form a document clustering model for the nonparametric alternative to LDA, the Dirichlet Process. As the structure of Shakespeare's tragedies is the focus of this work, we specifically cluster documents while modeling the text using a Hierarchical Dirichlet Process (HDP), which allows for a mixture model with shared mixture components, in order to capture the natural topic clustering within a play. Using collapsed Gibbs sampling, the effectiveness of the clustered HDP is compared against that of LDA and an HDP without document clustering. This is done using both log perplexity and a qualitative assessment of the returned topics. Furthermore, clustering is performed and analyzed individually on speeches from each of ten tragedies, as well as with a combined corpus of acts.

Table of Contents

List of Tables

List of Figures

Section 1 Motivation

In machine learning, the overarching goals are usually pattern identification and prediction. These can be applied in a variety of fields, from computer vision to social networks. In natural language processing, a specific and fundamental focus is topic modeling, the clustering of words within a set of documents (Blei, Ng, & Jordan, 2003). While such a cluster is unable to label the broad themes of a paper in the sense that a paper is about 'neurology' or 'cryptography', one could see that a document has a high percentage of words from the topic that contains 'neurotransmitters', 'reaction', 'cortical', 'difference', 'effect', 'visual', and 'cortex' as likely about neurology, which could allow for a qualitative grouping of the documents based on their topics. (See Millar, Peterson, and Mendenhall (2009) for such an approach using Latent Dirichlet Allocation and self-organizing maps.)

Currently, in order to group documents together, we must rely on a post-hoc approach wherein documents are grouped based on the proportions of topics they have. However, in this work I will show it is instead possible to add a level to the clustering process, allowing the algorithm to find similar documents. This additionally will improve the topic modeling part of the algorithm, as the topics are created using information about which documents are similar. For this report, we focus on Shakespeare's tragedies, modeling the topics using a Hierarchical Dirichlet Process (HDP), and then combining the HDP with additional document-level clustering. This clustered HDP (cHDP) is more reflective of the underlying structure of Shakespeare's

plays, with individual speeches in scenes in acts. We apply such a multi-level model alongside less complex alternatives, in order to be able to qualitatively assess the quality of the clusters as well as a quantitative likelihood assessment.

1.1 Stopwords in Shakespeare

Before beginning analysis of a text, it is useful to first clean the words in the corpus. One of the two primary components of this process are stemming, or reducing a word to its root form - or stem, such as 'jogging' to 'jog', or 'apples' to 'appl'. For Python, the nltk.stem package[1] provides a variety of stemming algorithms, specifically Porter's Snowball Stemmer (Porter, 2001). This stemmer not only has a low error rate compared to other truncators, but was built from an algorithm that offers a language-independent approach to stemming (Jivani, 2011), and as such, does not rely on a dictionary look-up for stemming, a practice which would be muddied by the verbiage of Shakespeare.

The secondary component is the removal of stopwords, words which occur frequently across all documents, and thus have no distinguishing value. In modern English (as well as Shakespearean English) common stopwords are 'the', 'a', and 'and'. However, Shakespearean English also contains words that would not be caught by a traditional stopword filter, such as 'thee'. To address this, a list of Shakespearean stopwords was created using the corpus itself.

This was done using the TF-IDF score, which contrasts the Term Frequency, or

[1] http://www.nltk.org/api/nltk.stem.html

the number of times a word appears in a single document, with the Inverse Document Frequency, which measures how common a word is among all documents, (Salton & McGill, 1983). Common words are given a low score, with rare, or document-specific words given a high score. Together, the TF-IDF score of a word reflects the strength of its relationship with the document it appears in. For word w in document d, this is mathematically given by

$$score(w, d) = n_{w,d} * log(\frac{D}{D_w}) \tag{1.1}$$

where D is the number of documents, $n_{w,d}$ is the number of times w appears in that document, and D_w is the number of documents that contain word w. The 500 lowest scoring words were used as a supplement to the traditional English stopword list, and are provided in the appendix.

Section 2 Topic Models

Existing topic modeling algorithms share a common core structure, which can be seen in detail in the following sections. Topic models represent documents as a mixture model over a latent set of 'word topics' or distributions over words. Given that some number of topics are present in a corpus, each document contains different proportions of words from these topics. This structure is easily reflected in the fundamental generative model, Latent Dirichlet Allocation (LDA), first introduced by Blei et al. in 2003.

2.1 Latent Dirichlet Allocation

LDA is a three level model of words, topics, and documents where each document in a corpus is modeled as a finite mixture of topics, and each topic is modeled as a finite mixture over words. LDA assumes that words and documents are exchangeable, i.e. that their orders are arbitrary. This is commonly referred to as a 'bag of words' arrangement. It also assumes that the number of topics, K, is known and fixed.

LDA can be used to generate a model corpus as follows:

- Take K topics, and let the kth topic distribution be $\boldsymbol{\phi}_k \sim Dir(\lambda)$, where λ and $\boldsymbol{\phi}_k$ are vectors of length V.
- For document j, the distribution over topics takes the form of $\boldsymbol{\theta_j} \sim Dir(\alpha)$

where α and $\boldsymbol{\theta}_j$ are vectors of length K.

- For each word w in the document, draw topic $z_w \sim Mult(\boldsymbol{\theta_j})$ and word $w \sim Mult(\boldsymbol{\phi_{z_w}})$.

This corpus will have J documents and N_j words for each document.

Inference in LDA requires computing the posterior distribution of the latent variables in a document:

$$p(\theta, \mathbf{z}|\mathbf{w}, \alpha, \lambda) = \frac{p(\boldsymbol{\theta}, \mathbf{z}, \mathbf{w}|\alpha, \lambda)}{p(\mathbf{w}|\alpha, \lambda)} \tag{2.1}$$

This equation is intractable, but can be approximated in several ways, the most common of which are Gibbs sampling (S. Geman & D. Geman, 1984) and variational inference (Jordan, Ghahramani, Jaakkola, & Saul, 1998). For this paper, collapsed Gibbs sampling was used, which solves the updates only for $p(\mathbf{z}|\boldsymbol{\theta}, \boldsymbol{\phi}, \lambda, \alpha, \mathbf{w})$. The inference algorithm is described in section 3.1.

2.1.1 Document Linking with Topic Models

LDA serves well as a general framework for topic modeling but can be improved upon if other observed information is available. It is often possible to link groups of documents to one another by citations, authors, or journals. Intuitively, documents citing one another are likely to share topics. The Relational Topic Model (RTM) developed by Chang and Blei (2009) uses this added information to improve on LDA's uncovering of the latent data structure. RTM builds upon the LDA framework with

an additional binary link variable between all pairs of documents, which is zero if there is no link between documents and one if there is a link. Given a corpus of J documents with W words we add $y_{jj'} \sim \psi(*|\mathbf{z_j}, \mathbf{z_{j'}})$, which is the link variable between documents d and d', to the LDA parameterization. ψ represents the link probability function that estimates the probability of a link between two documents.

Steyvers, Smyth, and Griffiths (2004) took this linking further by grouping documents by the same author. They specifically create an author model, where authors are samples from a multinomial distribution over topics. They expand the LDA generation to

- Take the set of corpus authors, $\mathbf{A}$, and the sets of document co-authors A_j.
- Take K topics, and let the kth topic distribution be $\boldsymbol{\phi}_k \sim Dir(\lambda)$.
- For author a_j in document j, the distribution over topics takes the form of $\boldsymbol{\theta}_{\boldsymbol{a_j}} \sim Dir(\alpha)$.
- For each word w in the document, draw author $a_j \sim Mult(A_j)$, topic $z_w \sim Mult(\boldsymbol{\theta}_{\boldsymbol{a_j}})$ and word $w \sim Mult(\boldsymbol{\phi}_{\boldsymbol{z_w}})$,

so each word comes from the distribution from one of the topics associated with the word's author. This is similar to the process used in this paper, however this model requires not only knowledge of the complete set of authors before starting inference, but also requires that the document authors are observed for each word.

Without a specific topic model, latent document clustering has been performed on a general language model, as shown in Xu and Croft (1999). With an eye

towards improving information retrieval, they approximate clusters with a two-pass K-means clustering on the documents in the corpus. To determine similarity between a document and a possible cluster, they use the Kullback-Leibler divergence:

$$KL(j,c) = \sum_{w_i} \frac{n_{j,w_i}}{M} log \frac{n_{j,w_i}/M}{(n_{c,w_i} + n_{j,w_i})/(C+J)} \tag{2.2}$$

where C is the number of clusters, n_{j,w_i} is again the count of word i in document j, and n_{c,w_i} is the count of word i in cluster c. As the goal of Xu and Croft was speed over prediction, this approach does not account for the effect of multiple topics in a document. However, the document clustering could be improved by allowing for the cross-document grouping of words in topics, and does turn out to be helpful in informing predictions about individual words, as will be shown in this work.

2.2 Dirichlet Process Mixture Model

In the models just listed, we specifically look for K topics, but choosing this value is a non-trivial task involving a priori knowledge of the corpus. Ideally, this value should be informed by the data itself. This is made possible by turning the k-dimensional mixture into an infinite-dimensional Dirichlet Process Mixture Model, as described in Ferguson (1973). The transition can be seen more easily with an alternate parameterization of the LDA model as follows:

$$\begin{aligned} \boldsymbol{\theta_j} &\sim Dir(\alpha), \\ z_w|\boldsymbol{\theta}_j &\sim Mult(\boldsymbol{\theta}_j), \\ \phi_k &\sim G_0(\lambda), \\ w|z_w, \boldsymbol{\phi} &\sim F(\phi_{z_w}) \end{aligned} \tag{2.3}$$

In this model, first a cluster is drawn from the multinomial distribution given by $\boldsymbol{\theta}_j$, then a word w is generated from its cluster by sampling from the distribution $F(\phi_{z_w})$. We still retain z_w as a latent variable indicating the topic for word w. We can think of $\boldsymbol{\theta}_j$ now as the mixture weights given to each of the possible distributions, and it is given a symmetric Dirichlet prior with a K-length hyperparameter vector with terms α/K. Finally, a common prior $G_0(\lambda)$ is introduced for the cluster parameters, which corresponds to the previous topic distributions.

By letting K go to infinity, we see the dissolution of the individual α/K, requiring us to replace $\boldsymbol{\theta}$ with its infinite-dimensional analog $\boldsymbol{\pi}$. The process as a whole remains the same, giving the system

$$\begin{aligned} \boldsymbol{\pi_j} &\sim GEM(\alpha), \\ z_w|\boldsymbol{\pi}_j &\sim Mult(\boldsymbol{\pi}_j), \\ \phi_k &\sim G_0(\lambda), \\ w|z_w, \boldsymbol{\phi} &\sim F(\phi_{z_w}) \end{aligned} \tag{2.4}$$

2.2.1 Stick breaking

The GEM distribution can be thought of through a process known as stick breaking, where a stick of length 1 is broken into infinitely many pieces (Sethuraman, 1991). First we break the stick according to the proportion B_1, where $B_1 \sim Beta(1, \alpha)$, with the piece B_1 then assigned to π_1. The remaining piece is then broken according to the proportion B_2, in a similar fashion. This process repeats ad infinitum. Overall,

the process can be represented by the equations

$$\begin{aligned} B_k &\sim Beta(1, \alpha), \\ \pi_k &= B_k \prod_{t=1}^{k-1} (1 - B_t) \end{aligned} \tag{2.5}$$

We then can say $\boldsymbol{\pi}$ is distributed $GEM(\alpha)$.

Next we define the random probability measure G such that $G(x) = \sum_{k=1}^{\infty} \pi_k \delta_{x_k}(x)$ where δ_{x_k} is a probability mass at x_k. In such a way we can take $\boldsymbol{\phi}$ where $\phi_k \sim G$. We now say that G is distributed according to a Dirichlet Process, $G \sim DP(\alpha, G_0)$ where α is a concentration parameter and G_0 is the base measure. Having G_0 as a common parameter for all the ϕ_k ensures that the support of each ϕ_k is the same, as in Equation (2.4). In such a way we can also represent this new infinite mixture topic model as

$$\begin{aligned} G &\sim DP(\alpha, G_0), \\ \phi_k &\sim G, \\ w|z_w, \boldsymbol{\phi} &\sim F(\phi_{z_w}) \end{aligned} \tag{2.6}$$

The graphical representation of all parameterizations can be compared in Figure 2.1.

2.2.2 Chinese Restaurant Process

A slightly more intuitive way of illustrating the Dirichlet Process is through what is known as the Chinese Restaurant Process (CRP), which defines the distribution over the Dirichlet partitions of the DP (Aldous, 1985). In this analogy, we assume that there is an empty Chinese restaurant with infinitely many tables. The first customer sits at the first table and chooses a dish. Afterwards, each new customer sits at one of

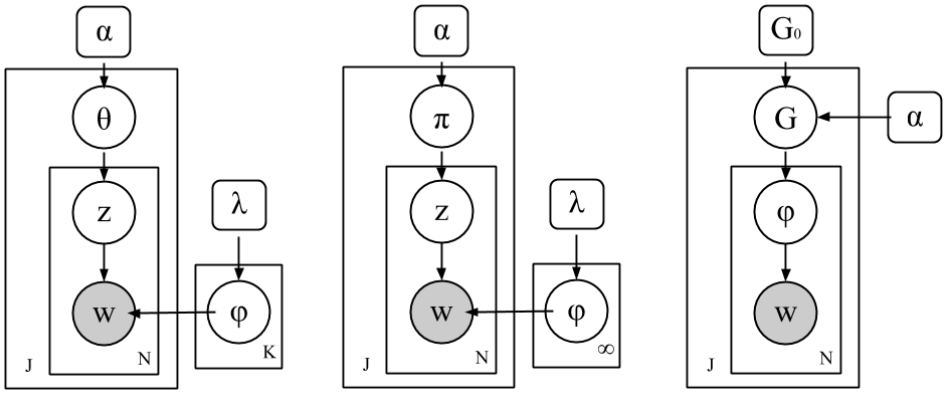

Figure 2.1: LDA model as a finite mixture model (left), Infinite mixture model using the stick- breaking representation (center) and Dirichlet Process mixture model (right)

the occupied tables with probability proportional to the number of people currently at that table. Mathematically, this is

$$P(i^{th} \text{ customer at table } k) = \begin{cases} \frac{n_k}{\alpha+i-1} & \text{if table } k \text{ is occupied} \\ \frac{\alpha}{\alpha+i-1} & \text{if } k \text{ is a new table} \end{cases} \tag{2.7}$$

where n_k is the number of customers currently seated at table k. In a topic model, the customers correspond to words w, the tables to topic clusters z_w and the dishes to the topic distributions ϕ_k.

2.3 Hierarchical Dirichlet Process

By using a Dirichlet Process, we are able to cluster the words into topic groups which span over all the documents in the corpus. However, by additionally grouping words into documents, we want to ensure that the topics from each document are shared across the entire corpus, while simultaneously maintaining the global

popularity of a given topic. For example, it is important that the topics from a science article about brain growth in children also overlap with the topics from a document about basic anatomy.

This new structure is captured in adding another level to our generative process, giving a Hierarchical Dirichlet Process (HDP) (Teh, Jordan, Beal, & Blei, 2006). By drawing the base measures for each group of documents in turn from a global Dirichlet Process, we enforce the shared support of each lower level DP and overall relative likelihood of each topic. The base measure G_0 is in turn distributed with concentration parameter γ and base measure H. Each ϕ_k is now distributed according to G_j, for a group j. This updates (2.6) to

$$\begin{aligned} G_0 &\sim DP(\gamma, H), \\ G_j|G_0 &\sim DP(\alpha, G_0), \\ \phi_k &\sim G_j, \\ w|z_w, \boldsymbol{\phi} &\sim F(\phi_{z_w}) \end{aligned} \tag{2.8}$$

This is more easily understood in the context of the Chinese Restaurant Process with an extension that is aptly named the Chinese Restaurant Franchise (CRF). Similar to the single level model, the customers correspond to words and the dishes to the topic distributions. The restaurants (documents) have groups of words at tables, and each table will have a dish. All restaurants are part of the same franchise, and thus all share the same menu of dishes, with some dishes being more popular than others. The hierarchy can be understood visually in Figure 2.2, with the table-dishes level shown on top and the customer-table level shown beneath (Teh, 2007).

Now we also keep track of the dishes each table chooses, clustering those tables together, with the dish chosen for a table chosen in the same manner by which a customer chooses to sit at a table. Mathematically,

$$P(i^{th} \text{ customer at table } t \text{ in restaurant } j) = \begin{cases} \frac{n_{jt\cdot}}{\alpha+i-1} & \text{if table } t \text{ is occupied} \\ \frac{\alpha}{\alpha+i-1} & \text{if } t \text{ is a new table} \end{cases}$$
$$P(t^{th} \text{ table choosing dish } k) = \begin{cases} \frac{m_{\cdot k}}{\gamma+m_{\cdot\cdot}} & \text{if dish } k \text{ has already been chosen} \\ \frac{\gamma}{\gamma+m_{\cdot\cdot}} & \text{if } k \text{ is a new dish} \end{cases} \tag{2.9}$$

where we change notation slightly to n_{jtk} representing the number of customers in restaurant j at table t eating dish k, and m_{jk} representing the number of tables in restaurant j serving dish k. Marginal counts are given by dots, with $m_{\cdot k}$ standing for the total number of tables serving dish k.

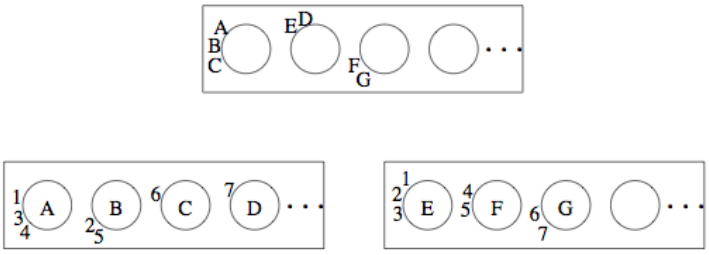

Figure 2.2: The hierarchy of the Chinese Restaurant Franchise. Top level, tables (represented by letters) are clustered together, each around a dish. Second level, customers (represented by numbers) are clustered at tables. Restaurants are split across the second level.

2.4 Adding Document Clustering

The document level clustering is added on top of the HDP model, where the restaurants now correspond to a cluster of documents. This addition follows a similar structure as the topic clustering described above with each document cluster

$c_j \sim D \sim GEM(\tau)$. By adding this level into the HDP, the resulting clustered HDP (cHDP) system becomes

$$\begin{aligned} G_0 &\sim DP(\gamma, H), \\ c_j \sim D &\sim GEM(\tau), \\ G_{c_j}|G_0 &\sim DP(\alpha, G_0), \\ \phi_k &\sim G_{c_j}, \\ w_{j,n}|z_{j,n}, \boldsymbol{\phi} &\sim F(\phi_{z_{j,n}}) \end{aligned} \tag{2.10}$$

Because the clustering occurs at the document level rather than at the word level, we again expand our notation such that $w_{j,n}$ is the nth word in document j, and $z_{j,n}$ is that word's topic. Furthermore, it is important to note that under the HDP parameterization in the previous section, the sampling updates for the tables and topics are coupled together. This poses issues for this new document clustering. In the HDP analogy, we have all the words at tables within a single document. With the document clusters, we would like the words to freely associate within the other words in their document cluster. This requires a change of parameterization, which will be discussed in section 3.2.3. Overall, however, we can note that the document cluster updates occur as

$$p(c_j = c*) \propto \begin{cases} \rho_{c*}^{(-j)} p(\mathbf{z}|\mathbf{c}) & \text{if } c* \text{ is an existing cluster} \\ \tau p(\mathbf{z}|\mathbf{c}) & \text{if } c* \text{ is a new cluster} \end{cases} \tag{2.11}$$

where ρ_{c*} is the count of documents in cluster $c*$

Section 3 Inference

In modeling the topics of a corpus, we are specifically looking to find the posterior probability of a topic given the text and hyperparameters. While the exact manner of computation varies according to the model specification, the overall approach is the same for each of the models specified in this work.

3.1 Latent Dirichlet Allocation

For the first model using LDA, the posterior probability is $p(\theta, \phi, \mathbf{z}|\mathbf{w}, \alpha, \lambda)$, which is given by the computationally intractable formula

$$p(\theta, \phi, \mathbf{z}|\mathbf{w}, \alpha, \lambda) \propto \frac{p(\theta, \phi, \mathbf{z}, \mathbf{w}|\alpha, \lambda)}{p(\mathbf{w}|\alpha, \lambda)} \tag{3.1}$$

Instead, we use Gibbs sampling. Of important note is the fact that the conditionals for $\boldsymbol{\theta_d}$ and for $\boldsymbol{\phi_{z_w}}$ are both solely depending on z_w, allowing the use of a collapsed Gibbs sampler by integrating out these latent variables. This leaves simply finding $p(z_w|\mathbf{z}_{-w}, \alpha, \lambda, \mathbf{w})$, where $\mathbf{z}_{-w}$ indicates topic allocations not including z_w.

For the derivation, we will switch notation from z_w indicating the topic assignment for word w, to instead z_i representing the ith word, and $\mathbf{z}^{(-i)}$ representing the topic assignments not including the ith item. Additionally, let n_{jkw} stand for the number of times word w is in topic k in document j.

$$p(z_i|\mathbf{z}^{(-i)}, \mathbf{w}) = \frac{p(\mathbf{z})}{p(\mathbf{z}^{(-i)})} \frac{p(\mathbf{w}|\mathbf{z})}{p(\mathbf{w}^{(-i)}|\mathbf{z}^{(-i)})p(w_i)} \tag{3.2}$$

$$p(z_i|\mathbf{z}^{(-i)}, \mathbf{w}) \propto \prod_m \frac{B(n_{j..} + \alpha)}{B(n_{j..}^{(-i)} + \alpha)} \prod_k \frac{B(n_{.k.} + \lambda)}{B(n_{.k.}^{(-i)} + \lambda)} \tag{3.3}$$

$$p(z_i|\mathbf{z}^{(-i)}, \mathbf{w}) \propto \frac{\Gamma(n_{jk.} + \alpha)\Gamma(\sum_{k=1}^{K} n_{jk.}^{(-i)} + \alpha)}{\Gamma(n_{jk.}^{(-i)} + \alpha)\Gamma(\sum_{k=1}^{K} n_{jk.} + \alpha)} \frac{\Gamma(n_{.k,w} + \lambda)\Gamma(\sum_{w=1}^{W} n_{.kw}^{(-i)} + \lambda)}{\Gamma(n_{.kw}^{(-i)} + \lambda)\Gamma(\sum_{w=1}^{W} n_{.kw} + \lambda)} \tag{3.4}$$

$$p(z_i|\mathbf{z}^{(-i)}, \mathbf{w}) \propto (n_{jk.}^{(-i)} + \alpha) \frac{n_{.kw}^{(-i)} + \lambda}{\sum_{u \in W} n_{.ku}^{(-i)} + \lambda} \tag{3.5}$$

An important point to remember here is the exchangeability of words in the corpus, which, as mentioned previously, allows for substitution of words without regard to order. This in turn allows for easily discounting the impact of the current z_i simply by decrementing the counts for that topic, instead of having to decouple that assignment from the joint distribution. For a more complete derivation, please see Darling (2011).

3.2 Chinese Restaurant Schemes

3.2.1 Chinese Restaurant Process

For the infinite dimensional case, we are able to update the prior for $\mathbf{z}$ based on equation (2.7) where we assume that the current word i is the most recent customer to arrive. The likelihood for the word under its topic remains the same, if the topic already exists. For a new topic, we instead use a conjugate prior of a symmetric

Dirichlet likelihood, $\frac{1}{W}$, since there are no counts available. Note this assumes that all words have equal probability, as we declared with our bag-of-words assumption. This gives

$$p(z_i = k|\mathbf{z}^{(-i)}, \mathbf{w}) \propto \begin{cases} n_{.k}^{(-i)} \frac{n_{.kw}^{(-i)}+\lambda}{\sum_{u\in W} n_{.ku}^{(-i)}+\lambda} & \text{if } k \text{ is an existing topic} \\ \alpha \frac{1}{W} & \text{if } k \text{ is a new topic} \end{cases} \tag{3.6}$$

3.2.2 Chinese Restaurant Franchise

As one might intuit, the sampling updates for the CRF are very similar to the CRP, with each level update still holding the form of prior likelihood of the cluster time the likelihood of the item in that cluster. At level one, we sample the table assignments for each word.

$$p(t_{ji} = t|\mathbf{t}^{(-ji)}, \mathbf{z}, \mathbf{w}) \propto \begin{cases} n_{jt.}^{(-i)} \frac{n_{jt.,w}^{(-i)}+\lambda}{\sum_u n_{.k_t,u}^{(-i)}+\lambda} & \text{if } t \text{ is an existing table} \\ \frac{\alpha}{m_{..}+\gamma} \prod_k (m_{.k} \frac{n_{..k,w}^{(-i)}+\lambda}{\sum_{u\in W} n_{..k,u}^{(-i)}+\lambda} + \frac{\gamma}{W}) & \text{if } t \text{ is a new table} \end{cases} \tag{3.7}$$

We have the likelihood of assigning word ji to table t as proportional to the product of the size of the table and the likelihood of the word under the topic for that table. For a new table, we must evaluate the likelihood of that word under all potential topics for that new table, including a new topic altogether. If a new table is chosen then we sample the topic according to

$$p(z_{ji} = k|\mathbf{z}^{(-ji)}, t, \mathbf{w}) \propto \begin{cases} m_{.k}^{(-i)} \frac{n_{..k,w}^{(-i)}+\lambda}{\sum_t n_{..k,t}^{(-i)}+\lambda} & \text{if } k \text{ is an existing topic} \\ \gamma \frac{1}{W} & \text{if } k \text{ is a new topic} \end{cases} \tag{3.8}$$

as with the CRP.

We then move on to the second level, to sample the topic assignments for each table.

$$p(z_{jt} = k|\mathbf{z}^{(-i)}, \mathbf{t}, \mathbf{w}) \propto \begin{cases} m_{.k}^{(-jt)} \prod_{v \in w_{ji} \forall t_{ji}=t} \frac{n_{..k,v}^{(-t)}+\lambda}{\sum_u n_{..k,u}^{(-t)}+\lambda} & \text{if } k \text{ is an existing topic} \\ \gamma \prod_{v \in w_{ji} \forall t_{ji}=t} \frac{n_v^*+\lambda}{\sum_u n_u^*+\lambda} & \text{if } k \text{ is a new topic} \end{cases} \tag{3.9}$$

For an existing topic, again we have the size of the topic as the prior likelihood of that topic multiplied by the likelihood of all that table's words under that topic. For the new topic, a similar approach is used, adding on the joint likelihood of all the table words under the new topic. n_v^* represents the count of word v with the new topic. It is denoted separately here as the counts solely use the information for the current table of words, in contrast with the other counts in the updates, which explicitly ignore the information provided by the cluster being updated.

3.2.3 Using an augmented representation

As discussed in section 2.4, in order to cluster across documents, we need to decouple the updates. To do this, we sample from G_0 such that the conditional posterior factorizes across groups.

As shown in Teh et al. (2006) if we condition on the table topics, $G_0|z_{jt} \sim DP(\gamma + m_{..}, \frac{\gamma H + \sum_{k=1}^{K} m_{.k}\delta_{\phi_k}}{\gamma + m_{..}})$ This gives

$$\begin{aligned}
\boldsymbol{\beta} = (\beta_1, ..., \beta_K, \beta_u) &\sim Dir(m_{\cdot 1}, ..., m_{\cdot K}, \gamma) \\
p(\phi_k | \mathbf{t}, \mathbf{k}) &\propto h(\phi_k) \prod_{ji: k_{jt_{ji}} = k} f(x_{ji} | \phi_k) \\
G_u &\sim DP(\gamma, H) \\
G_0 &\sum_{k=1}^{K} \beta_k \delta_{\phi_k} + \beta_u G_u
\end{aligned} \tag{3.10}$$

If we directly assign items to their mixture components, we can simplify the updates. Using just z_{ji} instead of t_{ji} and k_{ji} is sufficient alongside $\boldsymbol{\beta}$ and $\mathbf{m}$, as shown in Fox, Sudderth, Jordan, and Willsky (2007).

$$p(z_{ji} = k | \mathbf{z}^{(-ji)}, \boldsymbol{\beta}) \propto \begin{cases} (n_{j.k}^{(-ji)} + \alpha\beta_k) \frac{n_{..k,w}^{(-i)} + \lambda}{\sum_t n_{..k,t}^{(-i)} + \lambda} & \text{if } k \text{ is an existing topic,} \\ \alpha\beta_u \frac{1}{W} & \text{if } k \text{ is a new topic} \end{cases} \tag{3.11}$$

$$\boldsymbol{\beta} | \mathbf{t}, \mathbf{k} \sim Dir(m_{\cdot 1}, ..., m_{\cdot K}, \gamma) \tag{3.12}$$

The updates for each of the m_{jk} are given by

$$p(m_{jk} = m | n_{j.k}, \boldsymbol{\beta}, \alpha) = \frac{\Gamma(\alpha\beta_k)}{\Gamma(\alpha\beta_k + n_{j.k})} s(n_{j.k}, m) (\alpha\beta_k)^m \tag{3.13}$$

where $s(n, m)$ are unsigned Stirling numbers of the first kind.

However, instead of computing $\mathrm{p}(m_{jk} = m)$ for all m and j, we can instead simply partition the words in restaurant j with topic k using a Chinese Restaurant Process with concentration $\alpha\beta_k$, and then taking $m_{\cdot k}$ as the number of partitions. This follows the example of Fox et al. (2007), and is more efficient than computing the array of Stirling numbers when the $n_{j.k}$ are large.

3.3 Adding Document Clustering

We now use this augmented representation to handle the updates for the clustered HDP.

$$p(z_{ji} = k|\mathbf{z}^{(-ci)}, \boldsymbol{\beta}) \propto \begin{cases} (n_{c.k}^{(-ci)} + \alpha\beta_k)\frac{n_{..k,w}^{(-i)}+\lambda}{\sum_t n_{..k,t}^{(-i)}+\lambda} & \text{if } k \text{ is an existing topic,} \\ \alpha\beta_u \frac{1}{W} & \text{if } k \text{ is a new topic} \end{cases} \tag{3.14}$$

where $n_{c.k}^{(-ci)}$ is the number of times topic z_{ji} appears in the cluster c.

The $\mathbf{m}$ are now the set of all m_{ck}, but remain updated with $m_{.k}$ as the number of partitions of words in neighborhood c with topic k, and the $\boldsymbol{\beta}$ updates as with Equation (3.12).

For the updates to the document clusters, we have a multiplicative term for the joint topic likelihoods

$$x_{c_j} = \begin{cases} \prod_k \frac{\Gamma(n_{c*.k}^{(-c*)}+n_{.jk}^{(-j)}+\alpha\beta_k}{\Gamma(n_{c*.k}^{(-c*)}+\alpha\beta_k)} & \text{if } c* \text{ is an existing cluster,} \\ \prod_k \frac{\Gamma(n_{c*jk}+\alpha\beta_k)}{\Gamma(\alpha\beta_k)} & \text{if } c* \text{ is a new cluster} \end{cases} \tag{3.15}$$

Which results in

$$p(c_j = c* |\mathbf{c}^{(-j)}, \boldsymbol{\beta}, \boldsymbol{\rho}) \propto \begin{cases} \rho_{c*}^{(-j)} x_{c_j} \frac{\Gamma(\sum_k n_{c*.k}^{(-c*)}+\alpha\beta_k)}{\Gamma(\sum_k n_{c*.k}^{(-c_j)}+n_{.jk}^{(-j)}+\alpha\beta_k)} & \text{if } c* \text{ is an existing cluster,} \\ \tau x_{c_j} \frac{\Gamma \sum_k \alpha\beta_k}{\Gamma(\sum_k n_{.jk}^{(-j)}+\alpha\beta_k)} & \text{if } c* \text{ is a new cluster} \end{cases} \tag{3.16}$$

Section 4 Results

4.1 Model Comparison

A collapsed sampler was implemented for LDA, the augmented HDP, and the clustered HDP and run for 5000 iterations on a corpus of speeches from Macbeth. For LDA, K was set at 50, a value that was partially informed by the resulting number of topics from the other methods, λ was set to 0.1, and α was set to 1 following the rule-of-thumb of $\alpha = 50/K$. This followed the example of Griffiths and Steyvers (2004). The HDP and cHDP used $\alpha = 5$, $\gamma = 20$, with starting $K = 50$ The cHDP added $\tau = 10$ and starting $C = 15$. These values were informed by pilot runs of the two algorithms and were chosen to provide a reasonable spread over the topics and clusters.

Table 4.1 shows the top ten most common words in five example topics from each of the runs. While all the topic models show similar results in theme, those familiar with The Scottish Play will note that the HDP and cHDP produce more focused topics. For example, Topic 3 in the HDP contains many words from Act 4, Scene 1, with the witches preparing their cauldron, and Topic 4 from cHDP from Act 2 Scene 2, where Macbeth and Lady Macbeth have just murdered Banquo. Indeed, looking at the log word perplexity for each of the runs (calculated using the harmonic mean), we see that the hierarchical methods require much less information to code each word, with LDA's log perplexity of 7.537 compared to HDP's 1.445 and cHDP's 1.383.

4.2 Combined Corpus

The cHDP was run on a corpus of acts from all of the tragedies for 1000 iterations. A shorter runtime was necessary due to the enlarged corpus, as only 2000 iterations were able to complete within a 24 hour window on TACC's Lonestar computer, a Linux cluster with a peak performance of 302 TFLOPS. However, the algorithm does converge quickly enough to permit this drop in iterations, as evidenced by Figure 4.1, which shows the trend of the proportional log likelihood over time.

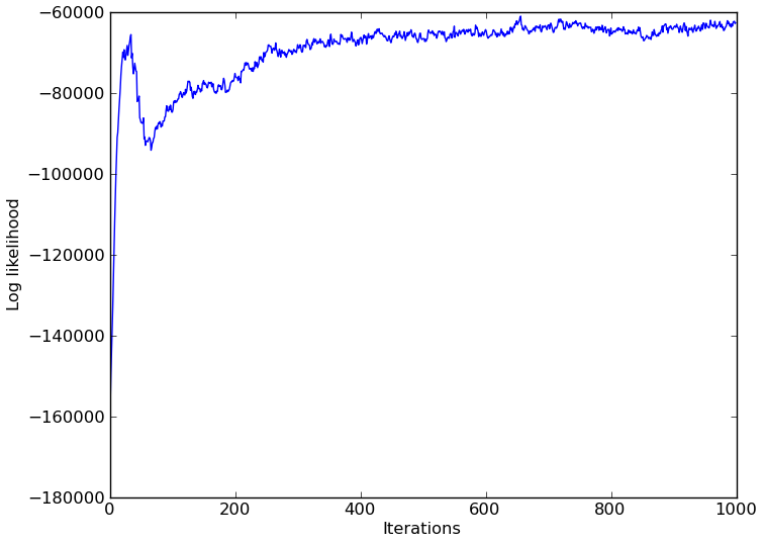

Figure 4.1: Convergence of the log likelihood for a corpus of all tragedy acts

The larger corpus additionally required a change in hyperparameters: $\alpha = 10$, $\gamma = 25$, and $\tau = 15$. The initial values for K and C were kept at 50 and 15, respectively. Selected topics are shown in Table 4.2 with a corresponding histogram shown in Figure 4.2. Note that the histogram also shows a Topic 0, which is not highlighted in the sampling in Table 4.2 due to its low prevalence compared with the

other topics. cHDP very acutely identified topics corresponding to words in each play, as evidenced by the spike in topic proportions for Topic 3 for Julius Caesar, Topic 7 for Coriolanus, and so on. However, the algorithm was also able to pull out words commonly found across all plays, as shown in Topics 2 and 5, with themes around honor and power as with Topic 2 and grief and murder for Topic 5.

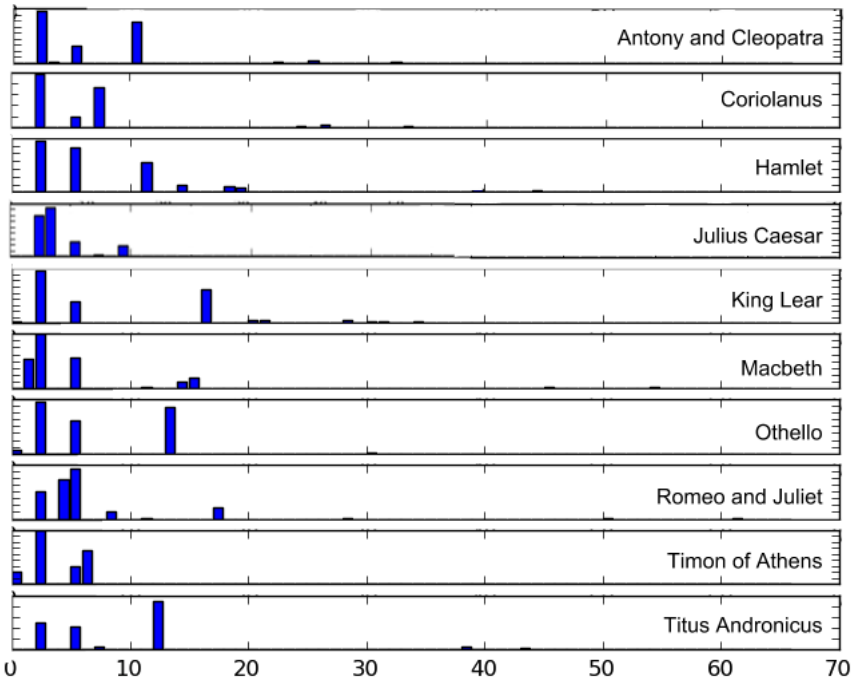

Figure 4.2: Histogram for words in each topic by play. Note the individual spikes for topics containing the main characters for that play, as well as the common topics across plays, specifically topics two and five.

With regards to the document clustering, cHDP did well in grouping acts from the same play. However, the algorithm did identify 26 clusters, compared to the 10 plays used in the corpus, thus resulting in a few single document clusters. The results

of the clustering can be shown in Figure 4.3. The model correctly clustered together most of the acts in the historically centered tragedies: Antony and Cleopatra, Julius Caesar, Timon of Athens, and Titus Andronicus. Romeo and Juliet acts were also mostly kept together. King Lear, Macbeth, and Othello provided the most trouble for the clustering, as no acts were linked within those plays. This is likely due to the cast and scene changes between acts, which are more prominent in those plays than in, for example, Julius Caesar, which does not have major changes in theme, as it is very focused on the politics of Rome. This clustering was quantitatively evaluated using the F_1 score, which can be thought of as a weighted average of precision and recall, and is given by

$$F_1 = 2\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{4.1}$$

or

$$F_1 = \frac{2 * \text{ true positives}}{2 * \text{ true positives} + \text{false positives} + \text{false negatives}} \tag{4.2}$$

(Van Rijsbergen, 1979). The act corpus clustering had an F_1 score of .921 / 1.

4.3 Individual Plays

The cHDP was also run on speeches from each of the plays individually, with the same hyperparameters: 5000 iterations $\lambda = 0.1$, $\alpha = 5$, $\gamma = 20$, $\tau = 10$ with starting $K = 50$ and starting $C = 15$. The plays were then ranked according to their number of topics, log perplexity, and F_1 scores for clustering speeches in scenes. Combined metrics are all given in Table 4.4 while the rankings are given in Table 4.3. The only significant correlations between complexity metrics were between the length-adjusted topic count and the length of the play (See Table 4.5).

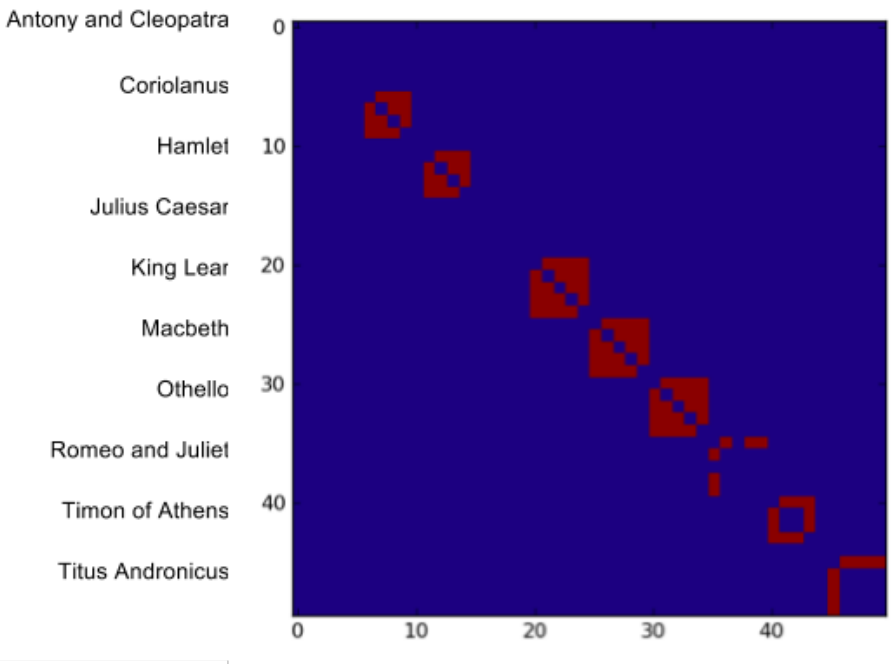

Figure 4.3: Clustering results from the act corpus. Blue represents a correct match or correct mismatch, and red shows a missed link.

LDA				
Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Scotland	power	receive	dagger	wife
cruel	present	grave	thane	office
Banquo	ten	sister	dearest	outward
honour	ember	women	thou'rt	dark
consequence	cauldron	fight	hang	desire
honest	newest	thank	thrice	knife
attempt	point	liege	charge	bold
bloody	wife	babe	live	danger
smile	remembrance	feast	horror	fought
thane	close	palace	t'were	fife

HDP				
Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
arm	courage	thick	dark	thane
sword	power	bellman	confess	Cawdor
Scotland	confound	expect	die	'tis
king	way	scale	night	seem
fortune	patience	silver	for't	honour
kern	act	tooth	moment	noble
cling	abound	chaudron	court	wouldst
supply	concord	dragon	thunder	Glamis
valour	crime	eclipse	bloody	crown
began	devote	goat	Scotland	dagger

cHDP				
Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
hail	Duncan	harm	dagger	king
soldier	tyrant	sleep	charge	'tis
kingdom	approach	bleed	face	Macbeth
praise	face	brave	smile	Banquo
'I	minion	except	horrible	thane
fie	belief	sooth	sleepy	murder
broil	true	stand	hark	honour
captive	wrong	root	carry	noble
didst	summer	given	anon	live
ought	nature	throne	gentleman	country

Table 4.1: Selected topics from Macbeth

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
Macbeth	'tis	Caesar	Romeo	'tis	Timon
Macduff	noble	Brutus	Tybalt	wit	Athens
tyrant	honour	Antony	Juliet	bed	servant
Scotland	fortune	Cassius	nurse	weep	Lucilius
king	god	Rome	banish	grief	Apemantus
Birnam	follow	Roman	Paris	madam	lordship
wood	'I	Casca	friar	true	honour
Dunsinan	son	honour	Mercutio	daughter	rich
cauldron	power	to-day	joy	murder	money
double	sword	capitol	prince	play	senate

Topic 7	Topic 10	Topic 11	Topic 12	Topic 13	Topic 16
people	Caesar	Hamlet	Rome	Cassio	king
Rome	Antony	king	son	moor	sister
Marcius	madam	Laertes	Titus	Iago	daughter
voice	queen	Horatio	Lavinia	Desdemona	Edmund
consul	Egypt	Denmark	brother	lieutenent	duke
home	Cleopatra	Ophelia	emperor	Othello	Gloucester
tribune	Pompey	England	Andronicus	Cyprus	letter
Coriolanus	war	queen	Marcus	Roderigo	Tom
noble	Rome	Gertrude	tribune	willow	France
country	eros	fine	Bassianus	Venice	Cordelia

Table 4.2: Selected topics from the combined act corpus

K	log perplexity	F_1
Coriolanus	Hamlet	Timon of Athens
King Lear	Titus Andronicus	Hamlet
Timon of Athens	Julius Caesar	Titus Andronicus
Julius Caesar	Othello	Othello
Hamlet	Coriolanus	Julius Caesar
Othello	Macbeth	Antony and Cleopatra
Romeo and Juliet	Romeo and Juliet	King Lear
Antony and Cleopatra	King Lear	Coriolanus
Macbeth	Antony and Cleopatra	Romeo and Juliet
Titus Andronicus	Timon of Athens	Macbeth

Table 4.3: Plays ranked by complexity metrics. K and log perplexity are ranked with highest scores given first, while F_1 is given lowest score first in order to keep more complex plays ranked highest.

Play	K	K/M	K/W	W	M	log perplexity	F_1
Antony and Cleopatra	75	0.06	0.03	2684	1232	1.269	0.886
Coriolanus	87	0.08	0.03	2768	1154	1.566	0.906
Hamlet	78	0.06	0.02	3353	1203	1.842	0.855
Julius Caesar	81	0.10	0.04	1921	836	1.81	0.876
King Lear	84	0.07	0.03	2906	1137	1.351	0.899
Macbeth	74	0.11	0.03	2347	695	1.383	0.908
Othello	78	0.06	0.03	2642	1234	1.616	0.866
Romeo and Juliet	76	0.09	0.03	2627	879	1.37	0.906
Timon of Athens	82	0.10	0.04	2230	800	1.196	0.825
Titus Andronicus	70	0.12	0.03	2349	606	1.823	0.859

Table 4.4: Metrics of play complexity for each of the ten tragedies

	K	K/M	K/V	V	M	log perplexity
K/M	-0.34					
K/V	0.25	0.48				
V	0.14	-0.73*	-0.88**			
M	0.45	-0.98**	-0.44	0.71		
log perplexity	-0.17	0.08	-0.29	0.07	-0.05	
F_1	0.09	-0.07	-0.23	0.13	0.13	-0.17

Table 4.5: Correlations between complexity metrics. * $p < 0.05$, ** $p < 0.01$

Section 5 Discussion

This work sought to investigate the effectiveness of document clustering as an extension onto the existing Hierarchical Dirichlet Process model framework. Effectiveness was measured by a qualitative assessment of the topics produced on runs for speeches within a single play (Macbeth) and for acts within a corpus of ten tragedies. For topics within Macbeth, the HDP and cHDP performed similarly, with both methods showing improved, more scene-centered clusters than those produced by LDA. This is due to the assignment of word tables to topics, which finds like words before assigning them to a topic. There did not appear to be a discernible qualitative difference in topics produced from the HDP and cHDP, which is to be expected, as the cHDP acts as an addition to HDP's model rather than as an adjustment. Furthermore, the HDP and cHDP saw similar measures for log perplexity per word, with cHDP showing slight improvement.

Looking at the other tragedies, the cHDP did not show any impact of the length of the corpus or on the number of words on the resulting number of topics, the perplexity per word, or the F_1 fit of the document clusters. That is not to say that the model does equally well for any length of corpus, as when using a larger corpus, the algorithm does perform slowly, with the time required to complete inference scaling with both the size of the corpus and with the number of topics found (which also scales with the size of the corpus). However, a quick convergence allows for a more practically feasible run time if samples are not needed to be drawn from the

results, which is the case for most practical applications of topic models. If desired, a possible improvement in computation time could be found by using variational inference instead of a Gibbs sampling approach, as variational inference has been found to converge in fewer iterations, even though there is a chance of convergence to the wrong statistics (Jordan et al., 1998).

For the full corpus, the topics found very clearly align with specific plays, with the exception of topics that span the entire corpus. Furthermore, these general topics occur in different proportions as would be expected for each of the plays. For examples, Romeo and Juliet has a higher proportion of the 'grief' and 'murder' topic than the 'honour' and 'power' topic, while Hamlet shows high proportions of both, and Julius Caesar shows the reverse pattern. In terms of document clustering, the cHDP obtained an F_1 score of 92.1% for the full corpus, with an average of 88.1% for the individual plays. Because the HDP (and, by extension, the cHDP) does have a tendency to overfit (Miller & Harrison, 2014), with a large number of topics containing fewer than ten words, the clustering tends to also produce a larger than necessary number of clusters, even though the clusters themselves accurately group their items.

With this ability to group documents, a next step for this work could be to apply the algorithm to a social network, allowing for clustering of speakers or listeners. Even beyond that, clustering could be performed on the pairs of speakers and listeners for an asymmetric modeling of speaker topics, taking advantage of this easily adaptable model for text. Alternatively, the cHDP could be very useful in terms of information retrieval, providing a more adaptive alternative to the commonly used K-means approach to document clustering (Steinbach, Karypis, & Kumar, 2000). Not

only would we again be able to use a data-driven value for the number of clusters, but this type of algorithm avoids a hard choice between a divisive or agglomerative approach, as topics can be added or combined at any step.

Appendix

the, of, and, to, i, you, me, that, it, is, a, my, this, be, in, your, thy, not, his, but, with, he, as, have, do, no, so, by, what, thee, for, all, shall, an, our, now, was, thou, there, o, from, men, let, will, him, would, at, on, how, here, we, are, tis, they, these, which, if, more, or, than, them, one, then, like, good, am, go, hath, say, make, were, too, upon, yet, out, mine, did, their, know, well, should, when, may, take, she, up, eyes, down, such, time, much, tell, own, who, man, see, where, nor, had, sleep, grace, none, life, little, hear, art, why, look, her, love, long, get, again, must, bear, both, set, sweet, us, myself, some, since, till, whose, even, those, never, seen, done, still, fear, ever, show, himself, eye, way, made, words, against, every, pity, heart, thus, old, thousand, dear, each, comes, whom, lies, think, fair, being, leave, else, hands, most, put, been, that's, best, forth, into, sun, thought, any, poor, blood, world, though, full, mind, without, ere, off, stay, pardon, ne'er, once, light, said, could, what's, reason, hand, peace, find, true, part, might, tongue, end, new, things, kind, truth, turn, found, head, thine, last, face, place, gentle, heard, many, false, hard, brought, break, under, stand, age, saw, hour, thing, hence, soul, sent, right, earth, nature, high, came, doth, wear, serve, fall, therefore, master, past, hast, company, lady, makes, next, scene, heaven, another, away, night, let's, death, meet, cause, lost, lose, yours, yes, dead, exeunt, father, sake, first, enough, needs, coming, through, cut, gave, goes, farewell, sure, spirit, methinks, give, act, gone, ay, canst, sight, doubt, hope, near, voice, live, remember, rather, indeed, friend, very, foot, faith, keep, neither, matter, stop, iv, shame, late, enter, woman, use, looks, other, virtue, rest, gold, having, young, back, bring, means, believe, after, welcome, can, only, cannot, between, help, soon, within, strange, dost, friends, talk, two, common,

purpose, iii, about, trust, mother, name, seems, known, v, call, pray, nothing, man's, sir, sit, lord, second, hold, alone, tears, before, exit, mean, unto, great, we'll, told, word, lay, ill, speak, yourself, i'll, while, nay, state, answer, better, content, three, thoughts, left, die, hither, mark, fellow, house, please, bid, day, over, come, kill, far, mad, cold, wilt, hours, ask, attend, wherefore, alas, begin, glad, born, need, often, want, call'd, try, seek, case, cry, mercy, god, drink, is't, grow, something, wonder, proud, worth, pass, walk, perceive, knows, suffer, ear, fly, almost, report, wrong, news, run, bosom, ground, woman's, twere, shows, beauty, forgot, holy, longer, sport, mouth, truly, itself, yield, haste, lead, attendants, hell, fell, shouldst, devil, sound, sad, course, note, less, sorrow, bless, sea, bears, send, bad, above, stir, throw, o'er, thank, water, read, fool, either, gives, lest, ho, kept, whither, merry, fit, strike, twas, golden, vile, comfort, sing, flesh, ii, effect, married, legs, wherein, villain, care, behold, pleasure, promise, soft, swear, fast, half, days, presently, teeth, takes, loving, wish, worse, entreat, doing, low, oath, whence, shape, together, gracious, wisdom, worst, same, kill'd

Bibliography

Aldous, D. J. (1985). Exchangeability and related topics. In *Ecole d'Ete de Probabilites de Saint-Flour XIII, 1983* (Vol. 1117). Lecture Notes in Mathematics. Springer-Verlag Berlin Heidelberg.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *The Journal of Machine Learning Research*, *3*, 993–1022.

Chang, J. & Blei, D. M. (2009). Relational topic models for document networks. In *International Conference on Artificial Intelligence and Statistics* (pp. 81–88).

Darling, W. M. (2011). A theoretical and practical implementation tutorial on topic modeling and Gibbs sampling. University of Guelph. Retrieved from http://u.cs.biu.ac.il/~89-680/darling-lda.pdf

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 209–230.

Fox, E. B., Sudderth, E. B., Jordan, M. I., & Willsky, A. (2007). *Developing a tempered HDP-HMM for systems with state persistence* (Technical Report No. P-2777). MIT Laboratory for Information and Decision Systems.

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-6*(6), 721–741. doi:10.1109/TPAMI.1984.4767596

Griffiths, T. L. & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, *101*(Supplement 1), 5228–5235. doi:10.1073/pnas.0307752101

Jivani, A. G. (2011). A comparative study of stemming algorithms. *International Journal of Computer Technology and Applications*, *2*(6), 1930–1938.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1998). An introduction to variational methods for graphical models. In *Learning in Graphical Models* (pp. 105–161). Dordrecht: Springer Netherlands. Retrieved April 18, 2015, from http://link.springer.com/10.1007/978-94-011-5014-9_5

Millar, J., Peterson, G., & Mendenhall, M. (2009). Document clustering and visualization with Latent Dirichlet Allocation and self-organizing maps. *FLAIRS Conference*, *21*, 69–74.

Miller, J. W. & Harrison, M. T. (2014). Inconsistency of Pitman-Yor process mixtures for the number of components. *The Journal of Machine Learning Research*, *15*(1), 3333–3370.

Porter, M. F. (2001). Snowball: A language for stemming algorithms. Retrieved from http://www.snowball.tartarus.org/texts/introduction.html

Salton, G. & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York: McGraw-Hill Book Co.

Sethuraman, J. (1991). *A constructive definition of Dirichlet priors*. DTIC Document.

Steinbach, M., Karypis, G., & Kumar, V. (2000). *A comparison of document clustering techniques* (Technical Report No. TR 00-034). University of Minnesota.

Steyvers, M., Smyth, P., & Griffiths, T. L. (2004). Probabilistic author-topic models for information discovery. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press.

Teh, Y. W. (2007). A tutorial on Dirichlet Processes and Hierarchical Dirichlet Processes. Retrieved March 29, 2015, from http://mlg.eng.cam.ac.uk/tutorials/07/ywt.pdf

Teh, Y. W., Jordan, M. I., Beal, M. J., & Blei, D. M. (2006). Hierarchical dirichlet processes. *Journal of the American Statistical Association*, *101*(746).

Van Rijsbergen, C. J. (1979). *Information retrieval* (2d ed). Boston: Butterworths.

Xu, J. & Croft, B. (1999). Cluster-based language models for distributed retrieval. In *Proceedings of 22nd International Conference on Research and Development in Information Retrieval*. New York: ACM Press.