Using scores to improve language modelling movie plot summaries

Jorge Sáez Gómez
Roelof van der Heijden
Francesco Stablum

December 22, 2014

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec velit est, fringilla quis mollis in, dapibus nec ipsum. Donec volutpat sapien nec nibh suscipit vehicula. In hac habitasse platea dictumst. Phasellus mattis, enim sit amet tincidunt auctor, ligula ipsum fermentum libero, ut gravida mauris risus ac magna. Vestibulum a tempor mi. Donec viverra feugiat magna, eget lobortis neque volutpat eu. Nullam vehicula vitae nunc in aliquet. Ut vulputate eget eros quis mollis. Curabitur eget egestas est. Vestibulum tincidunt nisl nec justo hendrerit, in ullamcorper mauris porta. Nullam erat tortor, aliquam non purus nec, facilisis sodales risus.

1 Introduction

In recent years, several successes have been booked for applying semantic analysis on user comments of movies. In this report we use those same techniques, but apply them to plot summaries of movies. Using these corpus of text we try to determine whether the contents of these summaries and the score these movies are rated with on the popular online movie database IMDb [**?**] are correlated. We do this by comparing the performance on two latent Dirichlet allocation models - one with and another without using the scores.

This project is part of the Natural Language Processing course of the UvA from Fall 2014.

2 Problem

In this section we describe the characteristics of the problem and take a closer look at the data set that we use.

2.1 Data set

The texts that we use in ths model are summaries of movies. These summaries have been written by users of the popular online movie database IMDb, with the intent to outline the events that occur in the movie.

This is fundamentally different from movie reviews, as the author is not supposed to convey his or her own opinion of the movie in the summary. However, this can obviously never be fully prevented, since the author has seen the movie in question and is willing to spend time and effort to write the summary. In this light we make an important assumption:

An important assumption we make is that the authors wrote the summaries voluntarily, without any compensation or external influence which might affect the writing of the author. Moreover, we assume that the summary also contains the authors personal opinion on the movie, although this does not have to be explicitly mentioned. Only if this assumption holds, can we try to find a correlation between the summary and the score of the movie.

An example plot summary from the movie Big Fish (2003) from IMDb can be found below. It was written by a user who wanted to remain anonymous.

> The story revolves around a dying father and his son, who is trying to learn more about his dad by piecing together the stories he has gathered over the years. The son winds up re-creating his father's elusive life in a series of legends and myths inspired by the few facts he knows. Through these tales, the son begins to understand his father's great feats and his great failings.

3 Approach

3.1 Model

In this section we describe the extended topic based model we used. It is taken from [**?**].

The generative version of LDA is as follows:

1. Draw topic proportions $\theta \mid \alpha \sim \text{Dir}(\alpha)$.
2. For each word:
   (a) Draw topic assignment $z_n \mid \theta \sim \text{Mult}(\theta)$.
   (b) Draw word $w_n \mid z_n, \beta_{1:K} \sim \text{Mult}(\beta_{zn})$.

Its graphical representation can be seen in Figure 3.1.

We however, use the an extended version of LDA, which makes use of the given scores. Because of this, a third step is added to the generative process:

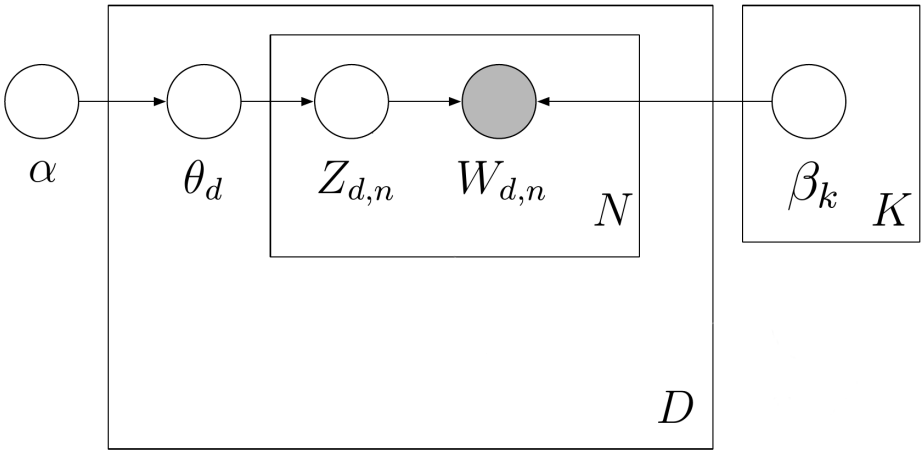

Figure 1: A graphical representation of traditional LDA model.

1. Draw topic proportions $\theta \mid \alpha \sim \text{Dir}(\alpha)$.
2. For each word:
   (a) Draw topic assignment $z_n \mid \theta \sim \text{Mult}(\theta)$.
   (b) Draw word $w_n \mid z_n, \beta_{1:K} \sim \text{Mult}(\beta_{zn})$.
3. Draw response variable $y \mid z_{1:N}, \eta, \sigma^2 \sim \mathcal{N}(\eta^\top \bar{z}, \sigma^2)$.

This version can be called supervised LDA or SLDA. Its graphical representation can be seen in Figure 3.1.

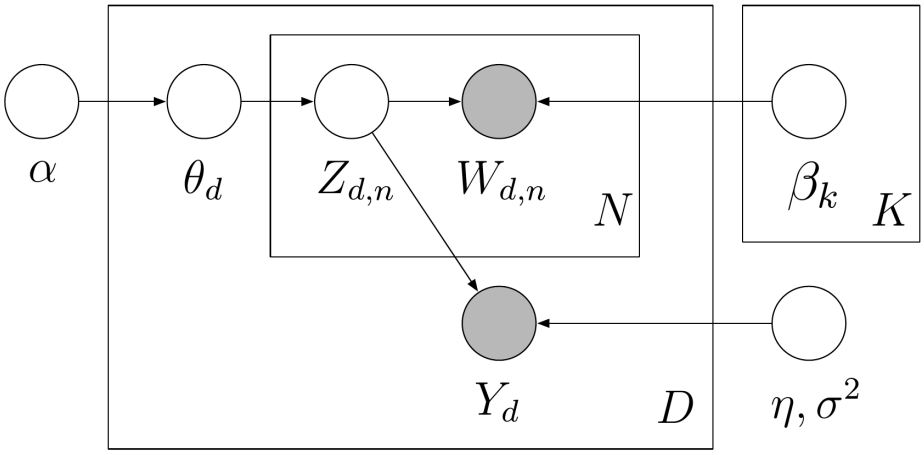

Figure 2: A graphical representation of our modified LDA model.

3.1.1 Collapsed Supervised Latent Dirichlet Allocation

We start by recalling the full likelihood expression (i.e. for all variables, both latent and visible) of the model:

$$p(\Phi, \Theta, S, Z, W \mid \alpha, \beta, \eta, \sigma) = \left[\prod_{k=1}^{K} \mathrm{Dir}(\varphi_k \mid \beta)\right] \times$$

$$\left[\prod_{d=1}^{D} \mathrm{Dir}(\theta_d \mid \alpha)\mathcal{N}(s_d \mid \eta^\top \bar{z}_d, \sigma) \prod_{i=1}^{N_d} \mathrm{Mult}(z_{d,i} \mid \theta_d)\, \mathrm{Mult}(w_{d,i} \mid \varphi_{z_{d,i}})\right] \tag{1}$$

$$\left[\prod_{d} \mathrm{Dir}(\theta_d \mid \alpha) \prod_{d} \mathrm{Mult}(\varphi_d \mid \theta_d)\, \mathrm{Mult}(w_d \mid \varphi_d)\right]$$

$$\int_{\varphi_0} \int_{\varphi_1} \dots \int_{\varphi_k} P(\theta, s, z, \varphi, W \mid \alpha, \beta, \eta, \sigma)$$

$$= \left[\prod_{k} \mathrm{Dir}(\varphi_k \mid \beta)\right] \left[\prod_{d} \mathrm{Dir}(\theta_d \mid \alpha)\mathcal{N}(\eta^\top \bar{z}_d, \sigma) \prod_{i}^{N_d} \mathrm{Mult}(z)\right]$$

$$\times \left[\prod_{d} \prod_{i}^{N_d} \mathrm{Mult}(w_{di} \mid \varphi_{z_d})\right] \to \prod_{d} \prod_{w} \prod_{k} \left[\mathrm{Mult}(w \mid \varphi_k)^{N_{dk}}\right]$$

$$P(\theta, s, z, w \mid \alpha, \beta, \eta, \sigma) = \left[\prod_{k} \int_{\varphi_k} \mathrm{Dir}(\varphi_k \mid \beta) \prod_{d} \prod_{w} \left[\mathrm{Mult}(w \mid \varphi_k)^{N_{dk}}\right]\right]$$

$$= \left[\prod_{d} \mathrm{Dir}(\theta_d \mid \alpha)\mathcal{N}(s_d \mid \eta^\top \bar{z}_d, \sigma) \prod_{i}^{N} \mathrm{Mult}(z_{di} \mid \theta_d)\right]$$

$$\times \left[\prod_{k} \frac{\Gamma(\beta)\Gamma(W\beta)}{\Gamma(N_k + W\beta)} \prod_{w} \frac{\Gamma(N_{kw} + \beta)}{\Gamma(\beta)}\right]$$

We will first integrate out the latent variable Φ. The first thing that can be noticed is that every φ_k is independently sampled, and thus can be integrated separatedly:

$$p(\Theta, S, Z, W \mid \alpha, \beta, \eta, \sigma) = \int_{\varphi_0} \int_{\varphi_1} \dots \int_{\varphi_K} p(\Phi, \Theta, S, Z, W \mid \alpha, \beta, \eta, \sigma) \tag{2}$$

At this point we need to rewrite part of equation (1) in order to be able to continue the derivation:

$$\prod_{d=1}^{D} \prod_{i=1}^{N_d} \mathrm{Mult}(w_{di} \mid \varphi_{z_d}) = \prod_{d=1}^{D} \prod_{w=1}^{W} \prod_{k=1}^{K} \mathrm{Mult}(w \mid \varphi_k)^{N_{dk}} \tag{3}$$

Where N_d represents the total number of words within document d, and N_{dk} representing the number of words within document d assigned to topic k.

$$p(\Theta, S, Z, W \mid \alpha, \beta, \eta, \sigma) = \left[\prod_{k=1}^{K} \int_{\varphi_k} \mathrm{Dir}(\varphi_k \mid \beta) \prod_{d=1}^{D} \prod_{w=1}^{W} \mathrm{Mult}(w \mid \varphi_k)^{N_{dk}}\right] \times$$
$$\left[\prod_{d=1}^{D} \mathrm{Dir}(\theta_d \mid \alpha) \mathcal{N}(s_d \mid \eta^\top \bar{z}_d, \sigma) \prod_{i=1}^{N_d} \mathrm{Mult}(z_{d,i} \mid \theta_d)\right] \tag{4}$$

We can now make use of the definition of the Dirichlet-Multinomial distribution in order to solve all the integrals:

$$p(\Theta, S, Z, W \mid \alpha, \beta, \eta, \sigma) = \left[\prod_{k=1}^{K} \frac{\Gamma(W\beta)}{\Gamma(N_k + W\beta)} \prod_{w=1}^{W} \frac{\Gamma(N_{kw} + \beta)}{\Gamma(\beta)}\right] \times$$
$$\left[\prod_{d=1}^{D} \mathrm{Dir}(\theta_d \mid \alpha) \mathcal{N}(s_d \mid \eta^\top \bar{z}_d, \sigma) \prod_{i=1}^{N_d} \mathrm{Mult}(z_{d,i} \mid \theta_d)\right] \tag{5}$$

We can proceed in an analogous fashion in order to integrate out the latent Θ parameter:

$$p(S, Z, W \mid \alpha, \beta, \eta, \sigma) = \int_{\theta_0} \int_{\theta_1} \dots \int_{\theta_D} p(\Theta, S, Z, W \mid \alpha, \beta, \eta, \sigma) \tag{6}$$

$$p(S, Z, W \mid \alpha, \beta, \eta, \sigma) = \left[\prod_{k=1}^{K} \frac{\Gamma(W\beta)}{\Gamma(N_k + W\beta)} \prod_{w=1}^{W} \frac{\Gamma(N_{kw} + \beta)}{\Gamma(\beta)}\right] \times$$
$$\left[\prod_{d=1}^{D} \mathcal{N}(s_d \mid \eta^\top \bar{z}_d, \sigma) \frac{\Gamma(K\alpha)}{\Gamma(N_d + K\alpha)} \prod_{k=1}^{K} \frac{\Gamma(N_{dk} + \alpha)}{\Gamma(\alpha)}\right] \tag{7}$$

Or, equivalently:

$$p(S, Z, W \mid \alpha, \beta, \eta, \sigma) = \left[\prod_{k=1}^{K} \frac{\Gamma(W\beta)}{\Gamma(\beta)^W \Gamma(N_k + W\beta)} \prod_{w=1}^{W} \Gamma(N_{kw} + \beta)\right] \times$$
$$\left[\prod_{d=1}^{D} \mathcal{N}(s_d \mid \eta^\top \bar{z}_d, \sigma) \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K \Gamma(N_d + K\alpha)} \prod_{k=1}^{K} \Gamma(N_{dk} + \alpha)\right] \tag{8}$$

3.1.2 Collapsed Gibbs Sampler

Using Bayes' theorem, we know that:

$$p(Z \mid S, W, \dots) = \frac{p(S, W \mid Z, \dots)p(Z \mid \dots)}{p(S, W \mid \dots)} \propto p(S, W, Z \mid \dots) \tag{9}$$

Where we did not write all model hyperparameters for the sake of clarity. Thus, in order to implement a Gibbs sampler for this model, we have that:

$$p(z_{d,i} = k \mid Z^{\setminus i}, S, W, \alpha, \beta, \eta, \sigma) \propto p(z_{d,i} = k, Z^{\setminus i}, S, W, \alpha, \beta, \eta, \sigma) =$$

More formulas still need to be written here...

We can further simplify the previous expression by removing those factors that are constant across all possible values for k, resulting in the following final collapsed Gibbs sampler:

$$p(z_{d,i} = k \mid Z^{\setminus i}, S, W, \alpha, \beta, \eta, \sigma) \propto \left[\prod_{k'} \frac{\prod_w \Gamma(N^{\setminus i}_{k'w} + \mathbb{I}(k' = k \wedge w = w_{di}) + \beta)}{\Gamma(N^{\setminus i}_{k'} + \mathbb{I}(k' = k) + W\beta)}\right] \times$$
$$\mathcal{N}\left(s_d \,\middle|\, \eta^T \cdot \frac{N^{\setminus i}_{dk'} + \mathbb{I}(k' = k)}{N_d}, \sigma\right) \prod_{k'} \Gamma(N^{\setminus i}_{dk'} + \mathbb{I}(k' = k) + \alpha) \tag{10}$$

Where we used $\frac{N_{dk}}{N_d} \equiv \bar{z_d}$. The Gibbs sampler can also be expressed in log-space probabilities, in order to get a more numerically-stable implementation:

$$\log p(z_{d,i} = k \mid Z^{\setminus i}, S, W, \alpha, \beta, \eta, \sigma) \propto$$
$$\sum_{k',w} \left[\log \Gamma(N^{\setminus i}_{k'w} + \mathbb{I}(k' = k \wedge w = w_{di}) + \beta) - \log \Gamma(N^{\setminus i}_{k'} + \mathbb{I}(k' = k) + W\beta)\right]$$
$$-\frac{1}{2\sigma^2}\left(s_d - \eta^\top \frac{N^{\setminus i}_{dk'} + \mathbb{I}(k' = k)}{N_d}\right)^2 + \sum_{k'} \log \Gamma(N^{\setminus i}_{dk'} + \mathbb{I}(k' = k) + \alpha)$$

3.1.3 Rewriting the log-gamma function

The logarithm of the gamma function can be rewritten as follows [**?**]:

$$\log \Gamma(z) = -\gamma z - \log z + \sum_{j=1}^{\infty} \left[\frac{z}{j} - \log\left(1 + \frac{z}{j}\right)\right] \tag{11}$$

where γ is the Euler-Mascheroni constant. We apply this to $\sum_k \sum_w \log\Gamma(N_{kw}+\beta)$, which then becomes:

$$= \sum_{k,w} -\gamma(N_{kw}+\beta) - \log(N_{kw}+\beta) + \sum_{j=1}^{\infty} \frac{N_{kw}+\beta}{j} - \log\left(1+\frac{N_{kw}+\beta}{j}\right)$$

$$= -\gamma(N+KW\beta) - \sum_{k,w} \log(N_{kw}+\beta) - \sum_{j=1}^{\infty} \frac{N_{kw}+\beta}{j} - \log\left(\frac{N_{kw}+\beta+j}{j}\right)$$

Note that the term $-\gamma(N+KW\beta)$ serves as a normalisation constant for this dataset. Since we do not need the exact probabilities but only the proportional probabilities during the algorithms execution, we can discard those terms.

$$\Rightarrow - \sum_{k,w} \log(N_{kw}+\beta) - \sum_{j=1}^{\infty} \frac{N_{kw}+\beta}{j} - \log\left(\frac{N_{kw}+\beta+j}{j}\right)$$

$$= - \sum_{k,w} \log(N_{kw}+\beta) - \sum_{j=1}^{\infty} \frac{N_{kw}+\beta}{j} - \log(N_{kw}+\beta+j) + \log(j)$$

$$= - \sum_{k,w} \log(N_{kw}+\beta) - \sum_{j=1}^{\infty} \left(\frac{N_{kw}+\beta}{j} + \log(j)\right) + \sum_{j=1}^{\infty} \log(N_{kw}+\beta+j)$$

$$= \sum_{j=1}^{\infty} \left(\frac{N+KW\beta}{j} + KW\log(j)\right) - \sum_{k,w} \log(N_{kw}+\beta) + \sum_{j=1}^{\infty} \log(N_{kw}+\beta+j)$$

$$= \sum_{j=1}^{\infty} \left(\frac{N+KW\beta}{j} + KW\log(j)\right) - \sum_{k,w} \sum_{j=0}^{\infty} \log(N_{kw}+\beta+j)$$

Again, $\sum_{j=1}^{\infty} \frac{N+KW\beta}{j} + KW\log(j)$ is a constant for this dataset, so we can discard it. This results in the following proportionality:

$$\sum_{k,w} \log\Gamma(N_{kw}+\beta) \propto - \sum_{k,w} \sum_{j=0}^{\infty} \log(N_{kw}+\beta+j) \tag{12}$$

Using similar steps, we can also simplify

$$\sum_{k} \log\Gamma(N_{k}+W\beta) \propto - \sum_{k} \sum_{j=0}^{\infty} \log(N_{k}+W\beta+j) \tag{13}$$

$$\sum_{k,d} \log\Gamma(N_{dk}+\alpha) \propto - \sum_{k,d} \sum_{j=0}^{\infty} \log(N_{dk}+\alpha+j) \tag{14}$$

$$\sum_{d} \log\Gamma(N_{d}+K\alpha) \propto - \sum_{d} \sum_{j=0}^{\infty} \log(N_{d}+K\alpha+j) \tag{15}$$

Putting these together gives us the following proportionality:

$$\log p(z_{d,i} = k \mid Z^{\setminus i}, S, W, \alpha, \beta, \eta, \sigma) \propto$$

$$\sum_{j=0}^{\infty}\sum_{k'}\log(N_{k'}^{\setminus i}+\mathbb{I}(k'=k)+W\beta+j)-\log(N_{dk'}^{\setminus i}+\mathbb{I}(k'=k)+\alpha+j)-$$

$$\sum_{j=0}^{\infty}\sum_{k',w}\log(N_{k'w}^{\setminus i}+\mathbb{I}(k'=k\wedge w=w_{di})+\beta+j)-\frac{1}{2\sigma^2}\left(s_d-\eta^\top\frac{N_{dk'}^{\setminus i}+\mathbb{I}(k'=k)}{N_d}\right)^2$$

In practice, however, we cannot calculate the infinite terms of the above sums, and thus we would stop the approximation at some arbitrary iteration J:

$$\log p(z_{d,i}=k\mid Z^{\setminus i},S,W,\alpha,\beta,\eta,\sigma)\propto$$
$$-\frac{1}{2\sigma^2}\left(s_d-\eta^\top\frac{N_{dk'}^{\setminus i}+\mathbb{I}(k'=k)}{N_d}\right)^2+\sum_{k'}\sum_{j=0}^{J}\log(N_{k'}^{\setminus i}+\mathbb{I}(k'=k)+W\beta+j)$$
$$-\log(N_{dk'}^{\setminus i}+\mathbb{I}(k'=k)+\alpha+j)+\sum_{w}\log(N_{k'w}^{\setminus i}+\mathbb{I}(k'=k\wedge w=w_{di})+\beta+j)$$

3.1.4 Estimating response parameters

The maximum a posteriori (MAP) estimate for the η hyperparameter can be inferred from the gradient of the complete model likelihood:

$$\nabla_{\eta_k}\log p(S,Z,W\mid\alpha,\beta,\eta,\sigma)=\sum_d\frac{\frac{N_{dk}}{N_d}\left(s_d-\eta^T\frac{N_{d\cdot}}{N_d}\right)}{\sigma^2}=$$

$$=\sum_d\frac{s_d\frac{N_{dk}}{N_d}}{\sigma^2}-\sum_d\frac{\frac{N_{dk}}{N_d}\left(\eta^T\frac{N_{d\cdot}}{N_d}\right)}{\sigma^2}=0$$

$$\Rightarrow\sum_d s_d\frac{N_{dk}}{N_d}=\sum_d\frac{N_{dk}}{N_d}\left(\sum_{k'}\eta_{k'}\frac{N_{dk'}}{N_d}\right)=\sum_d\frac{N_{dk}}{N_d}\left(\eta_k\frac{N_{dk}}{N_d}+\sum_{k'\neq k}\eta_{k'}\frac{N_{dk'}}{N_d}\right)$$

$$\Rightarrow\sum_d s_d\frac{N_{dk}}{N_d}=\eta_k\sum_d\left(\frac{N_{dk}}{N_d}\right)^2+\sum_d\left(\frac{N_{dk}}{N_d}\sum_{k'\neq k}\eta_{k'}\frac{N_{dk'}}{N_d}\right)$$

$$\Rightarrow\sum_d\left(s_d\frac{N_{dk}}{N_d}-\frac{N_{dk}}{N_d}\sum_{k'\neq k}\eta_{k'}\frac{N_{dk'}}{N_d}\right)=\eta_k\sum_d\left(\frac{N_{dk}}{N_d}\right)^2$$

$$\Rightarrow\sum_d\frac{N_{dk}}{N_d}\left(s_d-\sum_{k'\neq k}\eta_{k'}\frac{N_{dk'}}{N_d}\right)=\eta_k\sum_d\left(\frac{N_{dk}}{N_d}\right)^2$$

$$\Rightarrow\eta_k=\frac{\sum_d\frac{N_{dk}}{N_d}\left(s_d-\sum_{k'\neq k}\eta_{k'}\frac{N_{dk'}}{N_d}\right)}{\sum_d\left(\frac{N_{dk}}{N_d}\right)^2}$$

Trying to apply the previous formula as an update rule for η does not converge. Instead, the following update can be used:

$$\eta_k^{new} \leftarrow (1-\gamma)\eta_k^{old} + \gamma \frac{\sum_d \frac{N_{dk}}{N_d}\left(s_d - \sum_{k' \neq k} \eta_{k'} \frac{N_{dk'}}{N_d}\right)}{\sum_d \left(\frac{N_{dk}}{N_d}\right)^2 + \varepsilon}$$

With $1 \gg \gamma > 0$ in order for the previous series to converge and $1 \gg \varepsilon > 0$ is a smoothing constant.

4 Experiments

For our experiments we chose to use two different performance measures. The perplexity measure is commonly used within the Latent Dirichlet Allocation literature, and we included it for reference purposes. It is defined as follows:

$$perplexity \equiv \exp\left\{\frac{-\sum_{i=1}^{N} \log p(x_i)}{N}\right\} \tag{16}$$

Where $p(x_i)$ represents the probability the model gives for the i-th item within the test set. The lower the perplexity, the less "surprised" the model is of seeing its input, and thus better models it. This measure has a shortcoming, though. It is enough for the model to give a probability of zero to a word to drive this performance measure to infinity. This does not mean, however, that the model is infinitely bad, since the rest of the items might still have large probabilities. To overcome this problem, we also measure the performance in terms of the average inverse accuracy of the model:

$$accuracy^{-1} \equiv \frac{N}{\sum_{i=1}^{N} p(x_i)} \tag{17}$$

This measure should be interpreted as the number of words the model incorrectly predicts for each correctly predicted word. The lower this magnitude, the better the model.

We created a first experiment in order to check the validity of our approximation for the Gibbs sampler of our model. We selected different values for J and compared the predictive performance for each case. We ran three different MCMC chains and then averaged the results. Within each execution, 100 movies were randomly selected as the training set, and 20 as the testing set. We used 5 samples, taken every 4 steps of the Gibbs sampler, with the first 20 iterations discarded as the burn-in period. Our findings are summarized in table 4:

We can observe that the predictive performance tends to improve with larger values of J, as it should be expected, since the higher J is, the better the approximation becomes. This improvement is not, however, very consistent

J	1	2	3	4	5	6
Perplexity	4100	4125	4082	4251	3754	3881
Accuracy^{-1}	858	871	823	863	877	815

Table 1: Some caption here...

even when averaging across 3 chains, and thus we theoretize that the extra computational time spent on making J higher is better used if J is kept as low as possible and more movies and/or iterations are used instead. We then conclude this approximation is interesting, at least from a computational perspective.

We devised a second experiment in order to find out good values for the number of topics to use in our models. Again we did an equivalent set up to the one for the previous experiment, but this time changing the number of topics and whether or not the Gibbs sampler uses the movie score information.

Perplexity measure:

LDA topics		25	50	100
Using scores?	No	4128	5333	6954
	Yes	**4005**	5503	7082

Inverse accuracy measure:

LDA topics		25	50	100
Using scores?	No	845	1318	2049
	Yes	**805**	1372	2067

Figure 3: Some caption here...

We can see how all models start overfitting with, at least, more than 25 topics. This is most likely due to the fact that we only used 100 movie plot summaries for training, which is a rather small dataset. We also observe that, for 25 topics, not only the best predictive performance is archived, but also using movie results in an improvement. This improvement is very small, but we could systematically observe it in all the tests we run for this work.

5 Discussion