Danmarks Tekniske Universitet

42186: Model-Based Machine Learning

Predicting Hotel Review Ratings Using Probabilistic Models and Neural Inference

Markus Swegmark
s250031

Evita Karietaite
s232425

Ioannis Vlasakoudis
s232755

Andrea Madelena L. Bolvig
s211636

Date: May 29, 2025

Contents

1 Abstract

In this project, we investigate the use of probabilistic modeling and neural inference techniques to predict hotel review ratings based on text reviews. Using a dataset of over 20,000 TripAdvisor reviews, we implement a neural variational version of the Product of Experts Latent Dirichlet Allocation (ProdLDA) model to extract meaningful topics from the review texts. These topic distributions are combined with either reconstructed word distributions or raw bag-of-words representations to predict numerical ratings through a joint generative classification model. Inference is performed using Stochastic Variational Inference (SVI) with Pyro, and the model is evaluated using standard metrics such as MSE, MAE, R^2, and correlation. While different configurations were explored, a model combining topic proportions and raw word counts yielded the best predictive performance. The study highlights the potential of integrating topic modeling with rating prediction for enhancing decision-making in digital review platforms.

2 Introduction

Tourism is one of the fastest-growing industries worldwide, and in some countries, national budgets heavily rely on this sector. One of the key elements of an enjoyable vacation is a comfortable and well-rated hotel that meets customer expectations. In the modern digital age, online reviews have become one of the most accessible and influential sources of information about hotels.

Online reviews provide valuable insights into user satisfaction and impressions, often conveyed through text. These reviews can be further analyzed to extract sentiment or even predict a numerical rating. In this project, we analyze hotel review data from TripAdvisor, one of the largest hotel rating platforms in the world.

Having both sentiment and a numerical rating associated with each review can help potential customers make more informed decisions without needing to read through all the reviews manually. Additionally, extracting topics from reviews can enable more efficient filtering, allowing users to focus only on the aspects that matter most to them. For example, if a customer is specifically looking for a hotel with an excellent location, they could choose to view only the reviews related to that topic. This not only saves time but also enhances the decision-making process by highlighting relevant user experiences.

The main goal of this project is to analyze hotel reviews in order to predict user sentiment and numerical ratings based on the content of the reviews. Specifically, we aim to extract latent topics from the textual reviews using Latent Dirichlet Allocation (LDA), allowing us to identify recurring themes and topics that influence user satisfaction. Also, perform sentiment analysis on the reviews to determine the underlying sentiment (positive or negative) associated with each review. Build a classification model that can accurately predict the numerical rating (e.g., 1 to 5 stars) given by the customer based on the textual content and derived features. Finally, evaluate the performance of the chosen models and techniques, and analyze how well they capture and represent user opinions.

By achieving these goals, we hope to demonstrate the potential of combining topic modeling and sentiment analysis to support hotel selection and improve user experience in travel platforms.

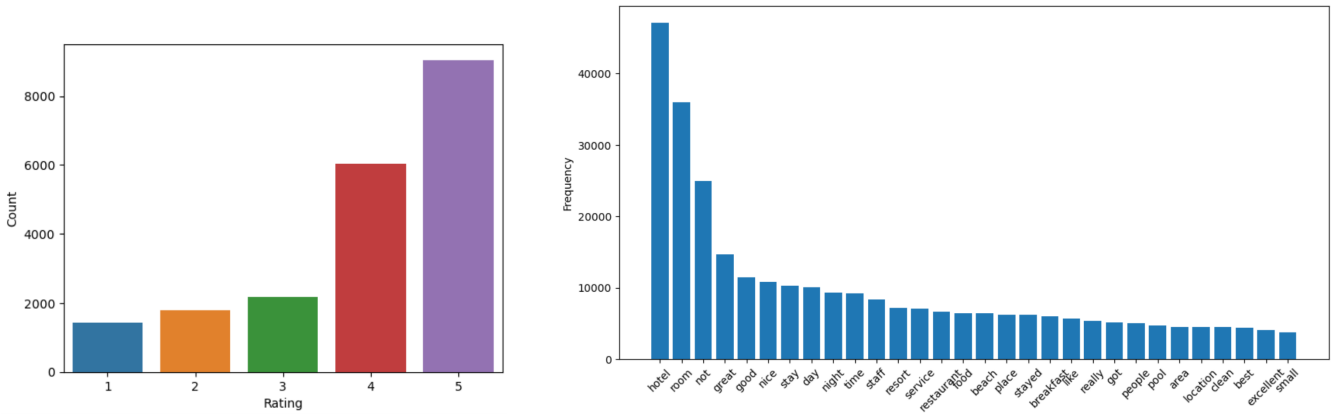
The structure of this report is as follows: section 3 provides an overview of the dataset used in our analysis. Section 4 describes the methods and techniques applied. Section 5 presents the results of our analysis. Section 6 concludes the report with a summary of findings and discussion of the challenges encountered. Finally, section 7 presents the conclusions.

3 Data

The dataset we use consists of 20,491 unique reviews from TripAdvisor. It contains two columns: the review text and the corresponding rating. The dataset originates from the publication by Md. Harun-Al-Naser Alam et al[1].

The ratings range from 1 to 5. The rating distribution is shown in Figure 1a, where more than 40% of the ratings are 5.

The reviews vary in both length and content. The shortest review contains 7 words, while the longest has 1,931 words. The average review length is 104 words. In total, the dataset contains 102,008 unique words. The most frequently occurring words are shown in Figure 1b.



(a) Rating distribution

(b) 30 most common words in reviews

Figure 1: Overview of review data from TripAdvisor

4 Methods

4.1 Preprocessing the data

Before analyzing the text data, we cleaned it using a few common preprocessing steps. First, text was converted to lowercase, and non-alphabetic characters were removed using regular expressions to remove punctuation, digits, and special symbols. To vectorize the text, it was cleaned, tokenized, words replaced by their stems, and then used a trigram-based CountVectorizer. This can help capture the meaning of phrases. Common English stop words except "not" were removed, and terms appearing in fewer than five documents or in more than 50% of documents were filtered out to reduce rare terms and enhance topic quality. Finally, the resulting sparse term-document matrix was transformed into a PyTorch tensor which will be used as as input for the neural topic model.

4.2 Model

Our implementation was done using the Pyro interface with variational inference and a guide for estimating the Evidence Lower Bound (ELBO). Several components of the model were approximated using neural networks, which according to academic research has proven to improve performance and training speed of probabilistic models [4, 6].

4.2.1 Natural Language Processing with Product of Experts Latent Dirichlet Allocation (ProdLDA)

For processing the text in reviews and providing an interface for prediction, we use a neural variational implementation of Product of Experts Latent Dirichlet Allocation (ProdLDA)[2]. Our approach introduces neural network approximations for the topic distribution parameter θ, which improves both performance and training speed compared to traditional variational inference methods [7, 5].

In this model, each of the D documents (reviews) is viewed as a mixture of K latent topics, where each topic is a distribution over words. For a given document d with N words, the topic proportions are encoded in θ_d. Each word $w_{m,n}$ is associated with a latent topic assignment $z_{m,n}$, which is sampled from θ_d, and the observed word is then drawn from the corresponding topic-specific word distribution φ_k.

Instead of performing variational inference analytically, we approximate the posterior over θ_d using an encoder-decoder neural architecture. The encoder network takes the bag-of-words representation of document d (denoted as $\mathbf{x}_d$) and maps it to the parameters of a logistic-normal distribution, representing the variational posterior $q(\theta_d|\mathbf{x}_d)$. This encoder consists of two fully connected layers with Softplus activation functions, dropout for regularization, and batch normalization for stability. It outputs both the mean and log-variance of the latent topic distribution.

$$p(\theta \mid \alpha) \approx p(\theta \mid \mu, \Sigma) = \mathcal{LN}(\theta \mid q_\theta(x))$$

where $q_\phi(x) = \mu, \Sigma$ is the encode network

The decoder network takes a sampled topic distribution θ_d and maps it back into the observed word space using a neural network. Specifically, the decoder approximates the probability distribution over the vocabulary using a parameterized function $p_\theta(\theta_d)$, where p_θ is a feed-forward neural network that outputs unnormalized word logits. These logits are then passed through a softmax activation to produce the predicted word distribution:

$$\hat{w}_d = \text{Softmax}(q_w(\theta_d))$$

where $p_\theta(\cdot)$ is the decoder neural network .

This decoded distribution $\hat{w}_d$ aims to reconstruct the word distribution of the original bag-of-words (BoW) input for document d. The reconstruction error, measured by the negative log-likelihood of the observed word counts under $\hat{w}_d$, contributes to the overall variational loss function. This encourages the decoder to learn topic representations that accurately capture the semantics of the input text.

The decoder defines a mapping from latent topic representations to word distributions through $q_w(\theta_d)$, which is optimized jointly with the encoder during training via stochastic variational inference.

4.2.2 Rating Prediction Through Classification

Each document has an observable rating $y_d \in \{1, 2, 3, 4, 5\}$ that we aim to predict. To achieve this, we employ a linear classifier that takes as input a latent reconstruction of the document's word distribution.

This reconstructed representation, denoted $\hat{w}_d$, is given trough the decoder network.

It is then used directly as input to a fully connected layer (classifier), which produces logits over the five possible rating categories:

$$\text{logits}_d = W\hat{w}_d + b$$

The predicted rating y_d is modeled as a categorical variable:

$$y_d \sim \text{Categorical}(\text{softmax}(\text{logits}_d))$$

The classifier is trained jointly with the variational autoencoder components by maximizing the evidence lower bound (ELBO), where the rating likelihood is treated as an observed variable.

4.2.3 Complete Model

The complete model is defined as follows:

$$\begin{aligned}
&\theta_d \sim \mathcal{LN}(\mu, \Sigma),\\
&\theta_d = \text{Softmax}(\log \theta_d),\\
&x_d \sim \text{Multinomial}(N_d, \text{Softmax}(W\theta_d)),\\
&y_d \sim \text{Categorical}(\text{Softmax}(V[\theta_d, x_d] + b)),
\end{aligned}$$

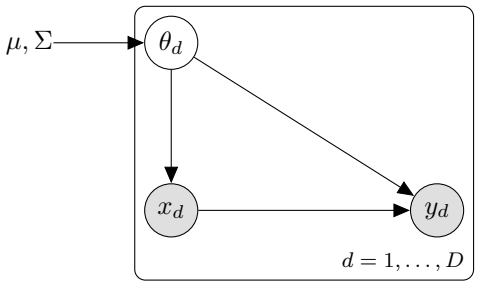


Figure 2: Plate diagram for neural ProdLDA with document-level rating prediction.

Here, D is the number of documents, K is the number of latent topics, and V is the vocabulary size. The topic proportions θ_d are obtained by applying a softmax transformation to a latent variable $\log \theta_d$, which is drawn from a multivariate Gaussian with learned mean and variance (via the encoder q_ϕ). The bag-of-words vector x_d is generated from a Multinomial distribution parameterized by the softmax-normalized decoder output $W\theta_d$, where $W \in \mathbb{R}^{K \times V}$ is the topic-word matrix learned by the decoder p_θ.

The observed rating $y_d \in \{1, 2, 3, 4, 5\}$ is predicted by a linear classifier that takes as input the concatenation of θ_d and x_d, and outputs a distribution over rating categories. This prediction is modeled as a categorical likelihood and is trained jointly with the topic model using stochastic variational inference.

Algorithm 1 Generative Process for Neural ProdLDA with Ratings

for each document $d = 1$ to D **do**
  Draw $\log \theta_d \sim \mathcal{N}(\mu, \Sigma)$
  Compute $\theta_d = \text{Softmax}(\log \theta_d)$
  Draw word counts $x_d \sim \text{Multinomial}(N_d, \text{Softmax}(W\theta_d))$
  Draw rating $y_d \sim \text{Categorical}(\text{Softmax}(V[\theta_d, x_d] + b))$
end for

4.3 Inference

In this model, inference is performed using Stochastic Variational Inference (SVI). We choose this inference method for both theoretical and practical reasons. Markov Chain Monte Carlo (MCMC) methods scale poorly in high-dimensional spaces and with large datasets, such as ours, which involves topic modeling with more than 20,000 documents and over 15,000 tokens . On the other side, SVI is well-suited to this setting and scales well with LDA models.

In our implementation, the encoder learns to approximate the topic proportions for each review by predicting the mean and variance of a distribution over the document's topic mixture. During SVI, these parameters are optimized by minimizing the Evidence Lower Bound (ELBO). During training, rating prediction also contributes to the ELBO, since it is treated as an additional observation. Thus, the model learns the topics, while classifying the reviews by rating, combining this way both unsupervised topic modeling with supervised learning.

In the inference, we also used the Adam optimizer with a learning rate of 0.01, which provides a balance between training speed and stability. This learning rate was selected, because smaller values led to slower convergence without improving accuracy, while larger values made the ELBO more unstable.

4.4 Model Evaluation

To evaluate the model's goodness of fit, we use four metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R^2), and Correlation.

Mean Squared Error (MSE): Measures the average of the squared differences between the predicted and actual values. It penalizes larger errors more heavily, making it sensitive to outliers.

Mean Absolute Error (MAE): Calculates the average of the absolute differences between the predicted and actual values. It provides an intuitive measure of the average prediction error in the same units as the target variable.

R-squared (R^2): Represents the proportion of variance in the dependent variable that is explained by the model. It ranges from 0 to 1, with higher values indicating a better fit.

Correlation: Assesses the strength and direction of the linear relationship between the predicted and actual values. It ranges from -1 to 1, where values closer to 1 indicate a strong positive linear correlation.

5 Results

Metric	Value
Mean Squared Error	0.8174
Mean Absolute Error	0.4244
R-squared	0.4623
Correlation	0.7289

Table 1: Rating prediction metrics

Table 1 presents the evaluation metrics for the rating prediction model, as discussed in Section 4.4. The relatively low Mean Squared Error (MSE) and Mean Absolute Error (MAE) indicate that the model performs reasonably well in predicting the numerical ratings based on review text. The R^2 value of approximately 0.46 suggests that the model explains about 46% of the variance in the actual ratings, which is a meaningful proportion given the complexity of natural language data. Additionally, the correlation coefficient of 0.73 indicates a strong and positive linear relationship between the predicted and actual ratings, further supporting the effectiveness of the model.

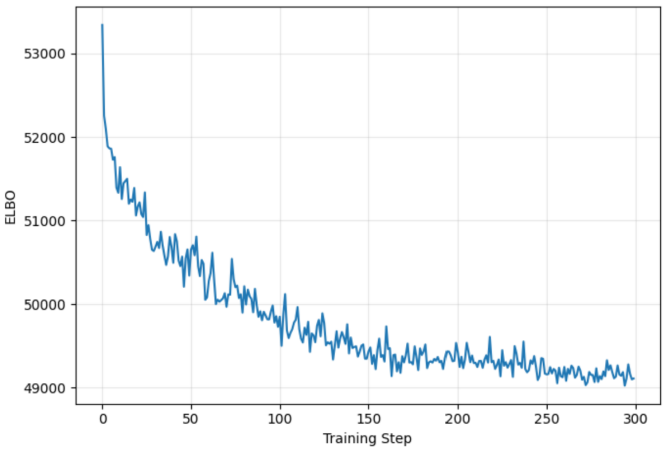


Figure 3: ELBO

Figure 3 shows the ELBO across training steps. The most rapid decrease occurs during the first 100 steps, after which the rate of decline slows down. From around step 150 onward, the ELBO stabilizes and remains relatively constant until the final training step at 300. This indicates that the model has likely converged, and further training would not yield significant improvements.

The bar chart in Figure 4 illustrates the correlation between each extracted LDA topic and the corresponding numerical ratings provided by hotel guests. Each bar represents a topic, with green bars indicating a positive

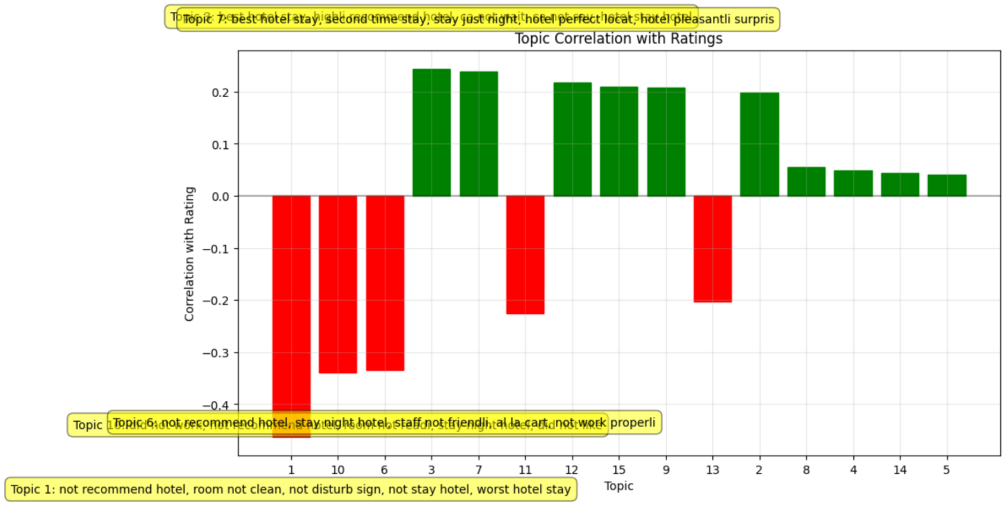


Figure 4: Topic correlations

correlation (topics that tend to appear in higher-rated reviews), and red bars indicating a negative correlation (topics commonly found in lower-rated reviews). Topics positively correlated with high ratings, e.g., Topic 7, include phrases such as "best hotel stay," "second time stay," and "hotel perfect location." These themes typically reflect a positive guest experience, often associated with repeated visits, satisfaction with location, or overall pleasant surprises. Topics negatively correlated with ratings, e.g., Topics 1 and 6, include terms like "not recommend hotel," "room not clean," and "staff not friendly." These topics clearly reflect guest dissatisfaction and are often linked with poor service, unclean conditions, or maintenance issues.

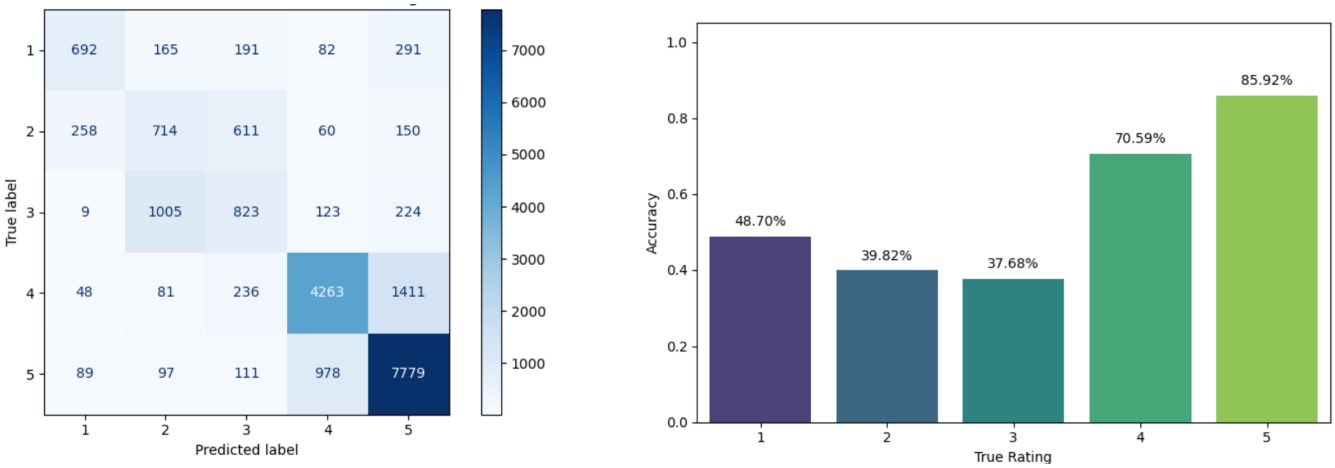
Topic 1	**Weight**	**Topic 3**	**Weight**
not recommend hotel	6.3333	best hotel stay	9.3399
room not clean	1.3705	highli recommend hotel	3.4363
not disturb sign	1.2634	ca not wait	3.0314
not stay hotel	1.2271	ca not say	3.0279
worst hotel stay	1.2020	hotel stay hotel	2.5858
did not work	1.1970	highli recommend stay	2.1380
room did not	1.1728	not wait return	1.9500
not wast money	1.1487	great hotel stay	1.9357
not work room	1.1451	saona island tour	1.8612
stay hotel not	1.0754	not say good	1.8048

Table 2: Top phrases and weights for Topic 1 and Topic 3

Table 2 presents the top phrases and their corresponding weights for most positive and most negative topics identified during topic modeling. Topic 1 is dominated by strongly negative expressions such as "not recommend hotel," "room not clean," and "worst hotel stay," indicating that this topic captures highly dissatisfied customer experiences. The high weight of the phrase "not recommend hotel" (6.33) suggests that this sentiment is a key theme in this topic.

In contrast, Topic 3 reflects highly positive sentiments, with phrases like "best hotel stay," "highly recommend hotel," and "great hotel stay" appearing with large weights. The strongest phrase, "best hotel stay," has a notably high weight of 9.34, indicating that this topic is closely associated with favorable guest experiences and strong endorsements.

These topics demonstrate the model's ability to distinguish between negative and positive themes in hotel reviews, which is valuable for understanding customer satisfaction and guiding future visitors toward relevant feedback.



(a) Confusion matrix

(b) Accuracy by rating class

Figure 5: Evaluation metrics

Figure 5 presents the evaluation metrics for the classification model. In the confusion matrix shown in Figure 5a, we observe that the model often predicts either the correct rating or a neighboring rating, indicating reasonable performance in capturing rating proximity. However, for rating class 1, there is a notable tendency to incorrectly predict class 5, likely due to class 5 being the most frequent label in the dataset. Additionally, we see clustering between classes 2 and 3, as well as 4 and 5, suggesting that the model struggles to clearly distinguish between these adjacent rating levels.

Figure 5b shows the classification accuracy for each individual rating class. The overall accuracy of 69.65% is a moderate result for a text-based classification task. The model performs best on higher ratings—particularly classes 4 and 5—while it has more difficulty correctly identifying mid-range ratings, especially classes 2 and 3.

6 Discussion

In our classification setup, the goal was to predict the review rating using a one-layer neural network. The output of the network is a vector Φ of size 5, where each element represents the predicted probability of each possible rating from 1 to 5. We tried two different approaches in terms of the inputs for the neural network. The main component in both cases was the θ vector, which represents the topic distribution for each document.

- **Model 1: Topic vector + ϕ (word distributions per topic)**
  In Model 1, we replaced the BoW with the topic-word distribution ϕ. It describes how likely each word is for each topic. This variable is stochastic and part of the generative topic model. The model used only θ and ϕ to predict the rating. This is also described in section 4.2.2. The classifier is trained jointly with the variational autoencoder components by minimizing the evidence lower bound (ELBO), treating the rating likelihood as an observed variable. While this model aligns closely with the underlying probabilistic graphical model (PGM), it performed worse than Model 2, likely due to the more abstract nature of the reconstructed input $\hat{x}_m$.
- **Model 2: Topic vector + Bag-of-Words (BoW)**
  In Model 2, we used both the topic distribution vector θ and the BoW representation of each document as input to the neural network. The BoW provides the word counts for each review. These two inputs were concatenated into a single feature vector, representing both semantic structure and raw word usage. Each

document has an observable rating $y_m \in \{1, 2, 3, 4, 5\}$ that we want to predict. To do this, we passed the combined feature vector to a fully connected layer (classifier), which outputs logits over the five possible rating categories. The predicted rating is modeled using a categorical distribution over these logits. This classifier is trained jointly with the variational autoencoder (VAE) components by maximizing the evidence lower bound (ELBO), where the classification likelihood is treated as an observed variable. Although BoW is a simple and fixed (deterministic) representation, this model gave better results than using only θ.

We also attempted to predict ratings using linear regression, but the model only predicted the average rating. We believe this is because the model's weights failed to capture the variance between different ratings. This can sometimes occur due to an overly restrictive prior. Although we experimented with changing the prior and other parameters, which did increase variation in the model weights, the predicted ratings remained centered around the mean. Based on the findings of the three approaches regarding the classification problem, we chose to go with Model 1. Although Model 2 gave better results, Model 1 is preferred because it aligns with the PGM. Furthermore, it uses only latent variables (θ and φ), and offers a more principled and generative approach to prediction.

Detecting negative sentiment in reviews was challenging due to the subtle and varied ways people express negativity. Using only tokenization was not effective enough in capturing negative reviews and creating topics with negative sentiment, since positive and negative sentiment reviews may had the tokens "good" in them, but the first was initially "was good" and the latest "was not good". To handle this initially, we tried to combine each instance of the word 'not' with the following word. The idea behind this, was to capture sentiment e.g., "not good", but in the end, it was still ineffective. So, we resorted to using tri-grams to capture sequences of three words, allowing us to detect patterns like "did not work" or "not worth buying". As we can see in the results, this method captured the negative sentiment of some reviews better than the bi-grams we also tried. A reason for this could be that sentiment was expressed in more subtle ways that weren't captured by the simpler bi-grams. Other methods we could have used include more advanced models, such as dependency parsing[3] or transformers that understand sentence structure more deeply. However, this could have increased complexity and the risk of miss-classification in terms of incorrectly interpreting phrases like "not only good but great" as negative.

During the training of the final model, we experimented a lot with the different key hyperparameters to identify the best combination. Starting with the number of epochs, we settled at 300, since the model has likely converged by that point, as mentioned in the Results section. For the learning rate, we started with smaller values, but the convergence was slower and the performance poor. At the end, we used the highest possible learning rate that maintained stability and the quality of the posterior approximation. Last but not least, is the number of topics. Although our dataset was large, using 15 topics did not led to underfitting. We experimented with values being at the range of 15-30, and 15 was the best in terms of results. We even explored the stick-breaking construction approach, in order to enable the model to infer the optimal number of topics automatically. However, due to time constrains, we were unable to fully implement this method and fine tune it.

Additionally, we created a balanced dataset with an equal number of reviews for each rating, as the original dataset was biased — over 40% of the reviews were rated as 5. However, this approach did not improve the model's predictive power. In fact, it further confirmed that the model was simply predicting the average rating, as all predictions then clustered around 2.5, so we proceeded with the whole dataset as it is.

7 Conclusion

In conclusion, we ended up choosing the model that did not perform the best, but it aligned with the PGM and together with its use of latent variables, it offered a more principled and generative approach. Additionally, we faced challenges regarding sentiment detection, especially for negative reviews. We found that simple tokenization was not enough to identify negative sentiment. The use of tri-grams helped to capture more nuanced expressions, however, more advanced methods could further improve this.

When tuning the hyperparameters regarding the model performance, 300 epochs, a carefully chosen learning rate, and 15 topics resulted in the best outcomes. Additionally, creating a balanced dataset with an equal number of reviews for each rating did not enhance the prediction accuracy.

Overall, our model may not be the most accurate, but it can still offer valuable insights and help users choose hotels based on what matters most to them. With further development, this approach could become more powerful and useful in real-world situations.

References

[1] Md. Harun-Al-Naser Alam, Won-Joong Ryu, and Sang-goo Lee. "Joint Multi-grain Topic Sentiment: Modeling Semantic Aspects for Online Reviews". In: *Information Sciences* 339 (2016), pp. 206–223. DOI: 10.1016/j.ins.2016.01.024.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent Dirichlet allocation". In: *Journal of Machine Learning Research* 3.Jan (2003), pp. 993–1022.

[3] Shivanee Jaiswal. *Natural Language Processing – Dependency Parsing*. 2021. URL: https://towardsdatascience.com/natural-language-processing-dependency-parsing-cf094bbbe3f7/ (visited on 05/28/2025).

[4] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[5] Yishu Miao, Lei Yu, and Phil Blunsom. "Neural variational inference for text processing". In: *Proceedings of the 33rd International Conference on Machine Learning* (2016), pp. 1727–1736.

[6] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models". In: *arXiv preprint arXiv:1401.4082* (2014).

[7] Akash Srivastava and Charles Sutton. "Autoencoding variational inference for topic models". In: *arXiv preprint arXiv:1703.01488* (2017).