Measuring the Impact of Text Summarization Before Topic Modeling on Travel Reviews

Serena Elizabeth Alvarez
sealvare@uci.edu

Atharva Deodhar
deodhara@uci.edu

Nilabjya Ghosh
nilabjyg@uci.edu

Joshua Michael Tucker
jmtucker@uci.edu

1 Introduction

Automatically analyzing large datasets of English text responses can be extremely difficult in certain applications. Humans have an ability to read the content of the text and then sort these responses out by category, which can provide a much more useful look at the overall statistics of the dataset. The problem is that this would require a human looking through every response in a potentially massive dataset.

Topic modeling is a way for machine learning to simulate that same process. By automatically learning and separating responses by topic, we can enable statistical analysis of massive datasets without manual human coverage. In this paper, we will be demonstrating this by extracting insights on a dataset of Tripadvisor review texts.

Latent Dirichlet Allocation, or LDA, is a kind of Bayesian topic modeling, which assumes the text can be split into a series of many small documents with relatively few topics each. It can be used to automatically discover topics and then categorize the documents into those topics on larger datasets than would be feasible for humans to manually read and categorize.

2 Methodology

To carry out our experiment, we will be using Latent Dirichlet Allocation (LDA) to preform topic modeling on a Trip Advisor Reviews dataset [7].

We will be creating two models: (1) On the full, unchanged dataset, and (2) On a transformed dataset with summarized reviews. Those models will be evaluated using log-likelihood and Topic Difference plot against itself then, will be compared using the metrics of perplexity, and coherence.

2.1 Model Overview: Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a popular statistical model well-suited for use with text data. In LDA, documents are represented as random mixtures over latent topics, each with a distribution over words. We generate in the LDA structure, fit our data to the model, then we evaluate and interpret the results.

Shown graphically in Figure 1 from [3], the LDA estimates K topics β and an assigns each word to a topic Z, given an input of D documents with N words and and number of topics K to generate. It estimates the proportions of each topic per-document θ and generalizes two parameters α and ν corresponding to the contents and proportions of each topic in the general case.

Preprint. Under review.

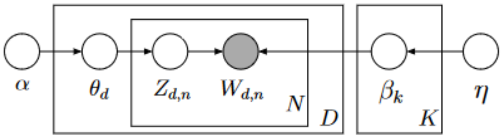


Figure 1: Graphical representation of the LDA Model.

2.1.1 Generative process for LDA

Adapted from the 2003 paper "Latent Dirichlet Allocation" [4], the simple process for generating a document w with LDA is as follows:

Choose the number of words N in document w:

$$N \sim \text{Poisson}(\xi)$$

Choose the topic distribution θ:

$$\theta \sim \text{Dir}(\alpha)$$

For each N words in the document:

Choose z_n based on topic distribution

$$z_n \sim \text{Multinomial}(\theta)$$

Then select a word conditioned on the topic z_n

$$w_n \sim p(w_n \mid z_n, \beta)$$

This will give us a A k-dimensional Dirichlet random variable of topics and words in a document will follow the joint distribution of:

$$p(\theta \mid \alpha) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{k} \alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \tag{1}$$

The joint distribution of the topic mixture θ, a set of N topics z, and N words w over the vocabulary V is given by:

$$p(\theta, z, w \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \tag{2}$$

where β is a k X V matrix where $\beta_{ij} = p(w^j = 1 | z^i = 1)$.

2.1.2 Model learning and evaluation with LDA

We will be using these three metrics to measure the performance and compare our models.

Log-Likelihood The primary way we assessed our models' performance during testing was to measure the probability of generating the input corpus given the model state. The simplest metric used to calculate this was taking the log-likelihood of producing the tokens required to generate the input corpus for a particular model. This log-likelihood can be expressed as:

$$\log p(\tilde{\vec{w}} | M) = \sum_{t=1}^{V} n_{\tilde{m}}^{(t)} \log(\sum_{k=1}^{K} \phi_{k,l} \cdot \theta_{\tilde{m},k})$$

as shown by [2], using $\phi_{k,t}$ to indicate the parameter for word t from topic k, and $\theta_{m,k}$ to indicate the parameter of topic k in document m; where M is the trained model, w_m is the word vector corresponding to document m, and $n_m^{(t)}$ is the number of times that word t appears in document m.

Perplexity Perplexity is a variation on the log-likelihood metric that was used to assess performance and in the initial phases to select cluster quantities. It is found by taking the reciprocal geometric mean of the log-likelihoods shown above:

$$p(\vec{W} | M) = exp\left(-1 * \frac{\sum_{m=1}^{M} \log p(\tilde{\vec{w}} | M)}{\sum_{m=1}^{M} N_m}\right)$$

as also shown in [2], for the same values as log-likelihood.

Coherence The team also assessed a quantitative coherence metric while selecting a cluster quantity for the model and to verify model improvement after training. We used the C_v coherence metric as given by [3]. (While writing this report, we came across research suggesting that the C_v metric is considered flawed compared to other options, but it is still useful enough for our purposes.) C_v compares each word to the entire set of words in a limited range using a context window, computes the normalized log-ratio for each to occur, and returns the mean of the results.

Results Table 1 lists the Perplexity and Coherence scores of the final models used.

Model	Full Text	Summarization
Perplexity	-7.45880	-8.16618
Coherence	0.36431	0.37663

Table 1: Final Model Performance

2.2 Related Work

To understand how our project fits in the evolving landscape of topic modeling, we have reviewed several papers that lay the foundations for our theory and give us insight into conducting this experiment. The foundational paper by Blei, Ng, and Jordan introduced the mathematical foundation of LDA and offered a generative model for documents that we base our understanding on. Additionally, research efforts such as those in "Applying LDA Topic Modeling in Communication Research" [8] further allowed us to understand the practical research applications of LDA.

3 Description

Our goal is to perform topic modeling for Trip Advisor Hotel Reviews. In this regard we will apply Latent Dirichlet Allocation. In order to apply LDA, we will first need to perform text pre-processing on the reviews before giving it as input to the model. Then, we will decide on the number of topics, which happens to be a hyperparameter to the model. Now, we train the model. The innovative approaches such as those discussed in "Smart literature review: a practical topic modeling approach to exploratory literature review" demonstrated how good LDA is at adapting to handle large datasets correctly. It is clear that the use of LDA is still relevant for research purposes- even in an era dominated by new and improved models.

Another approach, which we will use to analyze the reviews is to perform topic modeling on summary of the review.

Finally, we will compare the two LDA models, one created on the basis of the full review, another based on summary of the review and the topics generated by each.

4 Experimental setup and implementation

4.1 Implementation details

4.1.1 LDA Model of All Original Reviews

Steps

1. Use Natural Language Toolkit to generate a list of stop words. Add more stop words to this list
2. Load all the reviews in a Pandas dataframe
3. Use Gensim library to tokenize the sentences of the reviews. In the process. remove emails, newline characters and quotes.
4. Build bigram and trigram models using gensim.models.Phrases
5. Lemmatize the tokens
6. Create a dictionary representation of the reviews.

7. Generate a Bag-Of-Words representation of the reviews.
8. Build and train the LDA Model

```
# Build LDA model
lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
    id2word=id2word,
    num_topics=4,
    random_state=100,
    update_every=1,
    chunksize=10,
    passes=10,
    alpha='symmetric',
    iterations=100,
    per_word_topics=True,
    eval_every=1
)
```

4.1.2 Topic Count Finalisation

Topic Count	Coherence
4	0.3489761927949966
5	0.3404375450679008
6	0.28425225387545966
8	0.3484759940594897
10	0.377142505849098
15	0.3287628934226366

Table 2: Coherence Scores for various Topic Counts

We selected 5000 reviews and trained the LDA model for different topic counts. We observed the highest coherence scores for topic counts 4 and 10. With topic count 10, we noticed that there were identical topic keywords across multiple topics and there was overlap in the topic circles in the inter topic distance map generated with pyLDAvis. Hence, we went ahead with topic count 4.

4.1.3 Summarized Reviews and LDA Comparison

We want to understand if our topic modeler can still find the same or at least similar topics within summarized versions of the reviews. As the original data is preprocessed (special characters and stop words removed, sentences used as tokens), we cannot directly use off-the-shelf Python libraries such as `sumy`. These do not work well with text that is not in plain English. We used part of the code published by user Sandy M. to Medium [9] as it first preprocesses its own sample data similarly to what we are working with in the TripAdvisor dataset to generate 20,491 summaries, which can then directly be fed into the code we wrote for creating topic models.

To compare the full review model and the summarized model, we will first visualize the topic differences between the two. Topic Difference is primarily used to compare topic models to understand topic structure. It can be done within one model or between two models to understand how dissimilar topics are from each other. For our graphs in Figure 5 below, red (or almost red) cells represent strong correlation and blue (or almost blue) cells represent strong decorrelation between topics.

In an ideal world, we would want different topics to be highly decorrelated from each other which means they are have no influence on the other topics. This can be shown in Figure 4, and allows for clearer understanding of the topics and topic model.

In examining the topic differences within the full model, we can see clear strong correlation on the diagonal and moderate decorrelation everywhere else. This is expected and shows that the topics are reasonably decorrelated from the other topics within the model. Similarly, the topic difference plot within the summarized model has very strong correlation on the diagonal and a stronger decorrelation when compared to the full everywhere else. After summarizing, the difference between the topics within the model improves and shows no significant change from the baseline full model.

With this analysis, we are given evidence that the topics generated by our models are decorrelated enough to have good topic difference. In other words, both of our models have been shown to preform well in terms of topic separation and are good candidates to move on to the next step of comparison.

Now we will visualize the differences between the full and summarized models, showing how summarization can change topic distribution. As we can see from Figure 3, topics between the two models are very similar to each other. Additionally, the non-diagonal cells are light blue meaning that the topics of the full model are not strongly decorrelated to the topics of the summarized model. Intuitively this makes sense because the summarization has the same important content as the full model. So, when preforming LDA, it will pick up similar topics and topic distributions. These two models show high topic similarity to each other, and may indicate that there is little difference between the two. We will now examine other metrics to evaluate our model including log-likelihood, perplexity, and coherence score.

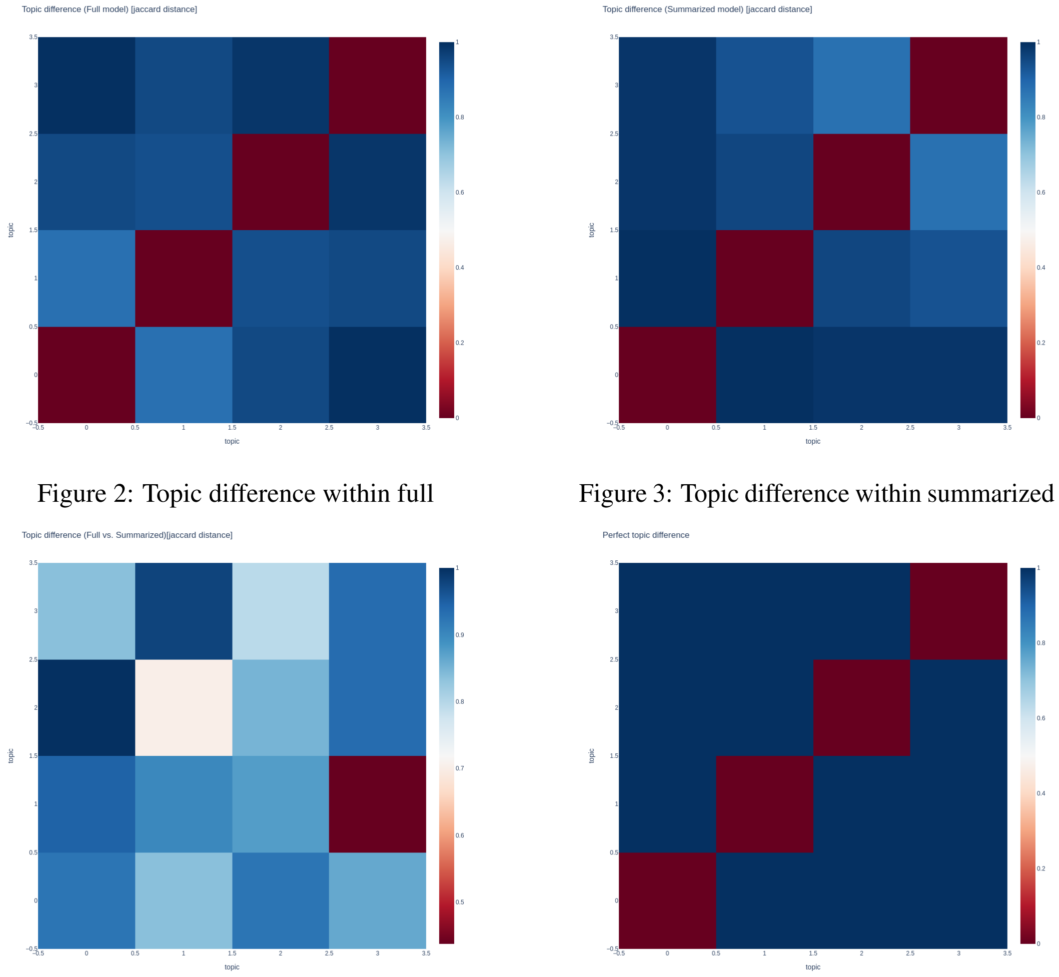


Figure 2: Topic difference within full

Figure 3: Topic difference within summarized

Figure 4: Between Models topic difference

Figure 5: Ideal Topic Difference Plot

Figure 6: Topic Differences for our experiment

4.1.4 Visualization of model structure

Figures 7-10 show the Log-Likelihood and Perplexity Score of the Full (blue) and Summarized (red) models. Figures 11-15 show the generated Word Clouds and Word Counts for the full and summarized models.

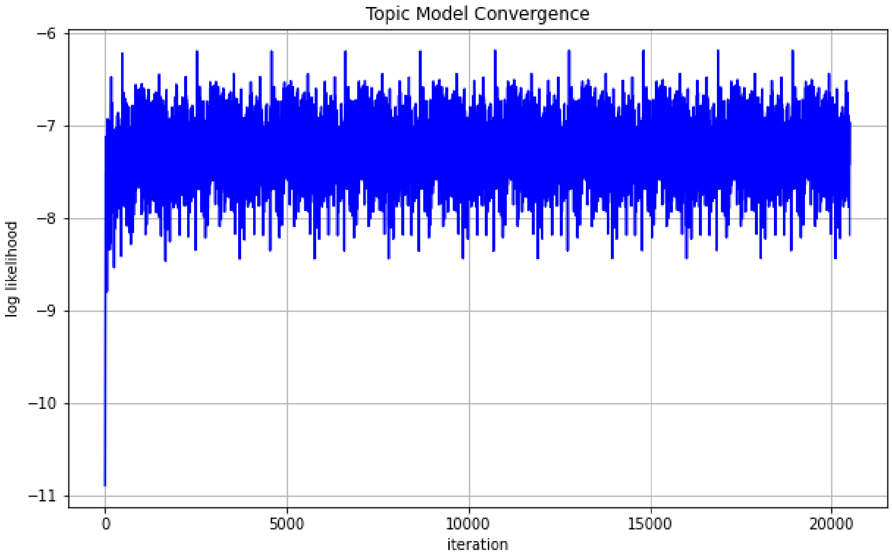


Figure 7: Log likelihood vs Iterations for Full Reviews

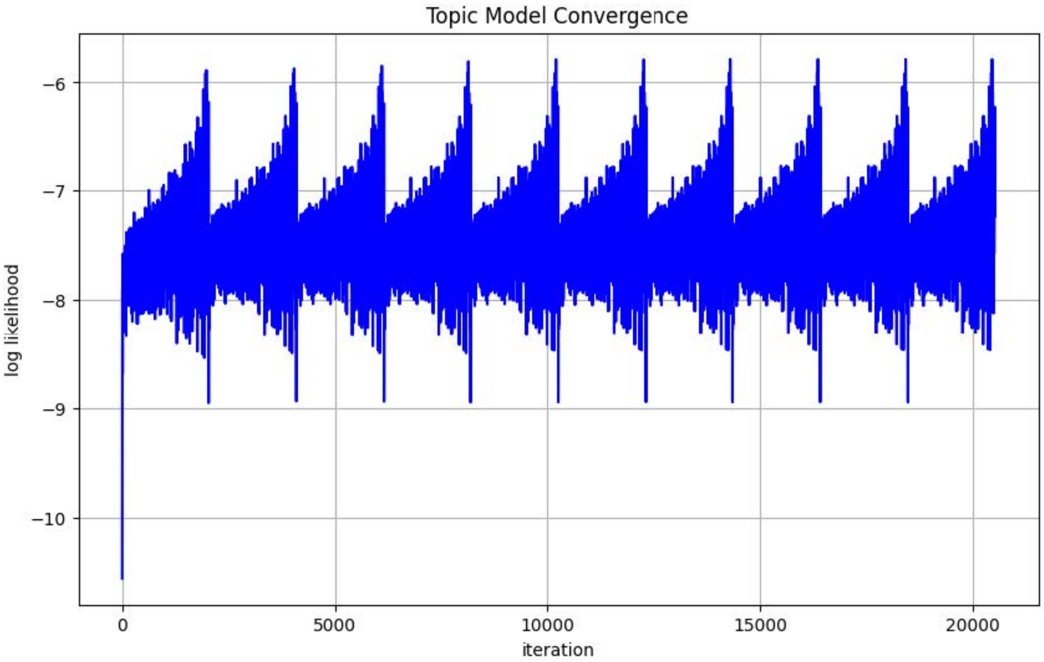


Figure 8: Log Likelihood vs Iterations for Summarized Reviews

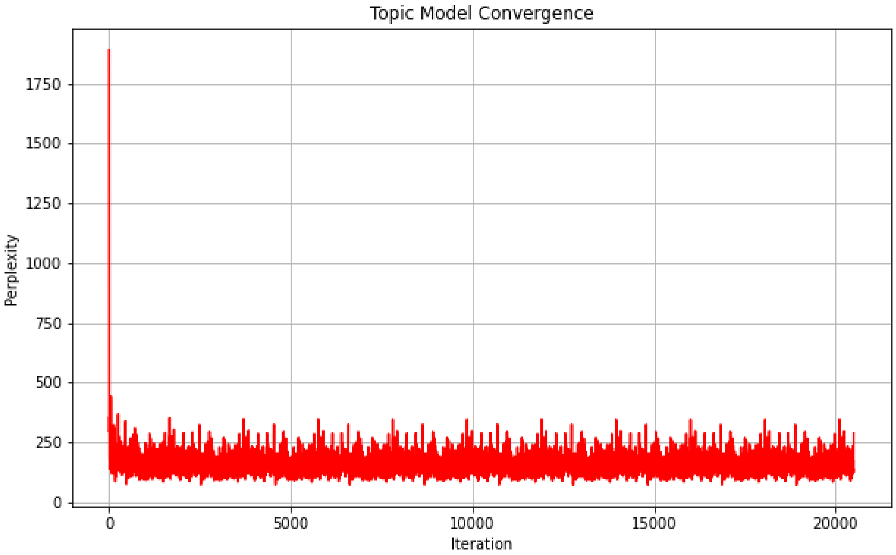


Figure 9: Perplexity vs Iterations for Full Reviews

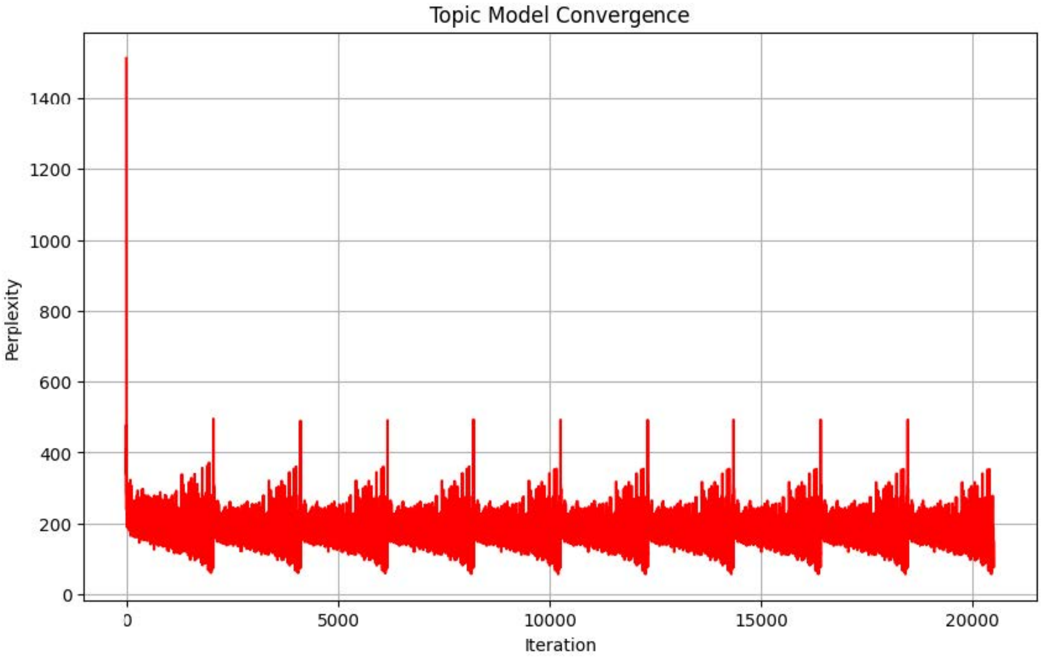


Figure 10: Perplexity vs Iterations for Summarized Reviews

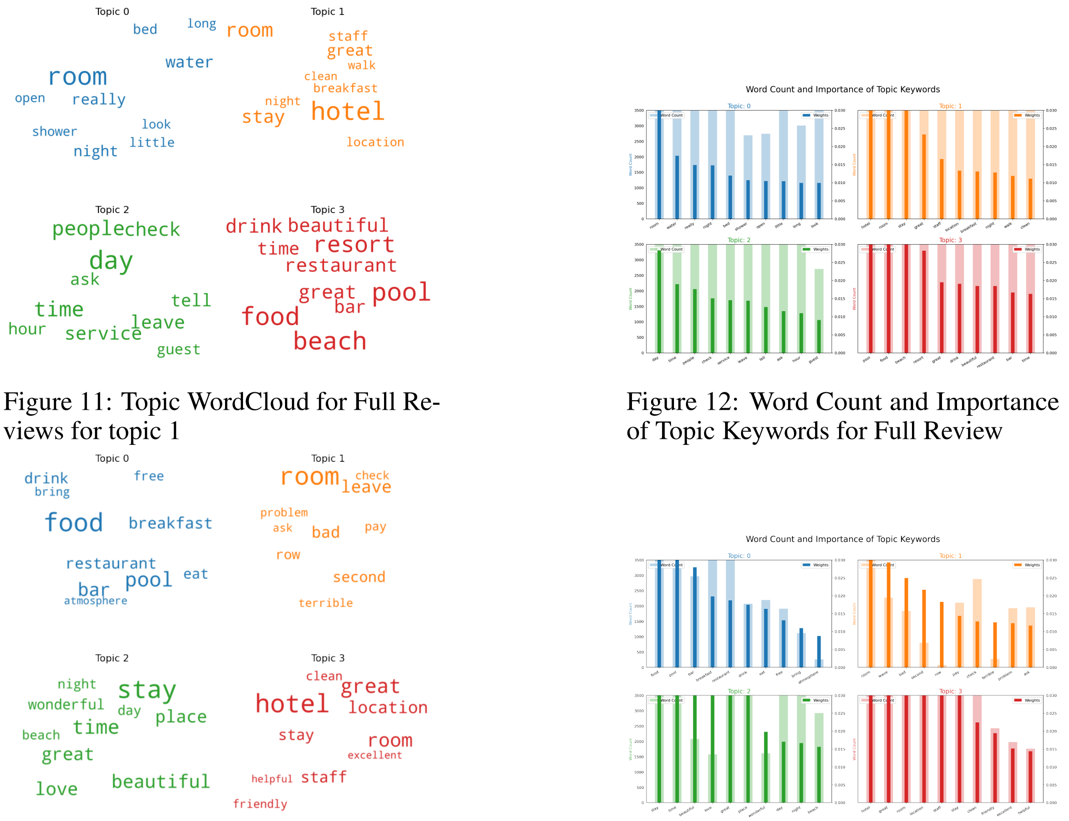


Figure 11: Topic WordCloud for Full Reviews for topic 1

Figure 12: Word Count and Importance of Topic Keywords for Full Review

Figure 13: Topic WorldCloud for Summarized Reviews for topic 1

Figure 14: Topic Word Count for Summarized Reviews

Figure 15: Learning Objective Visualization

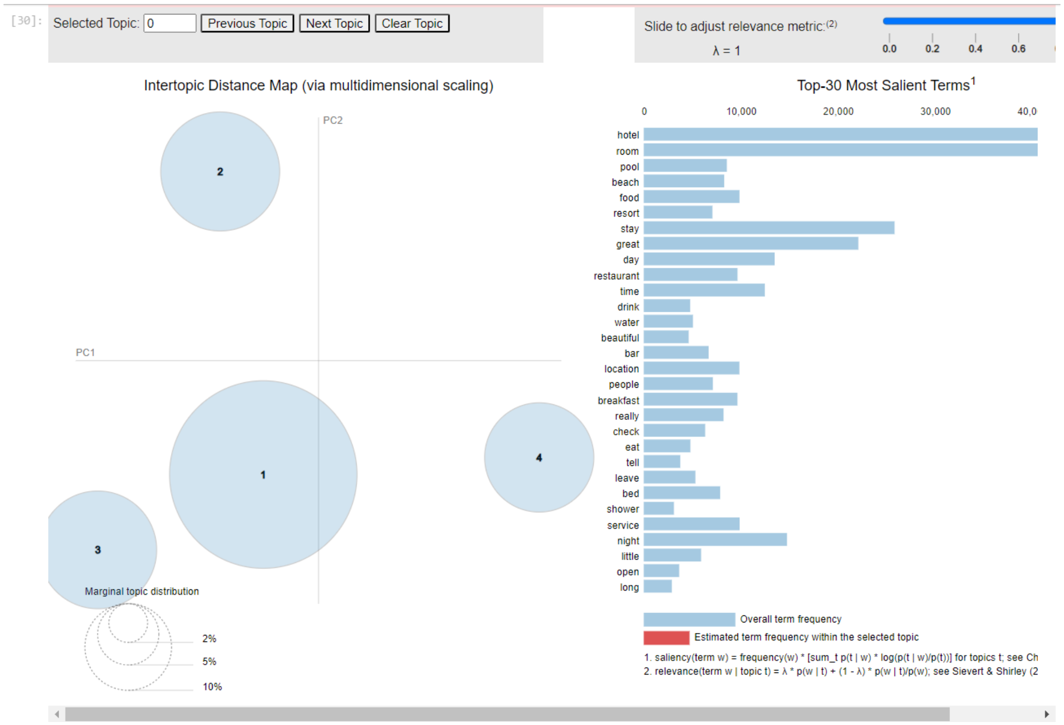
4.2 Software and tools

1. Python
2. Gensim
3. Natural Language Toolkit
4. Spacy
5. Pandas
6. WordCloud
7. pyLDAvis

Adapted code from "Topic modelling(LDA)on Trip advisor dataset" Kaggle notebook [5], Gensim article on "How to Compare LDA Models" [6], and "Building a text summarization model" [9].

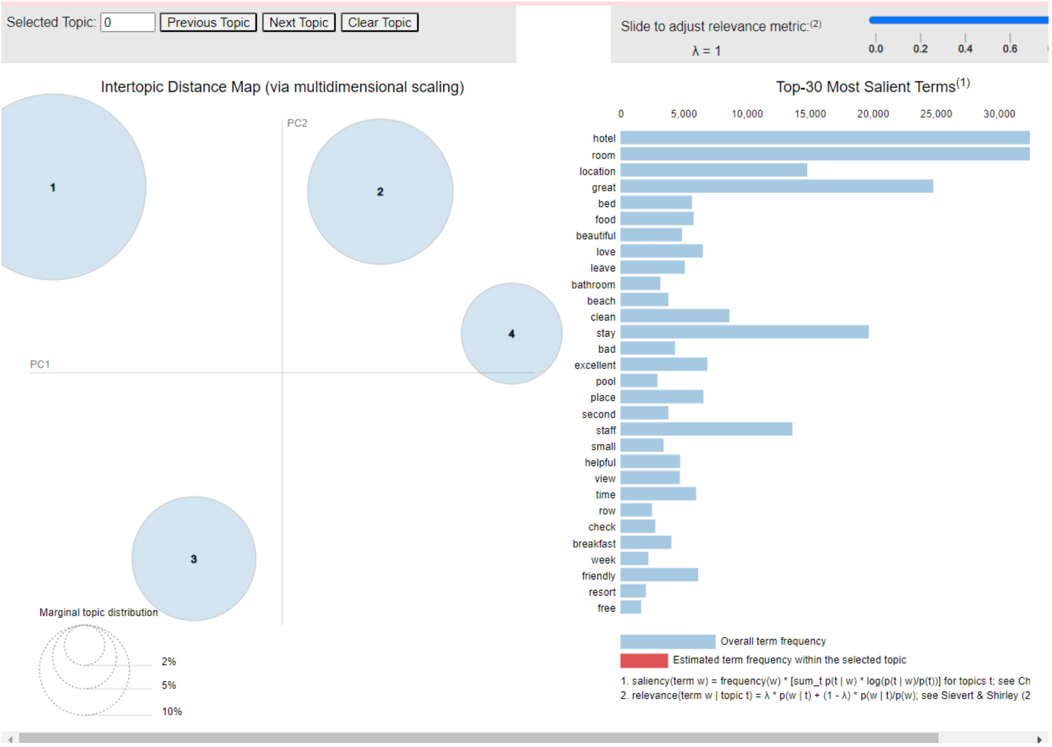
5 Results and discussion

We ultimately found that summarization did significantly impact the topic models generated for the TripAdvisor dataset. It is quantitatively apparent from the topic difference plots that only one topic was found by both models, all others except one were found to have no clear relation to any topic from the other model. Additionally, looking deeper at the actual words found within each topic reveals that once again, only one pair of topics across the two models share meaningful similarities (red word cloud in Figure 12 and green word cloud in Figure 14). It is likely that the fact that the data was not in plain english affected how accurate the summaries were, so future work on this problem may require better suited data and more sophisticated summarization techniques.



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Ch
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2

Figure 16: Inter-topic Distance Map for Full Reviews



1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Ch
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2

Figure 17: Inter-topic Distance Map for Summarized Reviews

6 References and contributions

6.1 References

[1] Röder, Michael, Andreas Both, and Alexander Hinneburg. "Exploring the space of topic coherence measures." Proceedings of the eighth ACM international conference on Web search and data mining. 2015.

[2] Chen, Si and Wang, Yufei. "Latent Dirichlet Allocation". Department of Electrical and Computer Engineering, University of California, San Diego. Retrieved June 12, 2024, from https://acsweb.ucsd.edu/ yuw176/report/lda.pdf.

[3] Blei, David M. (2012, June 2). "Probabalistic Topic Models". Department of Computer Science, Princeton University. http://www.cs.columbia.edu/ blei/talks/Blei_ICML_2012.pdf

[4] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation." Journal of Machine Learning Research 3 (2003): 993-1022.

[5] User imnoob. "Topic Modelling LDA on Trip Advisor Dataset." Kaggle Notebook. Accessed 06/04/24.

[6] Rehurek, Radim. "Comparing LDA Models." Gensim Documentation. Accessed 06/04/24. Available at: https://radimrehurek.com/gensim/auto_examples/howtos/run_compare_lda.html

[7] MVD, Andrew. "Trip Advisor Hotel Reviews." Kaggle Dataset. Accessed 06/04/24. Available at: https://www.kaggle.com/datasets/andrewmvd/trip-advisor-hotel-reviews?resource=download

[8] Maier, Daniel, et al. "Applying LDA Topic Modeling in Communication Research: Toward a Valid and Reliable Methodology." Communication Methods and Measures, vol. 12, no. 2-3, 2018, pp. 93-118. Taylor & Francis Online. Accessed 6/14/24. DOI: 10.1080/19312458.2018.1430754.

[9] M., Sandy (2023, May 6). Building a text summarization model. Medium. https://heartbeat.comet.ml/building-a-text-summarization-model-532f4446efc3

[10] Mathur, Akash. "Topic Modelling LDA on Trip Advisor Dataset." Kaggle Notebook. Accessed 06/04/24.