Topic Modeling and Visualization

1. Brief Overview of Models and Tools

In general, I used two different topic modeling / extraction methods and two different visualization tools. The results obtained and their comparison are presented below. The first topic model was the Latent Dirichlet Allocation (LDA) which is based on Google's word2vec method as implemented in the Python gensim module by Radim Rehurek [1]. It was already provided in the Python tools for Task 1, but I did additional tuning of parameters to get a better delineation of topics.

The second topic model I used was the Non-Negative Matrix Factorization (NMF) as implemented in the Python scikit-learn module [2]. According to the documentation, NMF has two different objective functions: the Frobenius norm, and the generalized Kullback-Leibler divergence. I used the latter function which is an equivalent of the Probabilistic Latent Semantic Indexing (PLSI or PLSA). Therefore, the two topic models that I used were **LDA and PLSA**.

For visualization, I used **graph-tool** which is an efficient Python module for graph visualization [4]. The resulting tree diagram is somewhat similar to the example shown in the description of Task 1, but I employed other visual effects and colors (not sure what tool was used to create the example, maybe it was a different one). The second visualization technique is called "**word clouds**" and is best described by the author of this Python module here [5].

For Task 1.1, I analyzed all the Yelp reviews, and I compared a subset of positive reviews (4 and 5 stars) with a subset of negative reviews (1 and 2 stars) for Task 1.2. I extracted topics using the provided Python tool (py27_processYelpRestaurants.py), but I implemented the following changes to achieve my own, additional data transformation. First of all, I increased the number of restaurant review samples from 100,000 to 150,000 to get a more diverse representation. Secondly, I ran the processing file twice to get 20 different cuisines instead of just 10. I could have extracted even more cuisines, but in the interest of time I believed that this number was sufficient.

In order to extract positive or negative reviews, I inserted *if* statements in approx. lines 110 and 114 of the provided Python tool to check if *review_json ['stars']* had the required values. Then, I also extracted 20 sample cuisines and set the number of sample restaurant reviews to 150,000 (although, in practice it could be less because of a limited total number of reviews for a particular sample cuisine).

When running the LDA model, I saw a warning from the logger that there were too few updates in the model, so I increased the number of passes to 10 and iterations to 100 (defaults are 1 and 50). I also tried to play with different parameters, such as the number of topics and features. After reviewing the results of many different combinations of these parameters, I came to a conclusion that the less the number of topics and the number of features are, the more clear delineation between the topics is. Therefore, I usually used 10 to 20 topics, which approximately corresponds to the number of sampled cuisines, and 10,000 features (the defaults were 100 and 50,000). Probably, this can also depend on how many reviews each cuisine contains because the difference in the size of the review sample file for different combinations of cuisines can measure in dozens of Mb which means that some cuisines have a lot more reviews than others.

In the NMF-based PLSA model, there is no parameter for passes, but I increased the number of iterations to 400 (default - 200). The same as with LDA, I used 10-20 topics and 10,000 features. The number of top words shown for each topic was 15, which was used for the word cloud visualization.

The graph-tool tree diagram used the first 10 of these because 15 words per topic would make it too congested for an effective analysis.

The extra effort that I did to make the visualization more beautifully designed was adjusting the color map for the background, topic words, and other additional visualization features in order to make the diagrams more readable and, thus, more efficient.

2. Visualization Results for Task 1.1 (All Reviews)

Below I included LDA vs. PLSA diagrams prepared with the use of each visualization tool and a brief analysis of the results. 10 terms per topic were used for the tree diagram and 15 for the word cloud.

2.1 LDA vs. PLSA Visualized as Tree Diagrams

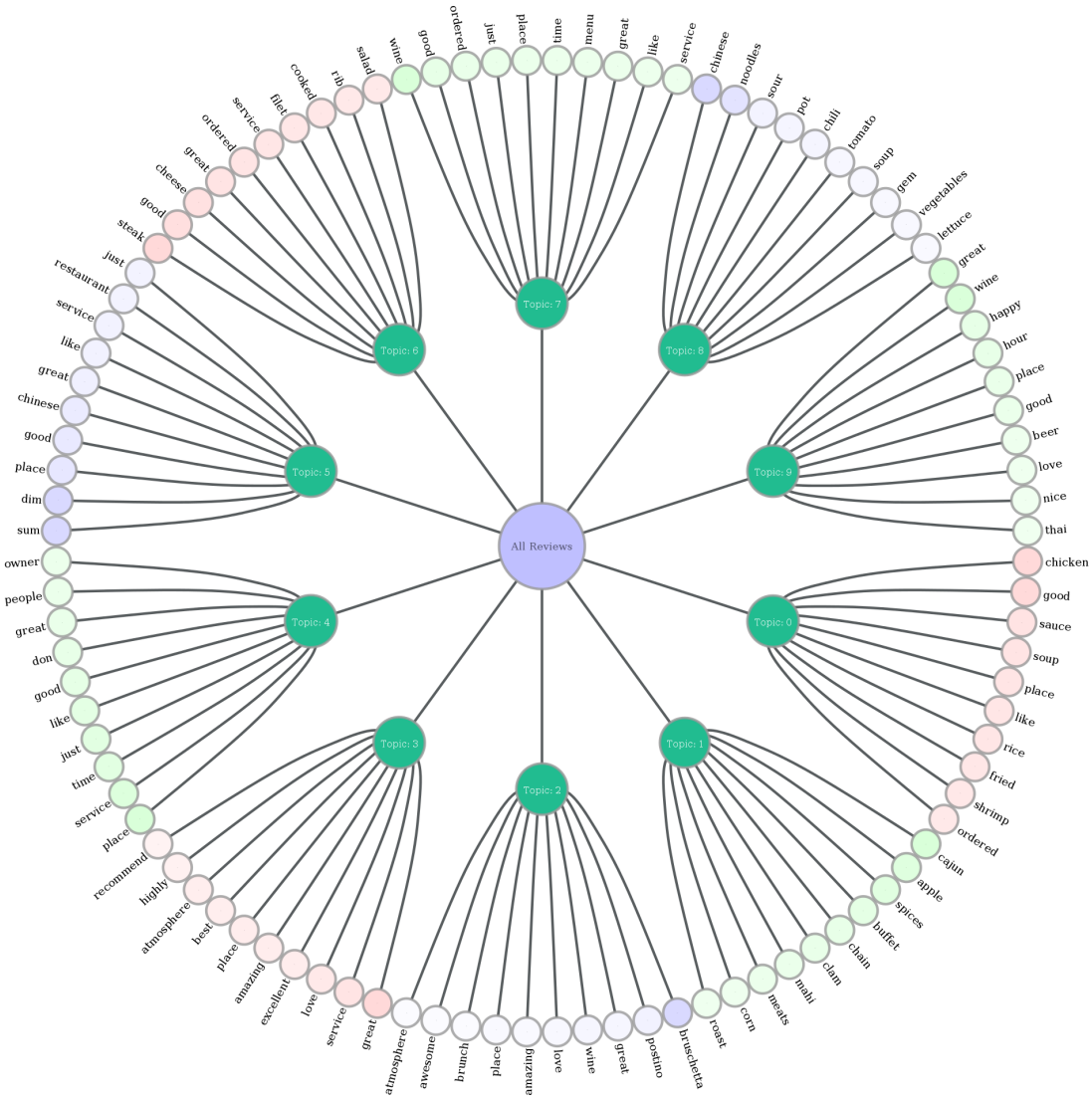

Figure 1. LDA Model Tree Diagram. All Reviews

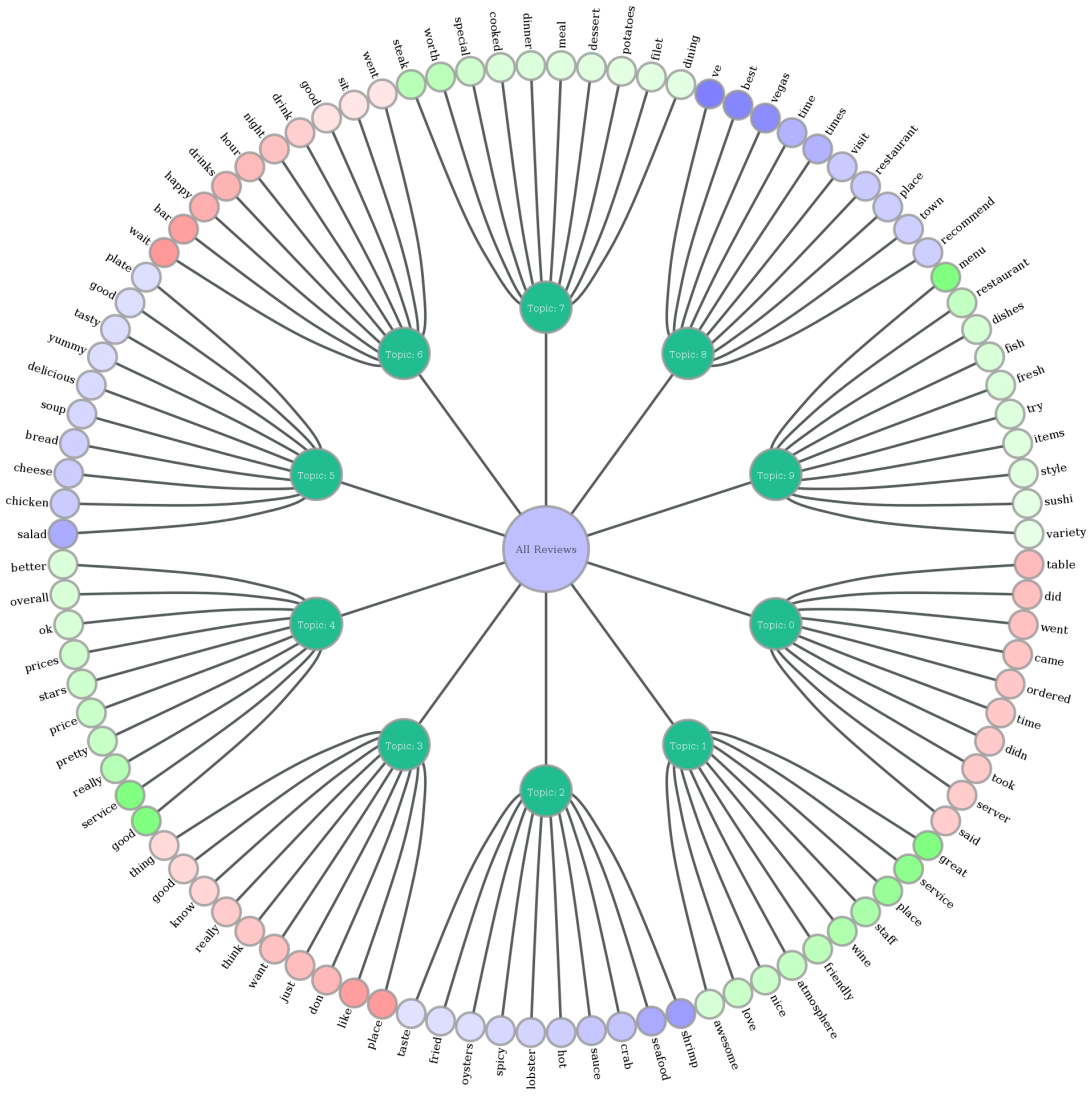

Figure 2. PLSA Model Tree Diagram. All Reviews

2.2 LDA vs. PLSA Visualized as Wordclouds

Figure 3. LDA (left) vs. PLSA (right) Wordclouds

2.3 Conclusion

In my opinion, LDA did a pretty good job of grouping words into topics. For instance, it managed to capture a sandwich place (topic 1), a meat place (topic 7), a Chinese restaurant (topic 9), a dim sum restaurant (topic 6), a wine bar (topic 3), excellence of service (topic 4), a bar in general (topics 8 and 10), etc.

PLSA, on the contrary, is somewhat weaker and has too many topics formed based on the general terms describing how good the service was (topics 1, 2, 4, and 5). But it also managed to group together terms representing various types of restaurants, such as seafood (topic 3 - pretty good!), steak house (topic 8), general bar/restaurant dishes (topic 6), sushi restaurant (topic 10), a restaurant in Las Vegas (topic 9), and a bar in general (topic 9). I also noticed that LDA is more flexible from run to run, while PLSA tends to provide almost the same exact results each time.

More work may need to be done in order to get more precise results. For example, the above was achieved without word normalization (singular vs. plural), lemmization, or POS tagging. It also seems that wordclouds are more convenient to compare side by side than tree diagrams.

3. Visualization results for Task 1.2 (Positive vs. Negative reviews)

Below I included LDA vs. PLSA diagrams prepared with the use of each visualization tool for two cases: positive and negative reviews (4-5 stars and 1-2 stars, respectively). I also included a brief analysis of the results. 10 terms per topic were used for the tree diagram and 15 for the word cloud.

3.1 LDA vs. PLSA Visualized as Tree Diagram

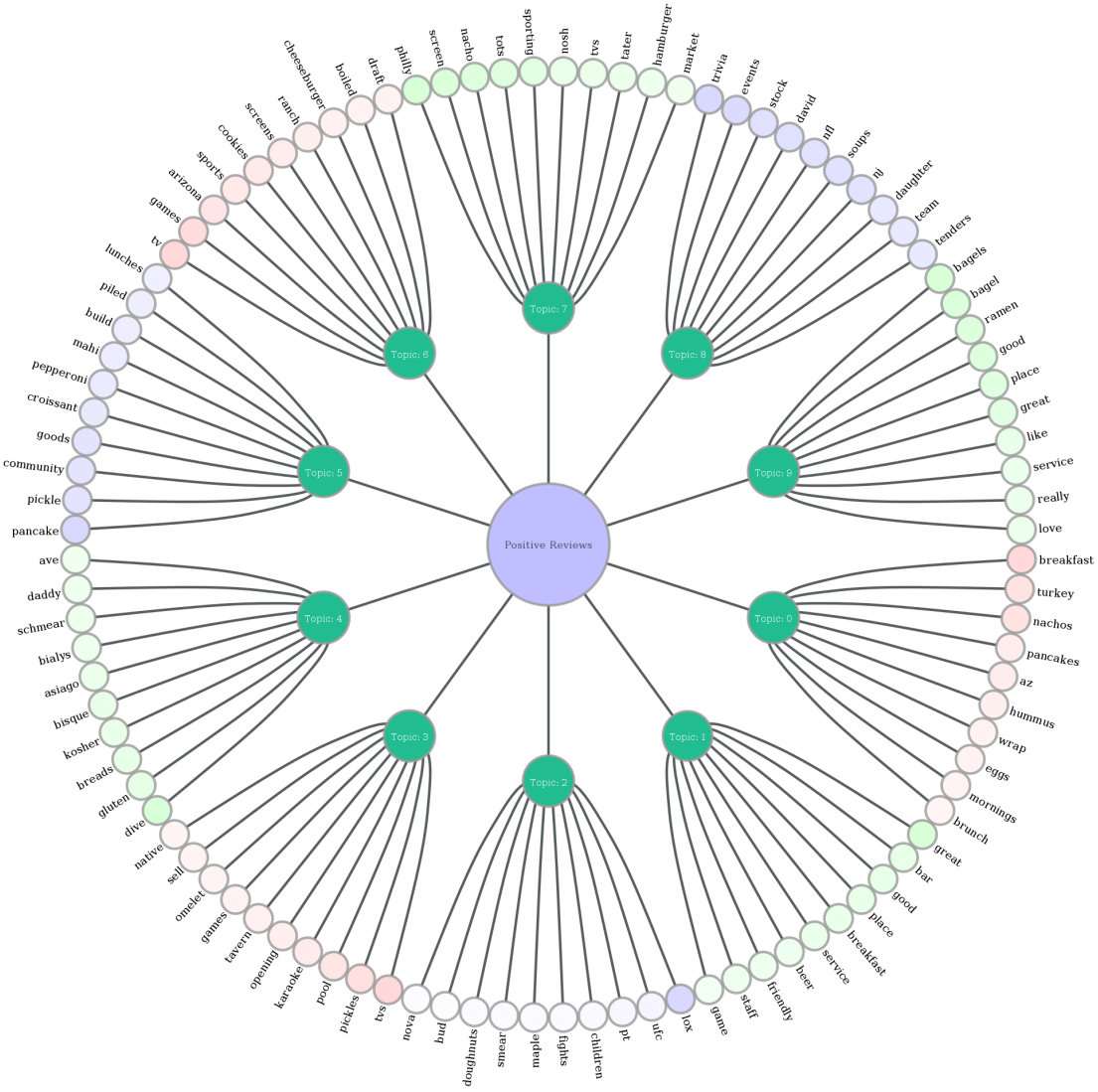

Figure 4. LDA Model Tree Diagram. Positive Reviews

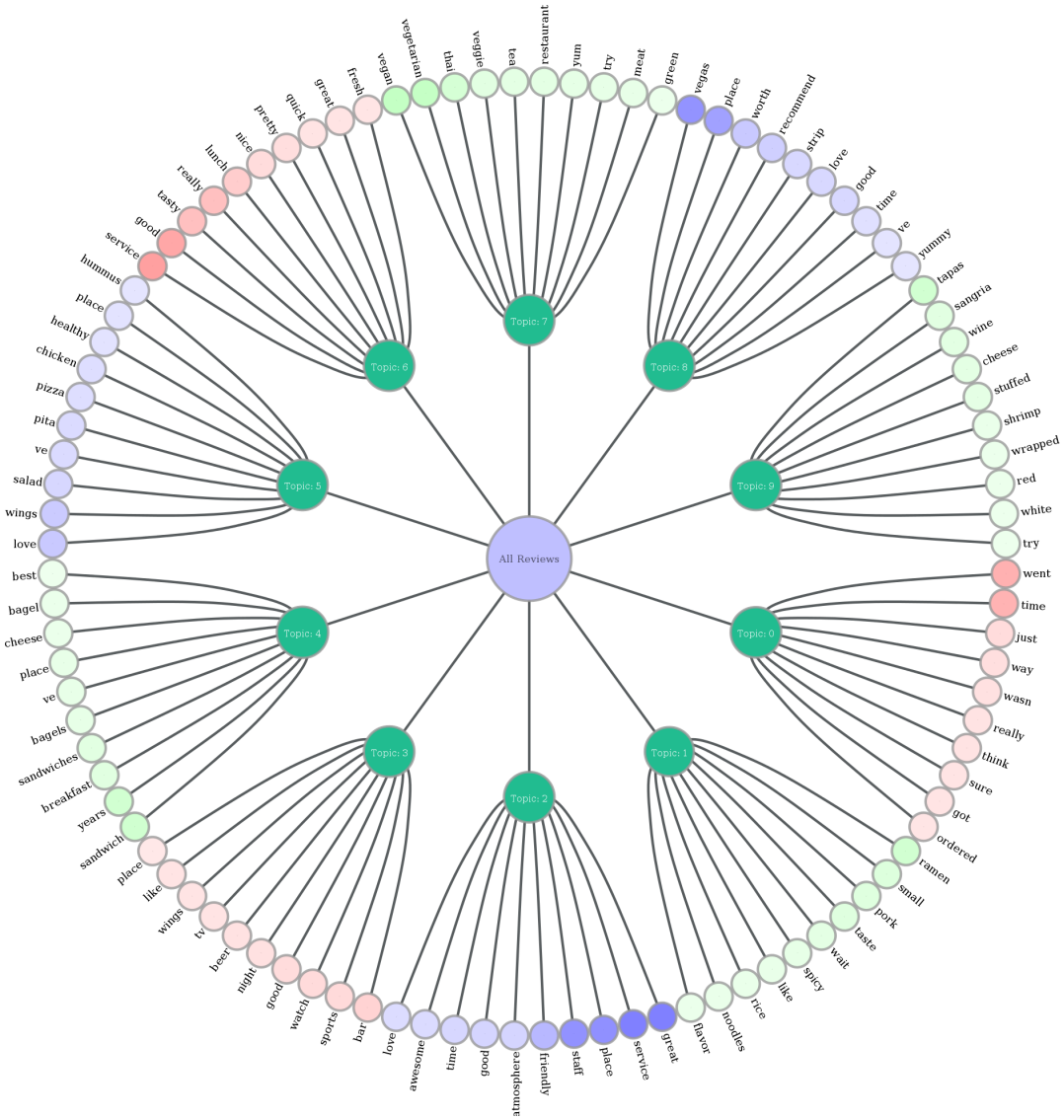

Figure 5. PLSA Model Tree Diagram. Positive Reviews

Note: please disregard "All Reviews" in the root node. These are positive reviews.

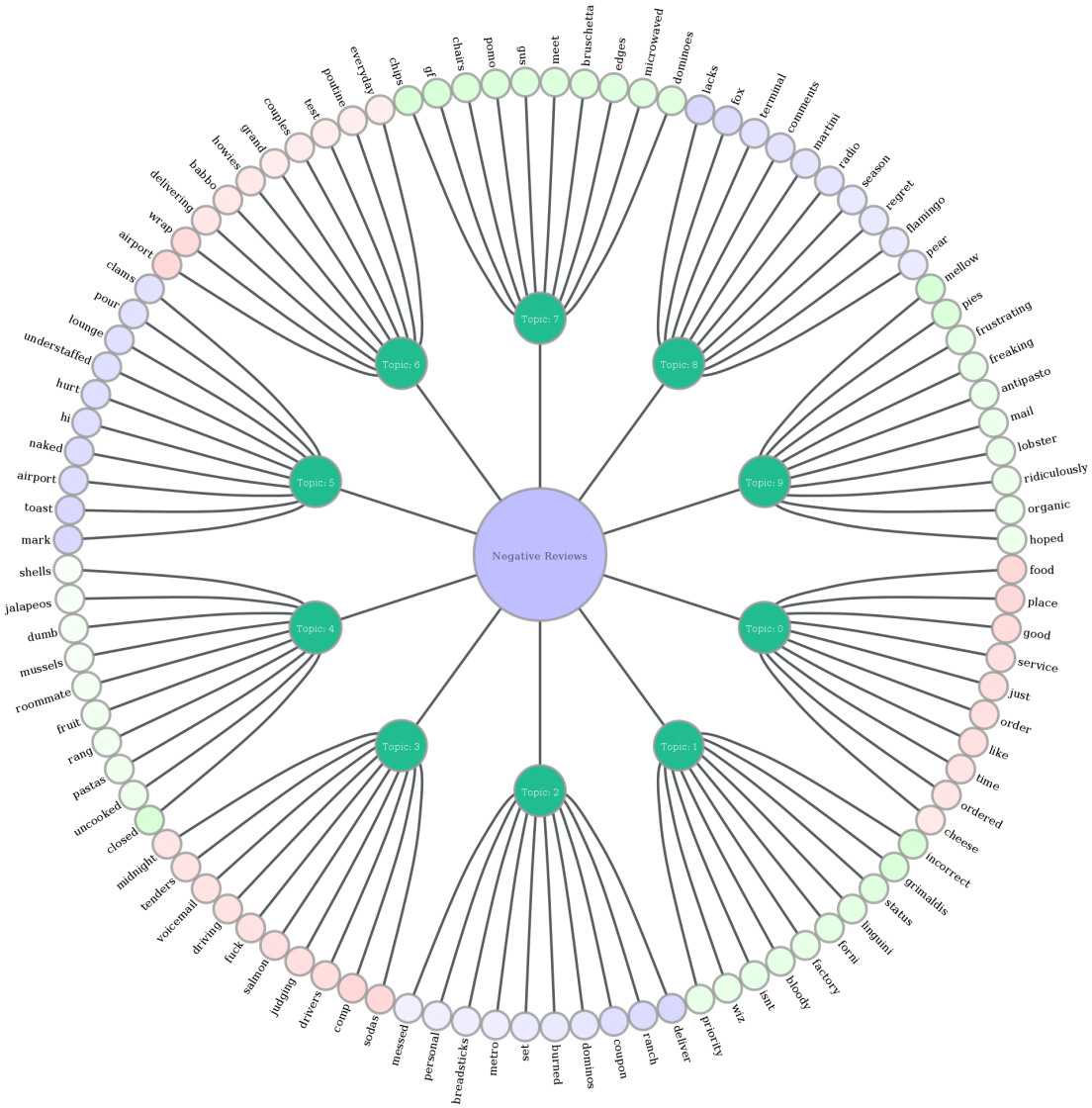

Figure 6. LDA Model Tree Diagram. Negative Reviews

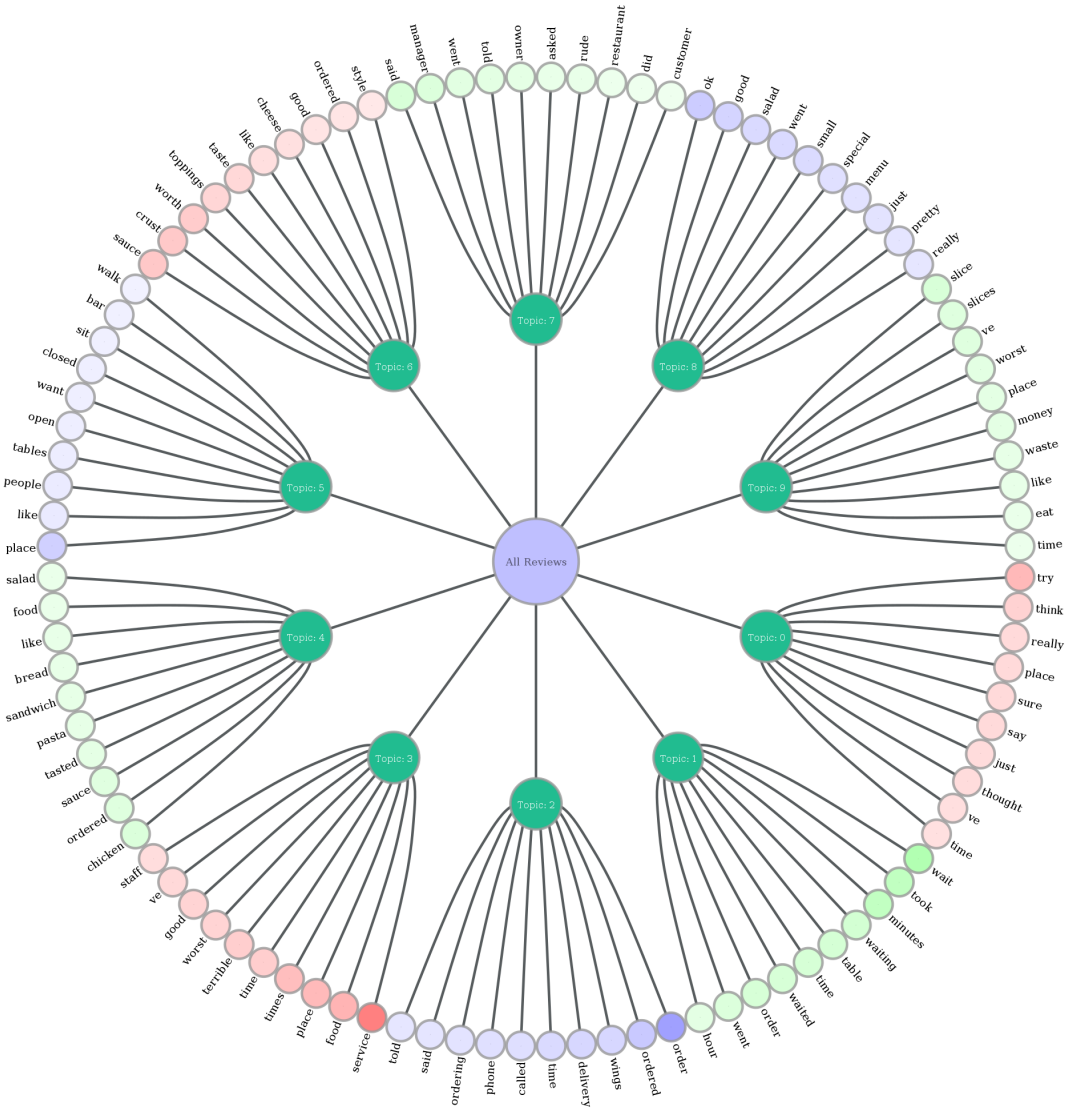

Figure 7. PLSA Model Tree Diagram. Negative Reviews

Note: please disregard "All Reviews" in the root node. These are negative reviews.

3.2 LDA vs. PLSA Visualized as Wordcloud

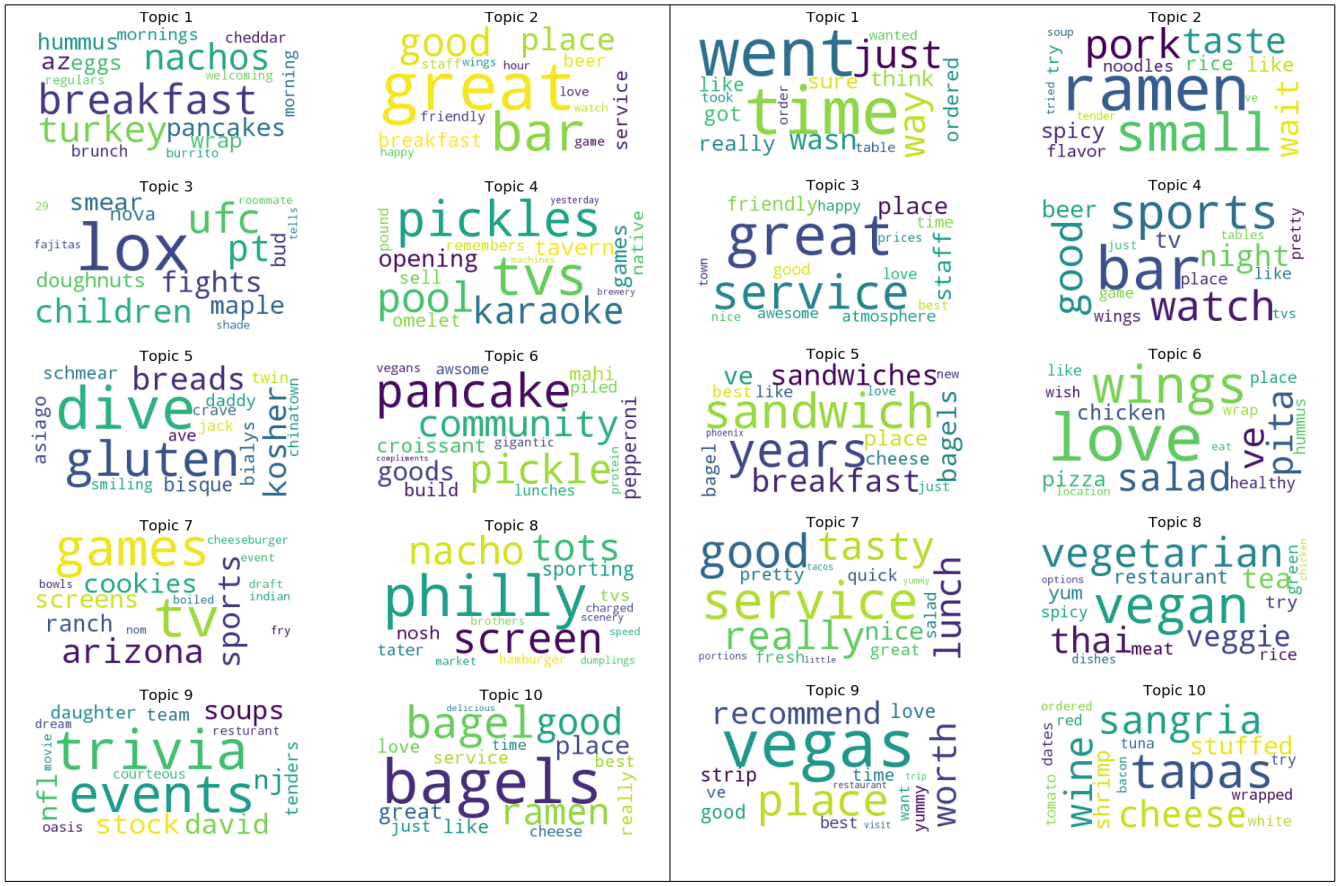

Figure 8. LDA (left) vs. PLSA (right) Wordclouds. Positive Reviews

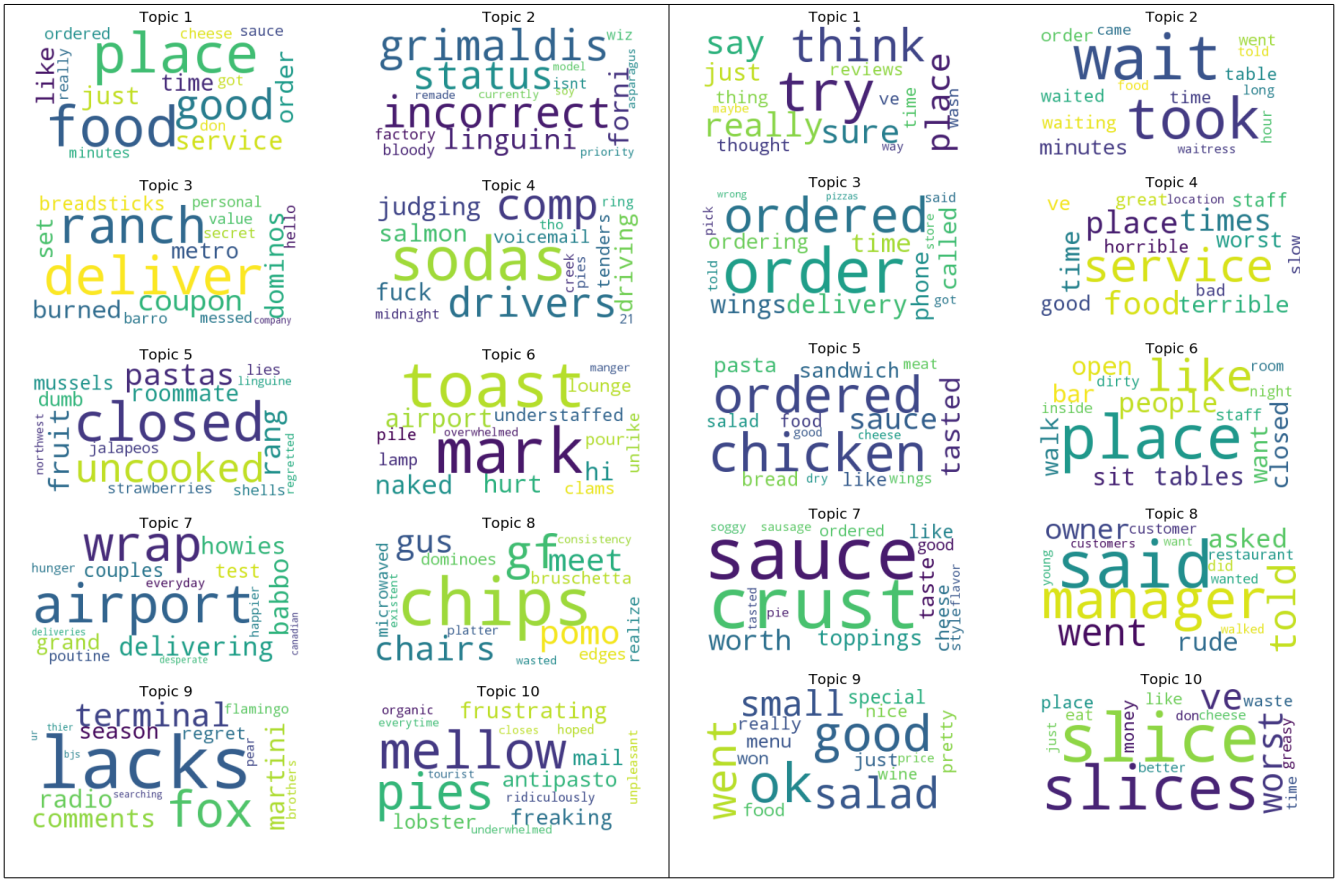

Figure 9. LDA (left) vs. PLSA (right) Wordclouds. Negative Reviews

3.3 Conclusion

In the negative results people still talk about dishes and different kinds of food, but they add adjectives and other attribute words (even obscene ones as seen from the LDA topic 4) to describe the poor service or low food quality.

4 Comparison of Visualization Techniques

It is clear that wordclouds are a better visualization technique if you want to compare different cases. Tree diagrams may be a fancy way to visualize things that looks more beautiful, but wordclouds are just more convenient and practical. Another difference between them: it is possible to include more than 10 words into a worldcloud without making it congested (up to 100 words) compared to a tree diagram which may become hardly legible if 15 or more words are included per each of the 10 topics.

5. References

1. Gensim LDA model description: https://radimrehurek.com/gensim/models/ldamodel.html
2. Sklearn NMF PLSA model description: http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html
3. Topic extraction with NMF and LDA: http://scikit-learn.org/stable/auto_examples/applications/plot_topics_extraction_with_nmf_lda.html#sphx-glr-auto-examples-applications-plot-topics-extraction-with-nmf-lda-py
4. Graph tools: https://graph-tool.skewed.de
5. Wordcloud visualization: https://github.com/amueller/word_cloud