CS 598: Data Mining Capstone - Task 1

Xiaoming Ji

Software Environment

- OS: Windows 10
- Python: 3.6.8
  - gensim: 3.4.0
  - nltk: 3.4.1
- Visualization
  - d3: v4
  - d3-hierarchy : v1

Data Pre-processing

I performed the following steps:

- Tokenization: Split the text into sentences and the sentences into words. Lowercase the words and remove punctuation. In addition, filter out tokens that appear in,
  - Less than 15 documents.
  - More than 50% documents.
  - Keep only the first 100000 most frequent tokens.
- Words that have fewer than 3 characters are removed.
- All stopwords are removed.
- Words are lemmatized. Words in third person are changed to first person and verbs in past and future tenses are changed into present.
- Words are stemmed. Words are reduced to their root form.

LDA Topic Analysis

I tried both Metapy and gensim package for LDA topic analysis and finally decided to use gensim for this assignment due to its performance (running speed). The following parameters are used to build LDA model.

```
models.LdaModel(...,  eval_every=5, iterations = 1000, alpha='auto', gamma_threshold=0.01)
```

In addition, TfidfModel is used to transform review corpus. My experiment shows TF-IDF gives better results compared with bow corpus.

Visualization

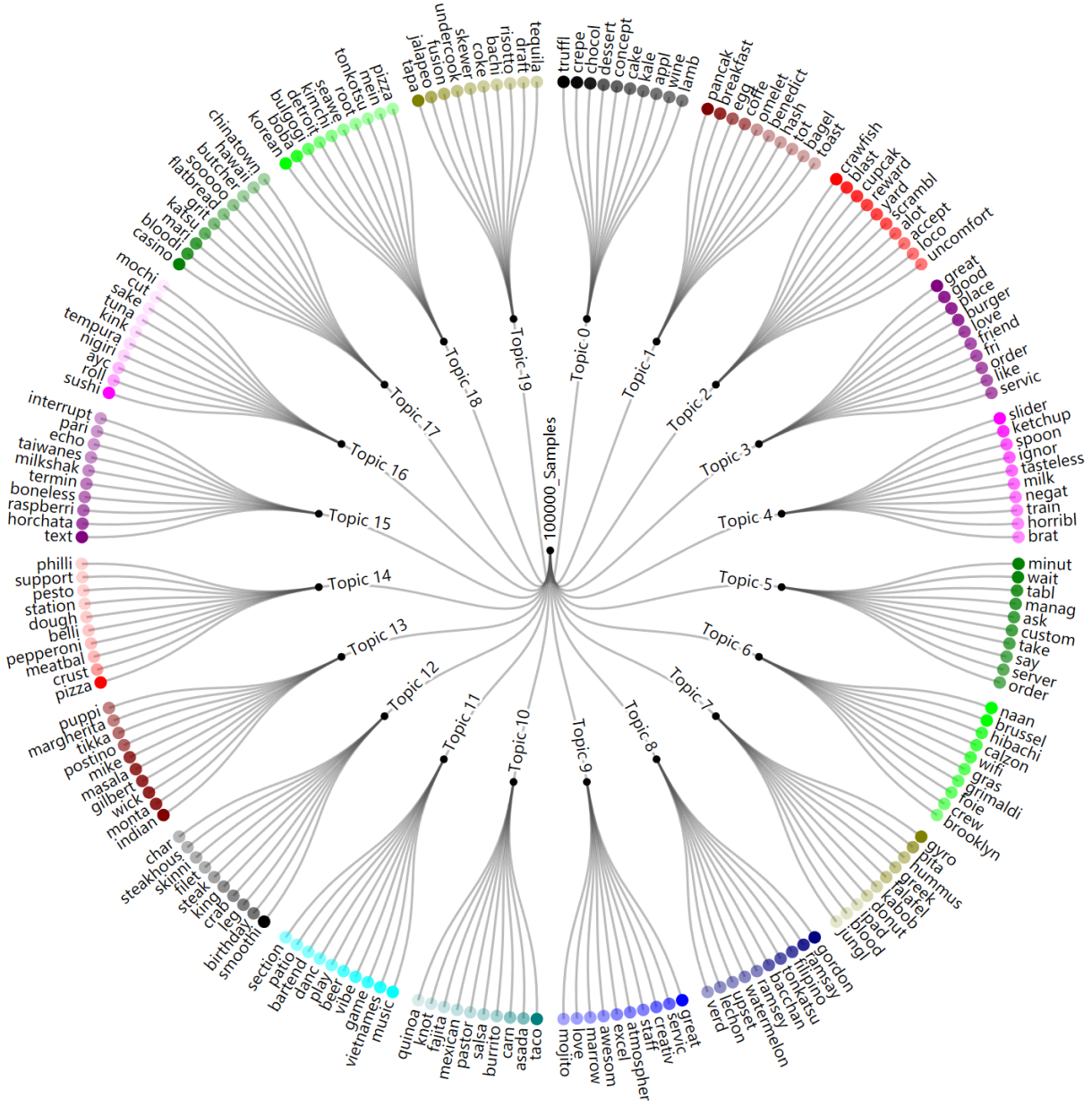
D3 is used to build radial dendrogram to demonstrate topic/words distribution. Different color is used for different topic. The opacity of a word node in a topic corresponds to its relative weight in such topic using the following formula:

```
fill-opacity = word_weight / max_word_weight_of_this_topic
```

Task 1.1

Use a LDA topic model to extract topics from all the review text (or a large sample of them) and visualize the topics to understand what people have talked about in these reviews.

Instead of extracting sample reviews using `py27_processYelpRestaurants.py` which is biased on small portion of randomly selected cuisines, I randomly select 100,000 samples from all restaurant reviews. This approach gives me a more general topics on all reviews. My experiment shows small number of topics won't give much meaning on each topic, thus I choose 20 topics for this analysis.



Topic Analysis

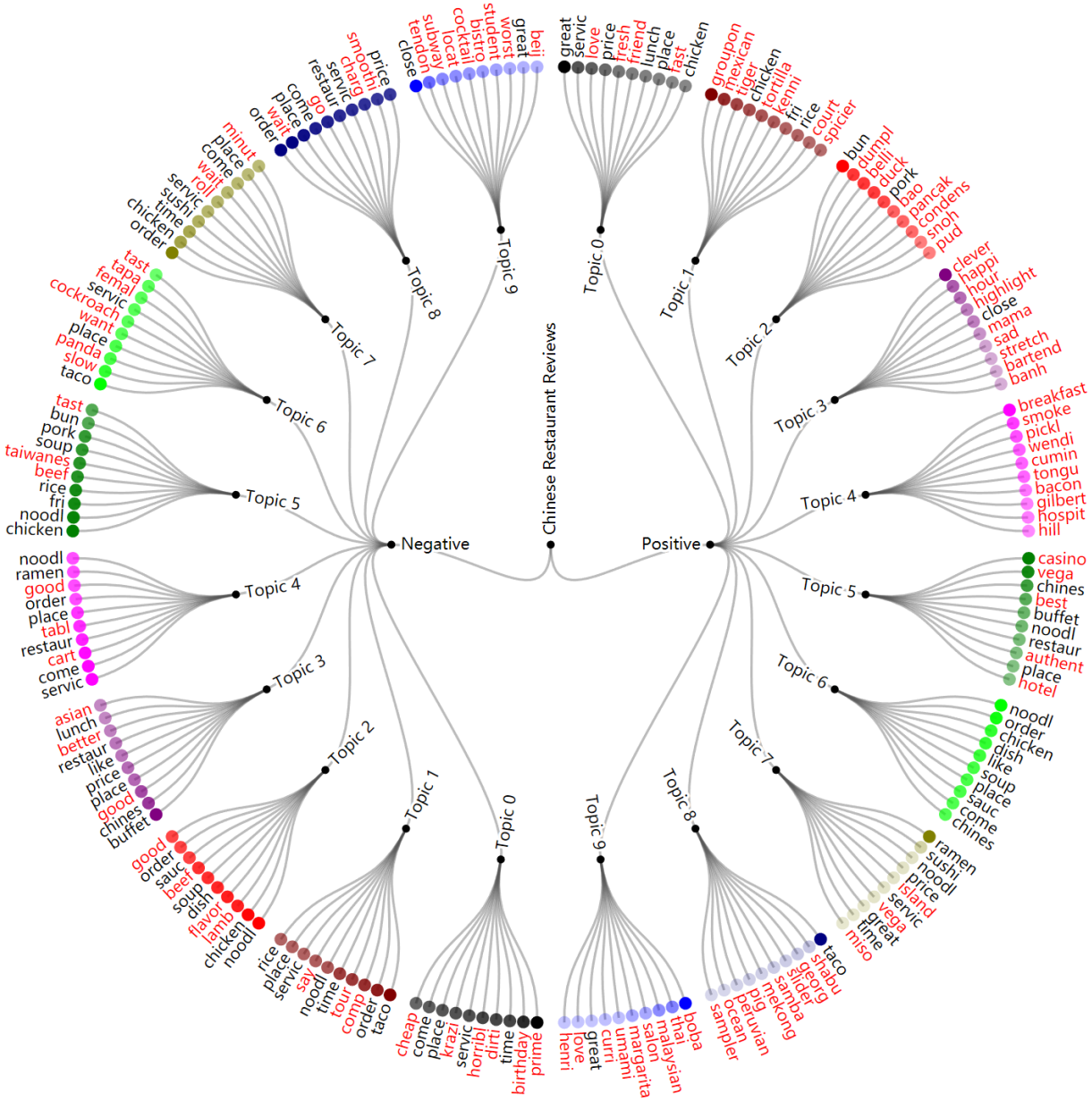
From the words distribution of these topics, we can clear see some relevant info in most topics. For example:

- Topic 1 talks about breakfast (pancake, breakfast, egg, coffe etc).
- Topic 10 talks about Mexican food (taco, burrito, salsa).
- Topic 11 talks about facilities (music, game, danc, bartend)
- Topic 16 talks about Japanese food (sushi, roll, tempura etc).

Task 1.2

Do the same for two subsets of reviews that are interesting to compare (e.g., positive vs. negative reviews for a particular cuisine or restaurant), and visually compare the topics extracted from the two subsets to help understand the similarity and differences between these topics extracted from the two subsets.

For comparison, I extracted topics for all negative reviews (stars < 3) and positive reviews (stars > 3) of `Chinese` cuisine.



Topic Analysis

- Top-10 words that only exist in one model is highlighted in red.
- From the diagram, we can see both topic models share a lot of common words, for example: `buffet`,

`noodl`, `ramen` and all top-10 words in topic 6 for positive topics exist in negative topics as well. This is not surprising given a lot of words are very relevant to Chinese restaurants.

- However, we do see a lot of top-words only exists in one model. Specifically, `horribl` and `worst` only exist in negative topics. `happi`, `love`, `best` and `hospit(ality)` only exist in positive topics. Such results match the sentiment perspective of these subsets.