Yelp review summarization using LDA and LexRank

Anton Oyung and Andrew Peng

Abstract

In this paper we explore applications of the Natural Language Processing algorithm, Latent Dirichlet Allocation (LDA), on Yelp restaurant reviews. Our goal is to find a practical and useful function for this algorithm by utilizing the subtopic output to create a general synopsis of all text reviews per restaurant. We do so by partitioning subtopics on rating and analyzing the rating distribution. By aggregating the various subtopics together we provide the user with a comprehensive unbiased assessment of a restaurant in one convenient paragraph.

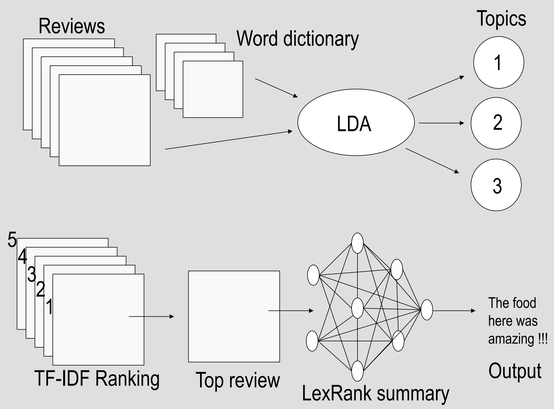
Introduction

When deciding where to eat, Yelp is hands-down the best place to find great new restaurants to try. However when it comes to actually choosing one of these highly coveted restaurants, Yelp falls short. Users must oversimplify restaurants by reducing them to star ratings or scour through mountains of user reviews of varying quality.

By parsing all of the text reviews for a restaurant and producing a comprehensive and representative summary, we allow users to make an educated decision when selecting what restaurant to dine at. We select fragments of user reviews to ensure authenticity. Additionally, we amass the reviews proportionally to the relevance of the subject to ensure an accurate representation of the overall consensus on the restaurant.

Implementation

- We used nltk, which is a natural language processing library and gensim, a topic modeling library.
- The system overview for our project is shown below



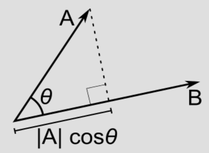
Latent Dirichlet Allocation

- LDA is a statistical model that generates various latent hidden topics that are found in a corpus of documents.
- LDA represents documents as mixtures of topics.
- It assumes each document is created by
  1. Deciding the number of words
  2. Choosing a topic mixture for the document according to the Dirichlet distribution (over K topics)
  3. Assign each word a topic sampled from the topic mixture.
  4. Use the assigned topic to generate the word.

Review Ranking

- After we have our topics (bags of words), we rank each review using tf-idf ranking comparing each review to the top topics.
- Each review is a W length vector, where W is the size of the word dictionary
- The *i*-th entry is the tf-idf value for that word. ($tfidf = tf \cdot idf$)
- Tf = term frequency = $count(word)/num\ words\ in\ doc$
- Idf = inverse document frequency = $\log Number\ of\ documents/Number\ documents\ with\ (word)$
- We create a word vector for our topics as well and take the dot product with each review to find the most relevant using cosine similarity

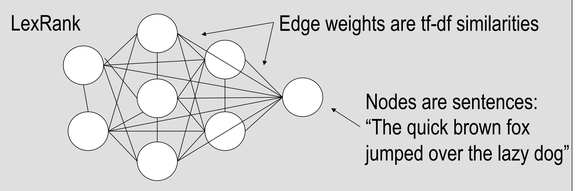
$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



Summarization

- To create summaries of the best reviews, we used LexRank. The LexRank method represents each document as a graph with sentences as nodes and edges representing similarity again with tf-idf.

Summarization



- The PageRank algorithm is then applied to the resulting similarity matrix to rank the sentences. We chose the top 2 sentences of each review.

Results

Restaurant: Mon Ami Gabi
$$ - French, Steakhouse, Breakfast & Brunch

The Good:
"For me, the number one draw was the View not the view of the Paris Hotel & Casino, but the view from the outdoor patio of the Bellagio Fountains across the Las Vegas Strip. There are two dining options at Mon Ami Gabi: Indoor or Patio. We were in no rush to leave thanks to the Fountains of Bellagio view and relaxed pace of our meal."

The Bad: "A food runner brought water and bread fairly quickly. Our server made an appearance about 10 minutes after we were seated. I thought maybe the food would make up for the lack of good customer service here, but it didn't. We were here no later than 10:55 and I called ahead of time but the fact that I had to wait in that ridiculous line to check in so I was 30 minutes "late"."

Restaurant: Secret Pizza
$ - Pizza

The Good: "If you can find this place, you win! A friend recommended this place to me and I had to ask a cosmo worker where it was. After going down the hall though, there was this cool pizza joint. Ordered a couple slices and it tasted AWESOME after a night of Vegas. True NY Style pizza at its best!"

The Bad: "Personally I thought the pizza was alright. I went here twice after going out and each time they only had 2 choices of pizza =/ I kind of wish I made time to check this place out earlier in the evenings so I could have more varieties of pizza to choose from. This is the second time I have gotten sick from eating here! Everyone thinks this is the best pizza because they are so hungry and drunk, they can't tell the difference between crap and good food!"

Issues

- Quality of Reviews: Reviews from people often are full of grammatical and spelling errors that make parsing their text difficult
- Negative review bias: The negative reviews tend to be much longer than other reviews
- Positive review bias: The positive reviews tend to be much shorter and less specific
- Relevance: Not all sentences are relevant or make sense out of context (stories of people's experiences)

Future Steps

- One clear step we need to take is an optimization of our code. By optimizing the runtime we can find more applications for our work.
- Additionally, in order to draw the most relevant sentences we need to write our own ranking system in order prioritize more interesting and useful sentences
- With the addition of new reviews we can also track the change in perception of the restaurant over time. By evaluating our reviews restaurants can make changes and determine if their changes are helping.

Acknowledgements

We would like to thank professor Deborah Nolan for her support in our work and the SCF for providing us with additional computing resources.

References

LDA Python Tutorial: https://rstudio-pubs-static.s3.amazonaws.com/79360_850b2a69980c4488b1db95987a24867a.html

Yelp Academic Dataset: https://www.yelp.com/dataset_challenge

Lexrank.js: http://lexrank.herokuapp.com/index.html

Lexrank: https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume22/erkan04a-html/erkan04a.html