CS181
Ryan Sung
Kevin Cunanan

1. Title of your project

TF-IDF categorization of Yelp Datasets to Infer Categories based off Reviews

2. Problem formulation

Yelp provides a large dataset composed of various types of data ranging from business info to reviews/ratings. We wanted to be able to infer categories based off the reviews using a TF-IDF algorithm to map high frequency words with certain categories from a very large dataset that Yelp provides to users interested in entering their "Yelp Dataset Challenge." Looking at Yelp pages for certain accounts, we as users have noticed there to sometimes be too general of categorization of certain businesses (i.e. a Japanese bento box lunch restaurant being under the broad category of "Asian") or even just plain misrepresented categorization for certain businesses on Yelp – although business owners whose businesses are on Yelp can contact Yelp, claim their business and verify it, and proceed to provide more information on what services they offer as well as hours of operation, etc., the reality is that a lot of business owners might not make the effort to go through that entire process. For that percentage of businesses on the platform, the best way to actually go through and accurately categorize businesses according to what specific services they provide is to look at the reviews that users and consumers leave. In the culmination of all of the reviews, we can find a more accurate and specific categorization of the business, without the owner having to actually change the page themself.

Why is it interesting as a Big Data problem and who would use it if it were solved?
There are two different kinds of Yelp pages on the platform: verified business and non-verified businesses. The former describes businesses in which the owners themselves contacted Yelp and claimed the business as their own, adding legitimacy to the page, as well as giving the business owners the power to add or change information on the page.
Business owners do determine their own categories, but there are many businesses on Yelp that have not been claimed by their owners. Our approach would be valuable for promoting more business through Yelp. But the algorithm has many other applications. Obviously Yelp itself would be using our strategy, however, Yelp wouldn't be the only one benefitting, but also the businesses whose owners did not verify their pages.

3. Your strategy to solve the problem

To solve the problem, we took Professor Salloum's advice and went with the approach of a TF-IDF strategy. TF-IDF stands for *term frequency–inverse document frequency* and is defined as a numerical statistic intended to reflect how *important a word is* to a document in a collection of data. In our case, we defined how "important" a word was in our dataset depending on how often it showed up in reviews in a particular category, but also filtering out the words that also appeared frequently in other reviews within the same category but had no importance to us in terms of narrowing down the type of business (i.e. "the," "and," "but," etc.). The most tedious part was reducing the data to only the essential features that were needed to compute the TF-IDF score. As a result, we had to format and re-parse the data so Spark could take in the data more efficiently.

4. Functions of your software (see part 6 for a more in depth description)

We utilized Python and Spark to format, parse, and analyze the data.

part1: formatted data in more manageable manner

part2: parsed through data and deleted unwanted characters

part3: tf-idf algorithm on data

5. Results and Evaluation

In part3 of our project, we evaluated if the real values matched our expected values, outputting a "percent accurate" number based on the number of matching values divided by total tested. On a small dataset, the prediction accuracy is 100%. But on larger datasets, we would predict that it becomes less accurate.

Unfortunately, we couldn't process the entire dataset because our computers couldn't cache all the data to memory using Spark. As a result, we were only able to test up to 1000 separate reviews (took about twenty minutes to run part3.py). Regardless, we had still 100% accuracy using the first 50 entries of test data. This seems to make sense since the dataset is still relatively small. In other words, there may not be enough terms to give the program a tough decision. Consequently, we need more data before we can do more.

Future implementations of this program would (1) make it much more efficient. Once we accomplish that, we would continue testing the data with larger datasets and checking for accuracy. It'd be interesting to look into which terms are being matched with the category, and seeing if human intuition would matchup with our algorithm. Next, (2) we would sanitize the reviews by spell checking and removing any apostrophes or unnecessary characters. We would ideally get better results if the words didn't have errors and matched appropriately. For style points (3), we'd make all options accessible via the command line using sys arguments. That way we could make values like *minDocFreq* and HashingTF size configurable at runtime.

6. Contributions

Kevin - part1.py

- Spark, Python - Extracted data from JSON files and wrote essential features to file.

Ryan - part2.py

- Python - Clean the data by eliminating extra information written by Spark and convert into a CSV-like file to feed into part3.

Prof. Salloum + Ryan + Kevin - part3.py

- Parse the CSV file from part2.py, create the unique category labels, run the TF-IDF function on our dataset, create and train a model, then test its accuracy.

7. Bibliography

Diane Kim. *5 Things You Absolutely Need to Know as a Business Owner on Yelp*. FiveStars, 2015. Web.

Levene, M. *An Introduction to Search Engines and Web Navigation.*Hoboken, N.J.: Wiley, 2010. Web.

Russell, Matthew A. *Mining the Social Web.*Sebastopol, CA: O'Reilly, 2011. Print.

8. Submission

All documentation for running and code should be available in this GitHub repository.
https://github.com/kcunanan/CS181Fall2016Yelp