Second Capstone Project
Final Report

Problem Statement/Challenge

Given a set of Wikipedia articles, the challenge is to build a model to identify the primary topic of each article and analyze the content and structure of that article. The second part of the project will focus on conducting similar analyses of a group of Wikipedia articles. After topic modeling is complete for the corpus, I will cluster the articles based on their content and recommend similar articles based on those primary topics.

A model such as this will be vital to those interested in clustering, transforming, visualizing, and extracting insights from any unlabeled and unknown dataset. This model would be useful for a number of clients/customers. For example, law enforcement or intelligence services might find this model useful as they investigate documents discovered in a raid or review of forensics information.

Project Details and Description of the Data

For this challenge, I obtained articles from the Wikipedia package. I then accessed and analyzed the various components of the page (i.e., title, URL(s), images, links, summary, and full content). Then, I tokenized each word contained in the article and created a bag of words. To refine the results, I removed non-alphabetic characters as well as English stop words. I then used lemmatization to produce a much better and more concise bag of words based on the stems of the key words in the article. For this particular article--which was "Status of First Nations treaties in British Columbia"--terms such as 'treaty', 'nation', 'process', 'government', etc. were among the primary words/topics returned.

For the next portion of the project, I obtained 100 random articles using the Wikipedia library. Due to the volume of articles, I encountered a number of initial terms/pages that caused disambiguation errors; essentially caused by one term corresponding to multiple Wikipedia articles. The error—the same seen if one were searching for an ambiguous term directly in Wikipedia—provides a list of possible options representing the related articles. Therefore, I had to write code to catch the error/exception, select one of the options, access that page, and continue the loop. I used the core of that code to 1) generate the list of titles and 2) to collect the content for those titles. Then, I tokenized each article in a similar fashion as mentioned above and created a gensim corpus. I then created a gensim bag of words. The gensim bag of words for these 100 random articles revealed that top terms included "first", "also", system", "writing", and "team." I also used Tf-idf to find topics from a particular article with results based on the entire corpus. For my example, I selected the second article that produced the top terms of "goapele" with a score of 0.48 and "dawn" with a score of 0.21.

Finally, I used k-means to cluster the articles. For this model, I chose a k of 5. The majority of the articles fell into clusters 0 and 1. Using these clusters, I could easily recommend similar articles to a reader.

Conclusion

Overall, my initial findings suggested that my model worked quite well for analyzing only one article. The article was processed cleanly and efficiently. Tokenization and the initial bag of words highlighted topics found in the article. I was then able to verify the results by revisiting the title and summary of the article.

I experienced similar positive results for the group of articles. The gensim bag of words and Tf-idf performed quite well and provided a concise list of topics. I was pleased with the results of the k-means clustering and feel this could easily be integrated into an app or website that provides recommendations to users/readers based on their interests.