Searching Through Ted Talks: Data Challenge 5

Jacob Farner
Computer Science, COMP331
Occidental College
Los Angeles, California
jfarner@oxy.edu

Leopold Ringmayr
Computer Science, COMP331
Occidental College
Los Angeles, California
lringmayr@oxy.edu

***Index Terms*—TED Talk, search, search engine, sklearn, NLTK.**

I. Introduction

TED Talks, hosted by TED Conferences LLC, are a well known series of lectures given by leaders and visionaries across nearly any field. With the catchphrase, "Ideas worth spreading," TED Talks are often lauded for the interesting and thought-provoking nature that speakers present to audiences, often in creative manners. We were presented with a database of around 2500 TED Talk written transcripts along with URLs to watch each corresponding video. We were tasked with developing a method of querying the database and returning a given number of relevant TED Talks based on keyword input from the user. In other words, a search engine.

To explore and better understand how this works, we implemented our search engine in python, utilizing the popular Pandas toolkit for data organization, the Natural Language ToolKit (NLTK) for language processing, and SciKit Learn (sklearn) for feature extraction and computing similarities between search terms and transcript.

II. Related Work

Search engines are the foundation of the internet, and there have been countless projects prior to ours that look through databases to provide relevant results based on search terms. Creating better search engines is an active area in academic research, and there are massive markets surrounding search engines. In his book titled *9 Algorithms That Changed The World*, John MacCormick provides a comprehensive history of search engines in chapter 2, *Search Engine Indexing: Finding Needles in the World's Biggest Haystack*, and shows progression of search engine complexity in chapter 3, which introduces PageRank, the algorithm Google relied on in its early days. Specific examples of TF-IDF used for information retrieval are seen as well in *Learning Ontology Relations by Combining Corpus-Based Techniques and Reasoning on Data from Semantic Web Sources*, by Gerhard Wohlgennant, and *Enhancing Information Retrieval through Statistical Natural Language Processign: A Study of Collocation Indexing.* by Arazy, Ofer, and Woo.

III. Methodology

The main components for this classification problem are pre-processing of the raw CSV data, reading in and segmentation of the text data, feature extraction, and calculating cosine similarity in or to determine relevancy of TED Talk to search term.

A. Project Data and Data Preparation

The program begins by prompting the user for search terms which are collected, cleaned, and tokenized using NLTK. The user is also prompted for the number of returned items they would like to see. The returned list of TED talks will always be sorted from most relevant to least according to our model.

For this work, the text files are read in locally; the data is stored within a PyCharm project. The project data consists of 2,466 TED Talk transcripts and associated URLs in CSV format. Once the data is read in, it is converted to a dataframe using the Pandas Data Analysis Library for Python to allow for easier access and use. Transcripts are cleaned and tokenized so that content can be analyzed.

B. Feature Extraction

This project uses term frequency-inverse document frequency (TF-IDF) in order to determine which transcript best fits the user's search. TF-IDF provides a statistical analysis of term importance in a given document, returning high values for terms that provide information in a document. TF-IDF uses normalized term frequency in its final calculations order to account for potential bias in longer documents. Overall, TF-IDF is seen as a fairly robust model, and was used by Google in its early days. TF-IDF is calculated for each transcript as well as the search terms.

C. Classification

For the classification portion of the project, cosine distance was calculated between TF-IDF vectors of the search term and all TF-IDF vectors calculated from each transcript. Using cosine similarity takes vector angle into account rather than just value alone, allowing for better similarity analysis

IV. Results and Analysis

For this work, it is difficult to determine concrete metrics of success. In general, search engine accuracy is assessed through human judgement. We assessed our results by going to the links that our search provided and watching the TED talk to see if it was related to our query terms. In general, our model seems to work pretty well, especially when it is given just a few key words as search terms. Querying for TED Talks related to 'meditation' and requesting 3 results returned Andy Puddicombe's talk titled *All it Takes is 10 Mindful Minutes*, Matthieu Ricards talk titled *How to Let Altruism Be Your Guide*, and Claron McFadden's talk titled *Singing the Primal Mystery*. Watching these talks confirms that they are all related to meditation. Other searches yielded relevant results as well. The system generally did well with search terms on their own, for instance 'computer science', 'meditation', and 'traveling'. However, it was unable to return some relevant talks if the search terms did not directly match the transcript. Searching for 'computer science' in our tests failed to produce the top result on the TED website under computer science, which explains how hard drives work. While this is very relevant to computer science, the speaker, Kanawat Senanan, only mentions the word 'computer' once in the talk, and never says the word 'science', so it is not recognized as a highly relevant result in our model.

However, it struggled to provide accurate results when search terms were entire sentences. For instance, entering the query, "I want to learn about money." returns Nora York singing the song *What I Want*, and Jay Walker's talk *The World's English Mania*, which is about why individuals around the world want to learn English before it returns the third result, which is about tech corporations and economies surrounding them. So, in future search engines, it may be worth implementing a feature that filters out common search terms in addition to stop words and punctuation when the data is being cleaned.

We can also compare our search engine to existing search engines in order to assess our results. We compared our results to those from the official TED Talk search engine on the TED Talk homepage, and in general we had different top results, but our engine's results often showed up somewhere in the returned list on the TED website. The TED search engine is likely more robust than ours, and seems to take much more into account than transcript. It seems to prioritize more recent talks, and also appears to rely on talk description, speaker name, and tags while ours looks only at transcript.

V. Threats to Validity

As mentioned above, our search engine only takes transcript into account in order to determine relevancy. However, a better model would incorporate other aspects, as the official TED search engine does. Newer lectures in general would be more relevant, and older talks may even be outdated. Assessing description of each talk would improve our model as well, as the description provides a concise explanation of what each talk is about. That being said, if we were to choose one feature of a TED talk to assess in a search engine, actual content of each talk (i.e. transcript) is likely the best option. There is definite room for improvement, but improvements would rely on increased data.

VI. Conclusion

This project was able to consistently return relevant TED talks when given individual search terms, but struggled with longer sentences as search terms. There is definite room for improvement, but improvements would rely on increased data. This project served as a continuation of our exploration into natural language processing, and provided a fairly new concept compared to previous assignments, which generally focused on assigning tags through NLTK. We relied largely on human judgement to assess the accuracy of our model which may be criticized as subjective, but in this instance was appropriate.

Acknowledgment

Thank you to Professor Chen, who taught us NLTK fundamentals and the theoretical and practical underpinnings of natural language processing.

References

[1] MacCormick, John. Nine Algorithms That Changed the Future: The Ingenious Ideas That Drive Today's Computers. Princeton University Press, 2012. JSTOR, www.jstor.org/stable/j.ctt7t71s. Accessed 24 Apr. 2020.

[2] Neves, Deangela. "How to Build a Search Engine from Scratch in Python - Part 1." Medium, Medium, 2 Apr. 2018, medium.com/@deangelaneves/how-to-build-a-search-engine-from-scratch-in-python-part-1-96eb240f9ecb.

[3] "Results and Evaluation." Learning Ontology Relations by Combining Corpus-Based Techniques and Reasoning on Data from Semantic Web Sources, by GERHARD WOHLGENANNT, NED - New edition ed., Peter Lang AG, Frankfurt Am Main, 2011, pp. 159–194. JSTOR, www.jstor.org/stable/j.ctv9hj8nd.7. Accessed 24 Apr. 2020.

[4] Arazy, Ofer, and Carson Woo. "Enhancing Information Retrieval through Statistical Natural Language Processign: A Study of Collocation Indexing." MIS Quarterly, vol. 31, no. 3, 2007, pp. 525–546. JSTOR, www.jstor.org/stable/25148806. Accessed 24 Apr. 2020.