NLP Project: Content-based Fresh Air Episode Recommender
Jan. 26, 2022
Metis Online Flex
Julianne Skinner

Abstract

With this project, I explored the topics central to Fresh Air's interview content, and created a recommender to suggest show segments similar in content to one selection. I used Latent Dirichlet Allocation to model the topics present in a corpus of episode transcripts.

Design

I first reduced episode transcripts to several topic features using LDA, assuming each episode included multiple topics to varying degrees. Exploring the results, I visualized how topic prominence changed over time. I also investigated dynamic topic modeling and experimented with fitting a model, but found that fine-tuning it did not fit in the scope of this project. To recommend episodes similar to a particular episode, I used cosine similarity to find the episode with the most similar content, and added a couple of additional features, namely episode recency and length.

Data

To build the dataset, I scraped Fresh Air's show archives, which include transcripts for many of the episodes dating back to the mid-2000s. I was able to collect about 7000 transcripts, which were multimodally distributed in length, with shorter segments centered around 300 words and longer segments around 1000. On average, the corpus had about 1300 tokens per document after lemmatization.

Algorithms

Preprocessing:

Tokens were parsed using the spaCy library's text processing pipeline, to take advantage of its parsing, lemmatization and named entity recognition. Named entites tagged as "person" were removed to avoid giving individual names too much influence on topic definitions. Tokens were limited to text with at least 2 alphabetical characters, and the built-in stopwords from spaCy were amended with common speech phrases found in the documents.

Topic modeling:

For topic modeling, I used gensim's LDA model implementation to adequately model longer documents with multiple topics.

Recommender:

The recommender algorithm uses cosine similarity to find episodes with similar content, and recommendations are weighted by air-date recency and similarity in length.

Tools

- Selenium and BeautifulSoup: data collection

- Google Cloud: connected to an 8-core machine for additional computing power
- SpaCy and sklearn: document parsing
- gensim: LDAmodel
- pyLDAvis: topic visualization

Communication

Results will be shared in a slide deck during the course presentation.