Author: Tim Rappold

Project:

Political and Linguistic Polarization in Network News

Project Design

In this project, I analyze patterns in the langauge of four current affairs shows on MSNBC and Fox News.

My goal is to identify political polarization between the two news broadcast networks, which are respectively known for their liberal and conservative political leanings. There are two questions I'd like to answer:

1) How do the topics covered by conservative and liberal-leaning networks differ? Will an unsupervised machine learning algorithm, such as Latent Dirichlet Allocation, capture difference in pattern, and what are the overlaps? The project uses Latent Dirichlet Allocation for Topic Modeling.

2) Given a set of search terms and topics of interest, which network is more likely to provide the relevant coverage? This question is addressed using Latent Semantic Indexing.

Tools

This project uses the following libraries: `BeautifulSoup` and `Selenium` for data acquisition. `nltk` (the lemmatizer, in particular) and `gensim` for natural language processing, and `gensim`'s LDA and LSI models. I tried other tools and models as well, such as `SpaCy`, but these weren't used in the final product.

Data

We're looking at episode transcripts from the four different shows, two shows from each network, broadcast weeknights in competing time slots:

Time	FOX News	MSNBC
9pm	Hannity	Rachel Maddow Show
10pm	The Ingraham Angle	The Last Word with Lawrence O'Donnel

Show transcripts were readily available on the networks' respective websites. However, data acqusition was constrained by the limited availability of transcripts from *The Ingraham Angle*. As a result, the project transcripts from the last six months but no earlier.

Each show was broadcast about 120 times in that time span, so the total corpus will comprise speech transcripts from about five-hundred hour-long episodes.

Algorithm

For this project is built a code pipeline with the following structure:

.

+-- README.md

+-- bin/

| +-- `client.ipynb`

| +-- `scraper.py`

| +-- data/

| +-- `processtext.py`

| +-- `models.py`

+-- docs/

`scraper.py` acquired all show transcripts from *foxnews.com* and *msnbc.com* and serialized them in the `data` folder. "Manual" text processing was mostly done in `processtext.py`, especially via the `Episode` class. This class deploys a significant number of custom text processing methods. `models.py` applies further natural language processing (Lemmatization, removal of stopwords, etc) as well as LDA and LSI from the `gensim` package to the pre-processed text data. `client.ipynb` provides the interface for the project, including all plotting.

What I'd do differently next time

I'd add Word2Vec to the analysis and experiment with word and dcoument meanings.