Semester1Progress: Topic Modeling Ethnographic Interviews

Fiona Adams

1/3/2020

Introduction

This will be heavily dependent on the final product, so I'm waiting to write an intro/lit review!

Some useful lit/"lit" so far: https://www.researchgate.net/publication/299552252_Statistical_Topic_Modeling_for_News_Articles, https://arxiv.org/pdf/1808.01175.pdf, https://towardsdatascience.com/thats-mental-using-lda-topic-modeling-to-investigate-the-discourse-on-mental-health-over-time-11da252259c3, https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158, http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf

Dataset

This dataset is from the Minnesota Opioid Project, a collection of 50 ethnographic interviews conducted by Amy Sullivan. The interviewees are all Minnesotans influenced by the opioid epidemic. They include treatment professionals, people who previously had Opioid Use Disorder (OUD), and people who have seen a family member die from opioid overdose. Each interview has about 10,000 to 15,000 words, meaning topic modelling is a potential time-saver.

Goal

Make ethnographic interview analysis easier! Not unsupervised necessarily, but not *mind-numbingly tedious* as Amy describes it.

Tried

Noun/verb phrases: Split the documents on the sentence level, then got key noun/verb phrases for each sentence and made a new dataset with these phrases. Then, clustered noun/verb phrases together based on string distance. Result: unintelligible clusters, heavy overlap

Document comparison: Calculated cosine similarity & Euclidean distance between documents. See http://text2vec.org/similarity.html. Result: great for comparing and determining whether documents are similar, validating the usefulness of topic modeling. But, not useful for final product.

More noun/verb phrase: Using same approach as final (split on paragraph level, same cleaning), I took the top # (played around with this #, settled on 500 for computation reasons) of noun/verb phrases from the corpus. Then, I filtered paragaphs so that each paragraph is consisted of just its top noun/verb phrases. Afterwards, I performed LDA on these newly filtered paragargphs. Result: clearly labelable clusters, but clusters didn't give unique results, so not very useful for historians

Different dataset Repeated the above and final approach on a general dataset of Wikipedia articles, in line with methodology from https://arxiv.org/pdf/1808.01175.pdf. Hope was to compare clusters and see if it was easier to label given clusters from another dataset. But, this had a super long computation time. Result: Not useful–but may be worth approaching again

Cleaning the Dataset

For initial cleaning, I first split each interview into its respective paragraphs, then treated each paragraph as a separate "document" but label the paragraphs based on the document they originally came from (ex.

"doc 9" can be mapped to doc 9 in the original dataset). Then, I removed paragraphs with <20 characters, meaning most of the questions Amy asks in the interview are filtered out, and any "yes," "no," etc. answers are gone as well.

Then, using R's stopwords package, I took out "stop words," which included "it," "be," etc. An example of this can be shown below, where the first paragraph includes stopwords and the second does not.

```
## [1] "ga: my name is greg anderson. i am the social service supervisor for st. louis county public hea
```

```
## [1] "ga: name greg anderson. social service supervisor st. louis county public health human services
```

Then, I calculated skipgram probabilities to determine which words occurred together, to then concatenate them. For example, in this dataset, we expect the words "substance" and "abuse" to appear together, meaning their skipgram probability would be high and they would be concatenated into "substance_abuse" for easier clustering. Typically, word vectors are calculated using neural networks. The approach below, of finding words that occur together in the corpus of Minnesota Opioid Project interviews, uses only counting and linear algebra. This is great because it eliminates the need for pre-trained vectors in a deep learning approach, uses familiar techniques that are relatively easy to understand, and doesn't take too long computationally [@juliasilge]. More reasons to not use neural network approaches are here: [@multithreaded].

Skipgram probabilities: how often we find each word near each other word.

How to get these probabilities: Define a fixed-size moving window that centers around each word. What is the probability of seeing *word1* and *word2* in this window?

Defining the moving window size: When this window is bigger, the process of counting skipgrams takes longer. Julia Silge, a well-known data scientist at Stack Overflow, used windows of 8 words, so I decided to start with this. Going forward, I'm looking to take some more sophisticated steps to find the best window to use.

Concatenate words with high co-occuring probabilities: If probability of co-occuring is relatively high (in this case, I took the top 100 probabilities), then concatenate those words. This gave the best clusters, but hoping for a more unsupervised approach to finding the top skipgrams going forward, so will continue to hone this.

Here is an example of concatenating words that occur together using the words "like_just" which occurred together a few times in our dataset.

```
## [1] "km - say well honey (unclear) go oh things_like_that. really struggled started get angry. seeme
## [2] "km - well, just something effect judge great. sure flames will licking ass. little south dakota
## [3] "sp: absolutely making sense. just_thinking wanted go that. think that, strong, sorry gonna use
```

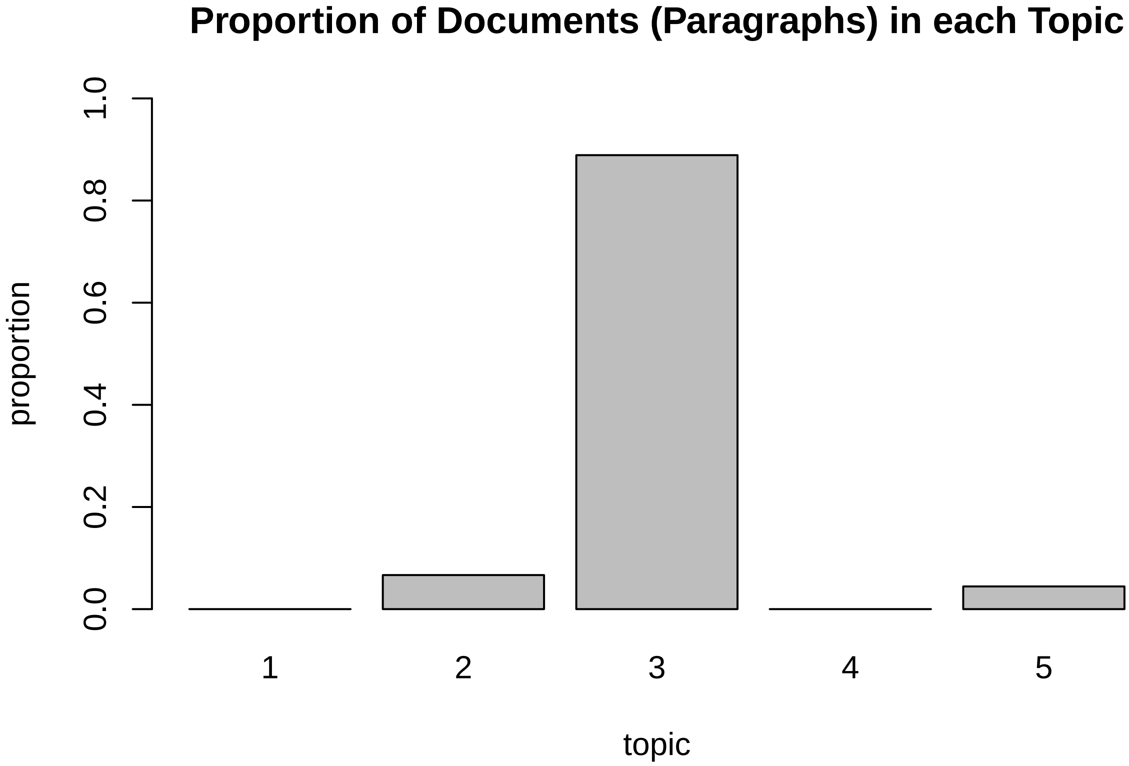
Future cleaning steps will likely include stemming (getting the base or root form of the word) and/or lemmatization (getting a different base form, or dictionary form of the word, the "lemma"). Stemming, for example, will transform "was" to "wa," while lemmatization will transform "was" to "be." Initial work with these methods led to some loss of meaning, but I am looking to revisit using different methods. I will also be looking into CountVectorizer: https://www.rdocumentation.org/packages/superml/versions/0.4.0/topics/CountVectorizer

Topic Modeling: Creating Clusters using Latent Dirichlet Allocation (LDA)

LDA gives a probabilistic topic model, with tables of 'words-versus-topics': the probability or chance of selecting a particular part when sampling a particular topic (category) and 'documents-versus-topics': the chance of selecting a particular topic when sampling a particular document or composite. In this case, we are using 'documents-versus-topics,' or more accurately, 'paragraphs-versus-topics,' because we treat each paragraph as its own document. LDA allows for "fuzzy" memberships to topics rather than outright ones as in k-means, which is “hard-clustering.” This provides a more nuanced way topic modeling. However, LDA is hard to tune and hard to evaluate

Haven’t yet written out full a “methods” section for LDA, but will add here soon. Want to wait on finalizing cleaning methods such that LDA provides easy-to-understand clusters.

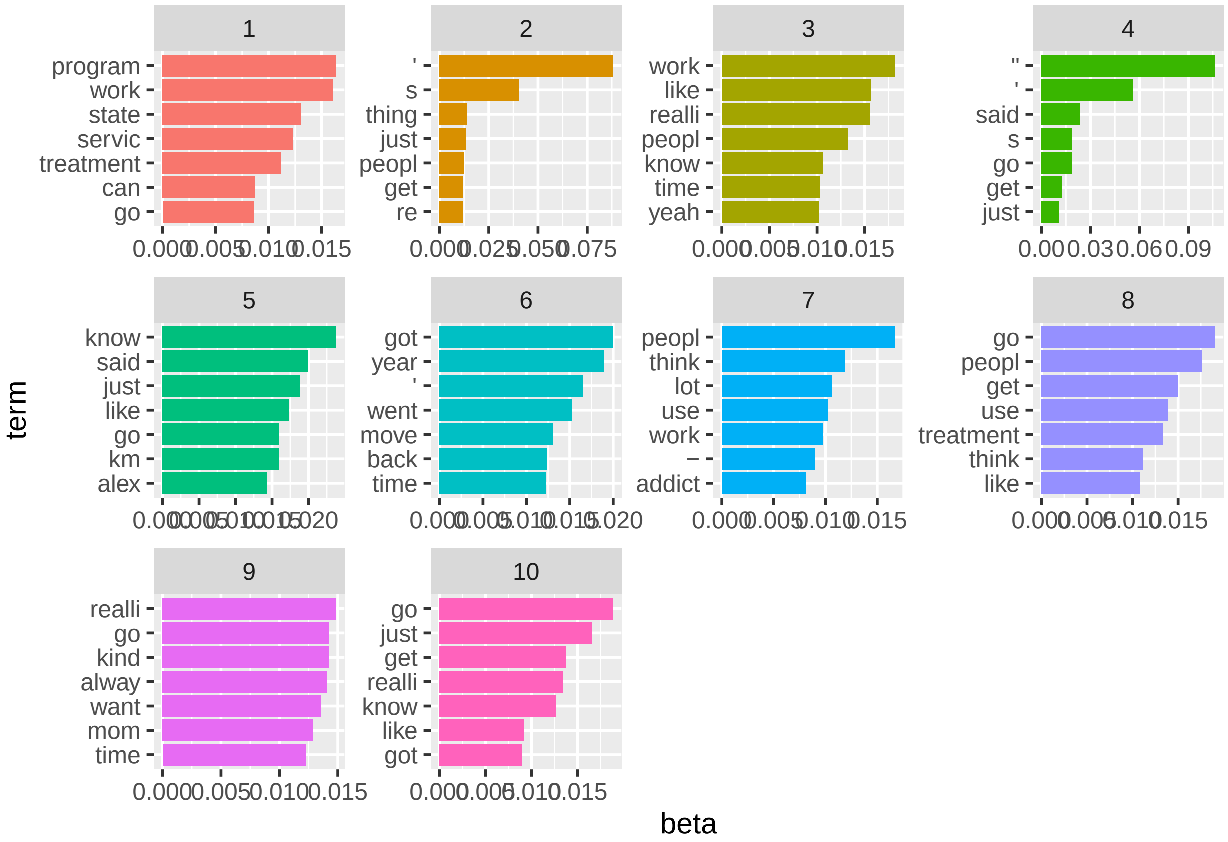
You can see below that with 5 topics, only 3 really have documents within them. In future, I will likely create a function that chooses a number of topics n such that proportion of documents in each topic is roughly 100/n +/- an error bound.



```
library(LDAvis)
lda_model$plot()
```

This code isn’t showing once knitted, but for me brings up a webpage with an interactive LDA/PCA model on it. To get there, my link is http://127.0.0.1:4321/, but you may need to run this .Rmd file to get it up.

Here is an alternate way of making an LDA model, without PCAs. (Needs Methods sec!)



Note that this is *not* a clustering on the full dataset as of yet (due to computation time concerns). Thus, these clusters may not be as useful as they would be on a full dataset of 50 interviews. Will be running the full dataset very soon! That said, these clusters aren't as useful as I wish they were in general.

Future modeling steps will likely include the following: • Re-run LDA where each "document" is a smaller phrase rather than a full paragraph

- See https://www.sciencedirect.com/science/article/pii/S0020025518308028 for comparison of methods–LDA has some big limitations!

- See if achieve better clusters using *Nonnegative Matrix Factorization*. See https://towardsdatascience.com/topic-modeling-for-the-new-york-times-news-dataset-1f643e15caac

- See if achieve better clusters using *Stochastic Block Model*. See https://advances.sciencemag.org/content/4/7/eaaq1360

- Determine whether worthwhile to attempt a *doc2vec* approach, using Python (a language I'm less familiar with). See https://arxiv.org/pdf/1808.01175.pdf and https://towardsdatascience.com/using-word2vec-to-analyze-news-headlines-and-predict-article-success-cdeda5f14751 and https://cs.stanford.edu/~quocle/paragraph_vector.pdf

Next Steps

Will be in final:

- Manually label each cluster based on its themes

- Map the words from each cluster back to sentences in each document

- Work on "auto-labelling" clusters using different dataset to train, ex. addiction medicine handbooks, or Wikipedia articles specifically about substance abuse

Dataset: Description of Each Person Interviewed

Mark Willenbring	Doctor at NIH
Verne Wagner	Father of son addicted to meth, started NarAnon group
Andrew Tuttle	Psychiatrist, became addicted to opioids
Lorraine Teel	Started program for people addicted primarily to opioids
Kathie Simon Frank	Mother of daughter addicted to opioids
Yussuf Shafie	CEO and the director of Alliance Wellness Center.
Marvin Seppala	Once addicted to amphetamines
Star Selleck	Father of son addicted to various substances
Shelley Roberts Gyllen	Sister to brother who died from narcotic overdose
Charles Reznioff	Doctor focused on addiction medicine
Sue Purchase	Harm reduction specialist
Kim Powers	Mother of daughter addicted to opioids
Cody Petrich	Son of mother addicted to opioids
Ann Perry	Mother of son addicted to opioids
Margarita Ortega	Ex-opioid user
Michael O'Neill	Ex-cocaine user and father of son addicted to opioids
Richard Moldenhauer	Worked at various drug treatment centers
Kirsten Milun	Mother of son addicted to opioids
Ian McLoone	Ex-opioid user and son of mother addicted to opioids
Rose McKinney	Mother of child addicted to opioids
Mary McCarthy	Harm reduction professional
Lori Lewis	Mother of son addicted to opioids
Robert Levy	Addiction medicine doctor
Wade Lang	Ex-opioid user
Maris Krause	Ex-opioid user
Chandra Kelvie	Daughter of parents who used opioids
Maggie Kazel NO TRANSCRIPT	Harm reduction professional
Jeff Kazel	Law enforcement professional
Dean Johnson	Father of son addicted to opioids
Chris Johnson	Addiction medicine doctor
Julie Hooker	Treatment professional
Janise Holter	Mother of child addicted to opioids
Deb Holman	Treatment professional
Chuck Hilger	Ex-opioid user and treatment professional
Carson Gardner	Mother of child addicted to opioids, researcher
Frank Eden Rae	Treatment professional focused on harm reduction
Carol Folkowski	Treatment professional
Adam Fairbanks	Treatment professional focused on harm reduction
Robin Evanson	Ex-opioid user
Nancy Espuche	Mother of child addicted to opiods
Gloria Englund	Mother of son addicted to opioids
Stephanie Devich	Treatment professional
Paula DeSanto	Treatment professional
Jamison Danielson	Treatment professional focused on harm reduction
Brandon Coleman	Son of mother addicted to opioids and ex-opioid user
Bill Cole	Father of child addicted to opioids
Emily Brunner	Addiction medicine doctor
Janie Bining Colford	Mother of child addicted to opioids
Linda Berry-Brede	Mother of child addicted to opioids
Thilo Beck	Addiction medicine doctor
Greg Anderson	Social worker