Text Analysis of *Hamilton*

Nathan Schor

March 13, 2022

Contents

1 Introduction 2

2 Literature Review 2

3 Data 3
- 3.1 Corpus 3
- 3.2 Stop Words 3
- 3.3 Sentiment 3

4 Research Design and Modeling Methods 4
- 4.1 Ontology 4
- 4.2 Sentiment Analysis 5
- 4.3 TF-IDF 6
- 4.4 Topic Modeling 7
- 4.5 Chat Bot 7

5 Results 8
- 5.1 Sentiment Analysis 8
- 5.2 TF-IDF 8
- 5.3 Topic Modeling 9
- 5.4 Chat Bot 11

6 Analysis and Interpretation 12
- 6.1 Sentiment Analysis 12
- 6.2 TF-IDF 13
- 6.3 Topic Modeling 13
- 6.4 Chat Bot 14

7 Conclusions **15**

1 Introduction

The purpose of this research is to apply natural language processing (NLP) to the lyrics of the musical *Hamilton*. As part of the analysis, we experiment with three different sources of stop words, four different sentiment lexicons, and tokenizing at the word vs. sentence level. Term frequency-inverse document frequency (*tf-idf*) is used to identify the most important terms for each character, a topic model is created to see if it can identify characters based on phrases most likely to belong to them, and we build a chat bot that can answer general questions about the musical and our analysis. We begin by building an ontology. Next, we examine how sentiment changes throughout the musical. Then, we use *tf-idf* to identify the most important word for each speaker. After, we see if the topic model is able to identify the speaker based on their words. All of this is performed using resources described in Silge and Robinson (2022). Lastly, we develop a chat bot that is able to answer questions on a corpus that contains *Hamilton's* Wikipedia page augmented with results from this analysis.

2 Literature Review

A number of earlier works look at the impact that the choice of sentiment lexicon and stop word dictionary play in NLP analyses. Shukla et al. (2017) apply sentiment analysis to songs. Khanna (2021) looks at removing stop words with different Python libraries. HaCohen-Kerner et al. (2020) look more generally at the role that preprocessing plays in NLP.

Table 1: Five rows from the *Hamilton* dataset.

Entry	**Title**	**Speaker**	**Lines**
1	The Election of 1800	Jefferson	’cuz I’m the President. Hey, Burr, when you see Hamilton, thank him for the endorsement
2	Blow Us All Away	Philip	I came to ask you for advice. This is my very first duel
3	It’s Quiet Uptown	Hamilton	I take the children to church on Sunday
4	Satisfied	Angelica	Have to be naive to set that aside
5	Right Hand Man	Washington	But the elephant is in the room

3 Data

3.1 Corpus

The main dataset is obtained from the website Kaggle (2022). It contains the entirety of the musical in a csv file with 3,634 rows and 3 columns. The *Lines* column gives the line that is spoken. The *Speaker* column gives the names of the people speaking (there can be more than one character singing at the same time). *Title* gives the name of the song. 5 random rows of the data are shown in Table 1. The main corpus for the chat bot is Wikipedia (2022).

3.2 Stop Words

Three sources are used to obtain the list of stop words. They are the SMART lexicon from Lewis et al. (2004), snowball from Porter (2008), and onix from bluehost (2012). SMART contains 571 stop words, snowball has 174 stop words, and onix has 404 stop words.

3.3 Sentiment

Three lexicons are used to obtain sentiment. They are the afinn dataset from Äruprup Nielsen (2011), the nrc dataset from Mohammad and Turney (2013), and the bing dataset from Hu and Liu (2004).

afinn assigns words an integer value in $\{-5, -4, ..., 4, 5\}$ with -5 having the most negative sentiment and 5 having the most positive sentiment. Swear words/insults have -5, words like “superb” or “breathtaking” are 5, and “some kind” is 0. There are 2,477 terms.

The nrc dataset assigns each of its 13,875 words to 1 of 8 emotional categories. For the purpose of this analysis, only words in the “positive” or “negative” emotion category are retained.

The only sentiment dataset ready for off-the-shelf use is the bing dataset. It classifies 6,786 terms as having either positive sentiment or negative sentiment. It is worth noting that roughly 70% of the terms are negative, so negative sentiment words are over-represented.

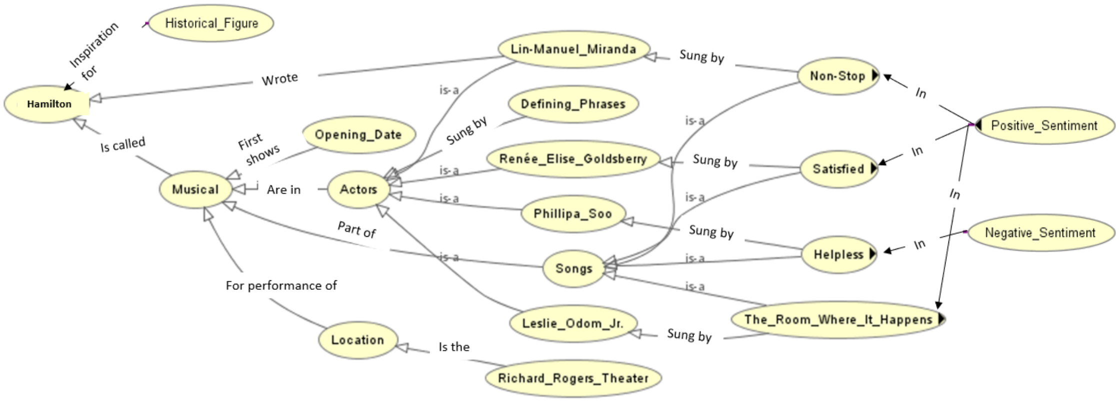
4 Research Design and Modeling Methods

4.1 Ontology

I began by thinking about which aspects of *Hamilton* would be interesting to analyze using NLP tools. To structure my thinking, I constructed an ontology. The key pieces of the musical are its songs and actors. Within songs, I was curious how the mood changed throughout the duration of the play. For the actors, I was interested in analyzing whether particular words or phrases could summarize a character’s behavior. Figure 1 shows the ontology I created using Protégé from Musen (2015). It served as my starting point for which kind of NLP questions to tackle.

The 4 main areas of investigation were analyzing mood via sentiment analysis, identifying the most important words for speakers using *tf-idf*, seeing if we could “uncover” the speaker using topic models, and creating a chat bot to

Figure 1: Ontology for *Hamilton*.



answer general questions about the play as well as information from the first 3 parts. Each of these are discussed in turn.

4.2 Sentiment Analysis

The two questions for this section are how much does differing the source of word sentiment affect the "time series" of the play's sentiment, and which song s has the greatest mood shift, defined as $max(Sentiment_s - Sentiment_{s-1})$ for $s = \{2, 3, ..., S-1, S\}$ ($s-1$ is not defined for $s = 1$).

The *Hamilton* lyrics were first tokenized into words and normalized to lower case. A song's sentiment was calculated as $Sentiment_s = \sum_{w=1}^{W} w$ where W is the total number of words in song s. This approach has a few major limitations. One is that a song's sentiment is not necessarily the sum of its parts. For example, *It's Quiet Uptown* starts off with the death of Hamilton's son and has very negative sentiment. The end of the song has a positive sentiment as Hamilton and Eliza rekindle their marriage. This simple counting fails to capture the mood shift that transpires during the song. Furthermore, tokenizing at the word level does not capture the impact of negative modifiers. For example, the sentence "I was not happy with the show" should have negative sentiment, but the term “happy” will count as positive sentiment.

The *sentimentr* package from Rinker (2021) helps to overcome some of these shortcomings. It tokenizes words at the sentence level rather than at the word level. This additional context can help to assign the correct overall sentiment to a sentence since its sentiment is not necessarily the sum of its individual tokens.

4.3 TF-IDF

tf-idf is comprised of two parts: term frequency *tf* and inverse document frequency *idf*. The idea is that “important” words are captured by the seemingly contradictory idea that important words are both frequent and rare. A word’s term-frequency is the number of times the word appears in the document divided by total number of words in the document (this is a way to control for the size of the document). The word’s inverse document frequency is the natural logarithm of the number of documents in the corpus divided by the number of documents in the corpus that contain that word. The idea is that a common word such as “the” will have a high term frequency, but is not necessarily important or informative. However, “the” will have a low inverse document frequency because it is likely that every single document in the corpus will contain the word “the”. Using *tf-idf* helps to balance these two by multiplying them together. Thus, a word with a high *tf-idf* both occurs frequently in a given document, but is rare in the corpus as a whole.

We use *tf-idf* as the metric to determine which are the most important words for a given speaker. For this analysis, we define a speaker as a character who has 10 or more solo lines. In the musical, many characters are singing simultaneously. In the dataset, it is challenging to parse out which words are attributed to each speaker since a list of speakers can be given for the same line (there is a many-to-many relationship between speaker and line). Thus we only look at lines where a speaker was the only singer in this analysis (which introduces some bias since the presence or absence of other characters could impact the speaker's word choice).

4.4 Topic Modeling

Topic modeling is an unsupervised learning method that seeks to group documents into k similar documents, or topics. Since it is unsupervised, there is no loss function that can be used to optimize k since we do not have a "ground truth." One of the most common algorithms for topic modeling is Latent Dirichlet Allocation (LDA) that treats every document as being comprised of multiple topics, and every topic as being comprised of multiple words. LDA estimates both of these quantities simultaneously. In this project, we look at the solo lines sung by Hamilton, Eliza, and Washington. We see if we can recover words that uniquely identify these characters. Thus, the true k is 3, and we see if the topic model is able to identify the 3 "topics."

4.5 Chat Bot

The final portion of this project creates a chat bot following Dass (2018) that can answer questions about the *Hamilton* Wikipedia page and on the previous 3 subsections. We do this using the following NLP pipeline:

1. Convert all strings to lower case
2. Tokenize the data at the sentence level
3. Perform lemmitization
4. Calculate the *tf-idf* for each token
5. Return the entry from the corpus that has the largest cosine similarity with the query

Cosine similarity measures the "distance" between two quantities by computing the angle θ between them. Each token is plotted as a vector using its *tf-idf*. As we see from the formula $cos(\theta_{token1,token2}) = \frac{token1 \cdot token2}{||token1|| \cdot ||token2||}$, the cosine similarity for two vectors is between -1 and 1. Two identical vectors will have a cosine similarity of 1. The chat bot only returns results for queries that have a cosine similarity of at least 0 with another sentence in the corpus.

5 Results

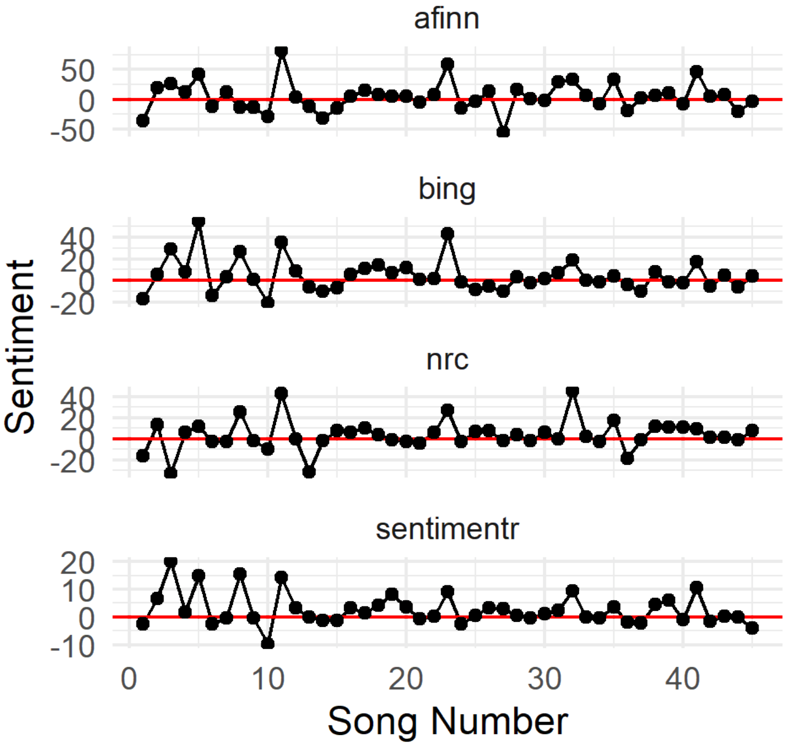
5.1 Sentiment Analysis

Figure 2 shows how sentiment changes over the course of the musical. The song number (1-45) is displayed on the x-axis. The y-axis shows the song's overall sentiment. Notice that the scale between sentiment datasets can differ; for example, sentiment for the afinn dataset is between -50 and 50, while sentiment from *sentimentr* is between -10 and 20. The important aspect is not the absolute value of the sentiment, but the direction (positive or negative) and a song's sentiment value relative to other songs using the same sentiment dataset.

5.2 TF-IDF

Table 2 displays the word with the largest *tf-idf* for each speaker. The SMART, snowball, and onix columns had those respective stop word dictionaries applied to the data. The "All Lexicons" column had all 3 stop word dictionaries applied, and "All Words" did not remove any stop words. Speakers who appear multiple times had ties for their most important word in at least 1 of the 5 processing coluns. For example, Lafayette's most important word when any kind of stop word processing is used is "onarchy", but without stop word removal his most important word is "oui". Similarly to section 4.3, a speaker is defined as

Figure 2: Sentiment analysis by sentiment dictionary.



someone who is the only person singing a given line. This removes ambiguity over which speaker is singing a line.

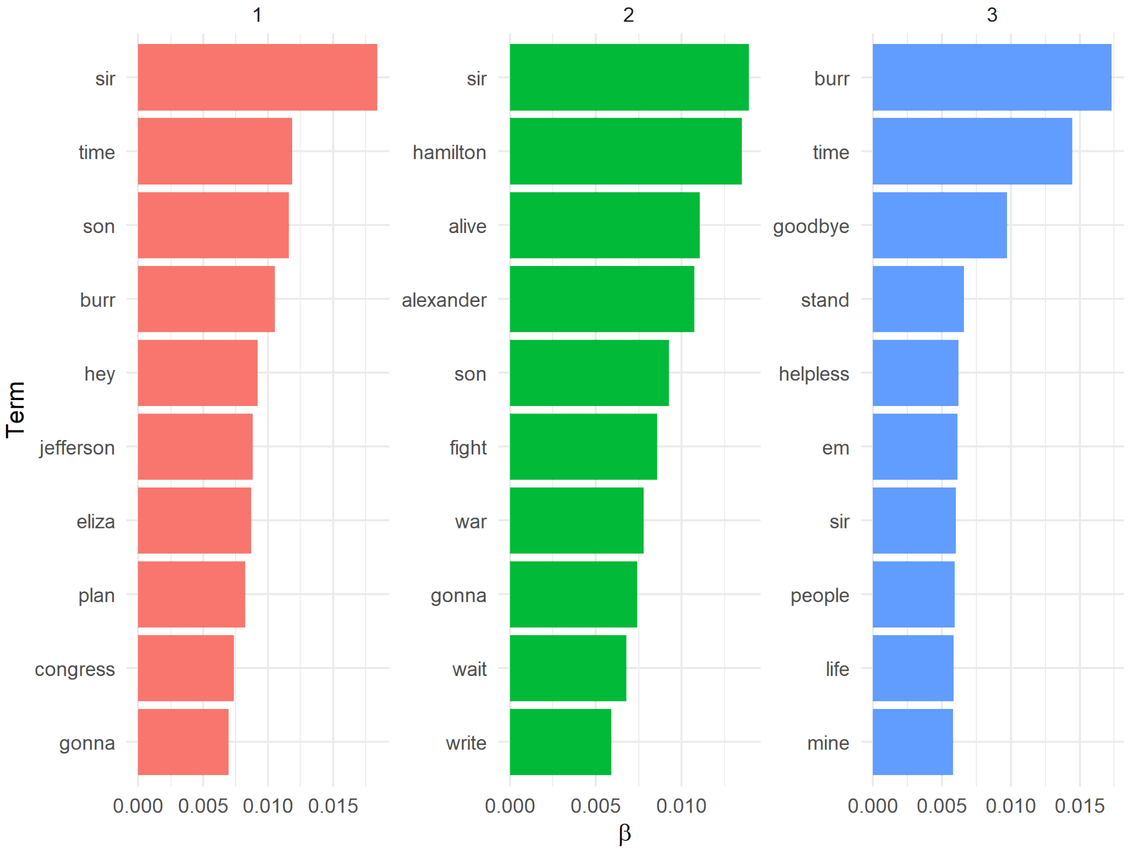
5.3 Topic Modeling

In Figure 3 we have a topic model. The y-axis shows 10 terms for each topic. On the x-axis is β, which is the probability that the given term was generated from its assigned topic. For example, the term "sir" has a .02 probability of being generated from topic 1. Notice that terms such as "sir" can overlap between topics. This seems reasonable given that all of the "topics" are speakers in the same play.

Table 2: Speaker *tf-idf* by stop word lexicon.

Entry	**Speaker**	**SMART**	**snowball**	**onix**	**All Lexicons**	**All Words**
1	Angelica	satisfied	satisfied	satisfied	satisfied	satisfied
2	Burr	room	room	immigrant	immigrant	room
3	Company	number	number	aaaah	aaaah	number
4	Eliza	huit	enough	look	huit	enough
5	Eliza	huit	enough	look	sept	enough
6	Ensemble	buck	buck	buck	buck	buck
7	Hamilton	sir	sir	sir	sir	i
8	Jefferson	what'd	what'd	what'd	what'd	what'd
9	Lafayette	onarchy	onarchy	onarchy	onarchy	onarchy
10	Lafayette	onarchy	onarchy	onarchy	onarchy	oui
11	Laurens	colonies	colonies	colonies	colonies	colonies
12	Lee	crisis	crisis	crisis	crisis	crisis
13	Madison	size	size	size	size	size
14	Maria	sir	yes	yes	sir	yes
15	Men	satisfied	satisfied	satisfied	satisfied	satisfied
16	Mulligan	hercules	hercules	hercules	hercules	hercules
17	Mulligan	hercules	hercules	hercules	hercules	lovin
18	Philip	deux	deux	deux	deux	deux
19	Philip	deux	deux	deux	deux	huit
20	Seabury	heed	heed	heed	heed	heed
21	Seabury	heed	heed	heed	heed	interests
22	Washington	goodbye	goodbye	goodbye	goodbye	goodbye
23	Women	helpless	helpless	helpless	helpless	helpless

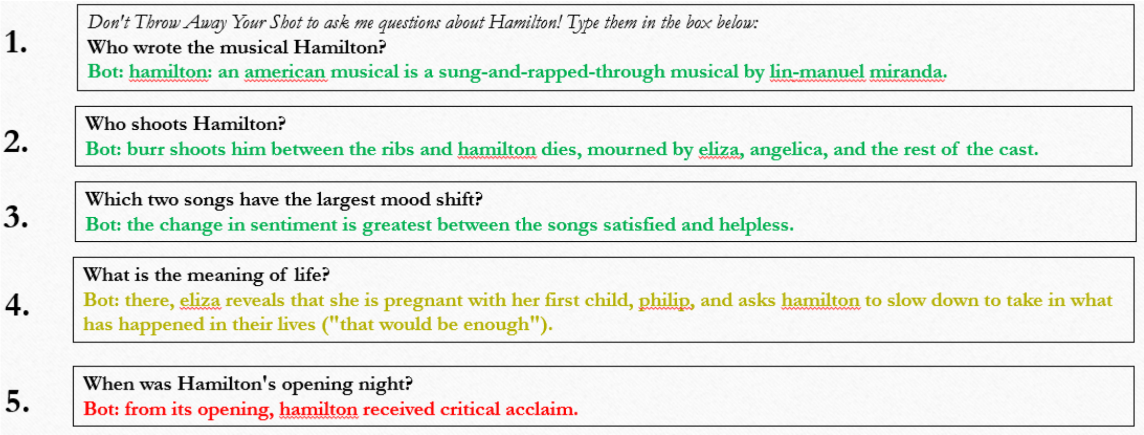
Figure 3: Topic model generated with lines from Hamilton, Washington, and Eliza.



5.4 Chat Bot

An example of the chat bot's output is shown in Figure 4. The text that is generated at the beginning of each conversation ("*Don't Throw Away Your Shot to ask me questions about Hamilton! Type them in the box below:*") is italicized. The user's question is in **bold** and the chat bot's response is colored. A green response indicates that the chat bot answered well, a yellow response means that the response was on the right track but not quite correct, and red means the response was unsatisfactory.

Figure 4: Example output from 5 queries fed to the chat bot.



1\.
Don't Throw Away Your Shot to ask me questions about Hamilton! Type them in the box below:
Who wrote the musical Hamilton?
Bot: hamilton: an american musical is a sung-and-rapped-through musical by lin-manuel miranda.

2\.
Who shoots Hamilton?
Bot: burr shoots him between the ribs and hamilton dies, mourned by eliza, angelica, and the rest of the cast.

3\.
Which two songs have the largest mood shift?
Bot: the change in sentiment is greatest between the songs satisfied and helpless.

4\.
What is the meaning of life?
Bot: there, eliza reveals that she is pregnant with her first child, philip, and asks hamilton to slow down to take in what has happened in their lives ("that would be enough").

5\.
When was Hamilton's opening night?
Bot: from its opening, hamilton received critical acclaim.

6 Analysis and Interpretation

6.1 Sentiment Analysis

Thoughtfully selecting a sentiment dataset is important to the results. Each of the sentiments in Figure 2 have large swings in sentiment up until song 11 (*Satisfied*), after which the sentiment has lower variance. All graphs identify *Satisfied* as one of the most positive songs and *Helpless* (song 10) as one of the most negative songs, consequently identifying the *Helpless* → *Satisfied* transition as the largest shift in mood. All graphs show that Act I ends on an extremely positive note with *Non-Stop* (song 23). However, afinn and *sentimentr* identify the final song (*Who Lives, Who Dies, Who Tells Your Story*) as having negative sentiment while nrc and bing classify the song as overall positive. There are a number of songs where the datasets diverge in classifying the song's valence. Thus, sentiment is conditional on the dataset used.

6.2 TF-IDF

Similarly, it is important to carefully construct a list of stop words that are *corpus specific*. For example, in Table 2 “sir” could reasonably be considered a stop word. It is interesting how changing the stop word preprocessing step affects the speaker’s most important word. For example, Burr’s most important words are “room” and “immigrant”. “room” is an inward looking term for Burr, reflecting his desire to be in the “room” when decisions are being made and to have a seat at the table. “immigrant” is an outward looking term, reflecting his complicated relationship with Hamitlon–who Burr often refers to as the “immigrant.” It’s also noteworthy that *tf-idf* identifies the “catch phrase” for a number of the speakers: “satisfied” for Angelica, “goodbye” for Washington, “yes” for Maria Reynolds, and “enough” for Eliza.

6.3 Topic Modeling

The topic model in Figure 3 was able to produce reasonable terms for each of the three speakers from which the topics were generated (Hamilton, Eliza, and Washington). Topic 1 seems to be about Hamilton–Hamilton is fixated on the concept of time and speaks frequently with his son (Phillip), Jefferson, Burr, and Eliza. Topic 2 appears to identify Washington. Washington talks frequently with Hamilton and is heavily involved with the Revolutionary War and the fighting. To Hamilton’s dismay, Washington refers to Hamilton as “son” (note that “son” appears in both the “Hamilton Topic” and the “Washington Topic” despite having a slightly different meaning). Eliza is the speaker identified in Topic 3. The term “helpless” is the title of one of her main songs and one of her catch phrases. She often speaks with Hamilton about “life” and unfortunately has to say “goodbye” to both her son Phillip and to Hamilton when they are each killed in duels. This topic modeling experiment illustrates that it has the potential to be used for text summarization in situations where the number of topics is not known *a priori*.

6.4 Chat Bot

In Figure 4, we see that the chat bot performs well on 3 queries, decently on 1 query, and poorly on the final query. A more sophisticated chat bot would be able to answer the question **"Who shoots Hamilton?"** with just the correct answer, "Burr", rather than providing all of the unnecessary context that is present in the Wikipedia document about the characters who mourn Hamilton. In query 3, we see that the chat bot correctly retrieves the relevant information for a query that is targeting the dataset augmented with the results from Figure 2. It seems plausible that a non-technical user would use the term "mood" rather than "sentiment", and we see that the chat bot has no problem retrieving the key sentence.

Query 4 is the most interesting. In a literal sense, the query does not explicitly answer the user's question; the question does not ask about Eliza's pregnancy or her request to Hamilton. However, the chat bot does (perhaps unintentionally) offer a profound response to the age-old question: *"What is the meaing of life?"*. The answer touches on the importance of family, as well as slowing down and taking things one at a time and being happy with what they do have instead of focusing on what is lacking. This example illustrates that while a chat bot which can only retrieve text rather than generate text certainly has limitations, it still has the potential to use its corpus in unexpected ways to produce interesting results.

The last query highlights a limitation of the cosine similarity approach. The bot relies too heavily on the word "opening" and returns irrelevant results. A more sophisticated bot would understand that a necessary (but not sufficient) response would contain a time component e.g. "The show opened **6 years** ago." It would also identify phrases that serve the same purpose as "opening" such as "started", "began", "premiered", and "first shown".

7 Conclusions

Often times, the most exciting parts of doing research are building models and generating informative tables and figures. However, the vast majority of time is spent in data acquisition, EDA, and data processing. This is especially true in NLP. Not only is each corpus different, but each use case with the same corpus is different. We need to take the time to make sure we truly understand the data we are working with so that we can feed useful information into our models. As the quote goes, "Garbage in, garbage out."

This project highlighted for me the importance of curating a corpus-specific stop word list. Off-the-shelf stop words can be a nice start, but the probability that the list was curated with your specific use case in mind is close to 0. The same is true for sentiment lexicons.

The chat bot showed me that something useful can be built with a relatively little amount of code. However, the amount of work necessary to see incremental improvement increases sharply as the chat bot gets better and better. Alternative methods like BERT would likely have produced a better chat bot and is something I would like to explore in future work.

References

bluehost (2012). Lextek. `https://en.wikipedia.org/wiki/Hamilton_(musical)`.

Dass, R. (2018). Create your chatbot using python nltk.

HaCohen-Kerner, Y., Miller, D., and Yigal, Y. (2020). The influence of preprocessing on text classification using a bag-of-words representation. *PLOS ONE*, 15(5):1–22.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Kaggle (2022). Hamilton lyrics. `https://www.kaggle.com/lbalter/hamilton-lyrics`.

Khanna, C. (2021). Text pre-processing: Stop words removal using different libraries.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397.

Mohammad, S. and Turney, P. (2013). Nrc emotion lexicon.

Musen, M. A. (2015). The protégé project: a look back and a look forward. *AI Matters*, 1(4):4–12.

Porter, M. F. (2008). Published online.

Rinker, T. W. (2021). *sentimentr: Calculate Text Polarity Sentiment*. Buffalo, New York. version 2.9.0.

Shukla, S., Khanna, P., and Agrawal, K. K. (2017). Review on sentiment analysis on music. In *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*, pages 777–780.

Silge, J. and Robinson, D. (2022). *Text Mining with R*. Printed by the author.

Wikipedia (2022). *Hamilton* (musical). `http://www.lextek.com/manuals/onix/stopwords1.html`.

Äruprup Nielsen, F. (2011). A new anew: Evaluation of a word list for sentiment analysis in microblogs. *CoRR*, abs/1103.2903.