Can a Topic Model Derive a Song's Topic given the Chorus/Hook?

Sadab Hafiz, Zachary Motassim
CUNY Hunter College
sadab.hafiz52@myhunter.cuny.edu, zachary.motassim92@myhunter.cuny.edu

Abstract

Music Information Retrieval(MIR) is a relatively new interdisciplinary field that requires more research in data collected from music. Topic modeling is a very useful NLP tool that can help us classify music into topics based on the lyrics. A very prominent algorithm that is used for topic modeling is the Latent Dirichlet Algorithm(LDA). Other researchers have already applied LDA and other topic modeling algorithms on different lyrics datasets. We used a relatively new dataset for our research. Our primary focus was on the chorus/hook of a song. This paper attempts to show whether or not a topic model can detect the topic of a song given only its chorus/hook.

1 Introduction

In the modern world, music has become a very important part of our lives. Popular music streaming services like Spotify, Apple Music, Youtube Music, etc. have made music very accessible in the 21st century. Different people have different tastes in music as music can be very subjective. Some people like to listen to only particular genres. Similarly, some people like to listen to songs that fall under a particular topic that they can relate to. People often want to listen to different songs based on their mood. Since music is accessible, it is often hard for a person to find the song that they are looking for. To solve/improve this situation, the field of Music Information Retrieval(MIR) has been very active over recent years. MIR is an interdisciplinary field that deals with the retrieval of information from songs. Applications of MIR include music classifications, music recommender systems, music transcription, music generation, instrument recognition, source separation, and many more (Schedl et al, 2014). While MIR researchers have been very active over recent years, most of the work has been done emphasizing the acoustic features of songs. This is because all songs have one thing in common, the sound! However, disregarding the lyrics is problematic as most popular songs of recent years have had meaningful lyrics. This is why NLP researchers like us have decided to do topic modeling on song lyrics to reveal useful information that contributes to the field of MIR. In our research, we apply the Latent Dirichlet Algorithm(LDA) on lyrics from the Music4all dataset to make a topic model. The goal of a topic model is to extract several topics from given documents. A topic model can be used to classify a given document into one of the topics created by the model. Thus, topic models are very useful for text classification. In contrast to other researchers, we focus on the chorus/hook of a song. Our goal is to check whether or not it is possible to classify a song based on its chorus/hook. A positive correlation would open a path to other research and algorithms focusing on specific parts of songs

2 Previous Research

While topic modeling has been done already on song lyrics, there is still a lack of published papers. Different researchers have used different methods and datasets for their research. Florian Kleefdorfer(2008) published their paper "OH OH OH WHOAH! TOWARDS AUTOMATIC TOPIC DETECTION IN SONG LYRICS" revealing tags that were acquired based on topic modeling songs from Verisign's lyrics collection. Kleefdorfer had lyrics of 32323 songs after getting rid of short songs, stopwords, meaningless words, infrequent words, and words that appeared in more than 70 percent of the documents(Kleefdorfer,2008). They used the Non-negative Matrix Factorization (NMF) technique for topic modeling. To evaluate the model, manual labeling was done with 6

individuals who attempted to provide a topic tag to the topic clusters that were derived from NMF. Kleefdorfer shared the winning tags with at least a 10% likelihood which included 41 out of the 60 total tags (Kleefdorfer,2008).

In addition to Kleefdorfer, Lucas Sterckx(2013) also tried topic modeling on song lyrics in their paper "Topic Detection in a Million Songs". While their paper talks about many different approaches to topic modeling, they end up using LDA and L-LDA. LDA provided an unsupervised model and L-LDA provided a supervised model. They used the Million Songs Dataset and musiXmatch dataset to get song lyrics (Sterckx,2013). The data was pre-stemmed and preprocessed in a bag of words format. For the L-LDA, 24 supertopics from the Green Book of Songs(GOS) dataset were used for the supervised topic tags (Sterckx,2013). The topic models were evaluated using semantic coherence with Wordnet and a variation of the Kurtosis evaluation called Excess kurtosis. Semantic coherence was evaluated with metrics such as topic intrusion. The unsupervised topic model was evaluated and compared using supervised data through different evaluation metrics (Sterckx,2013).

In addition to NMF, LDA, and L-LDA, there are other topic modeling techniques. To prove that different models can acquire unique and possibly useful additional topics, Alen Lukic(2014), in their paper "A Comparison of Topic Modeling Approaches for a Comprehensive Corpus of Song Lyrics", used the Pachinko Allocation Model(PAM) to do topic modeling on 763491 songs collected from the SongMeanings lyrics dataset. Lemmatization was applied after the removal of stopwords and useless words and ultimately a bag of words model was used for the PAM(Lukic,2014). Lukic also created an LDA model to compare with the topics from the PAM. It was found that PAM is successful in finding more undiscovered topics than LDA. However, further analysis and evaluation of the data was not possible due to the lack of enough human annotation as only 500 songs were annotated out of 763,491 songs(Lukic,2014). Lukic shared labeled topics from the LDA model and unlabeled clusters from the PAM to show that the PAM managed to get additional topics that were undetected by the LDA model.

3 Our Method

Due to its popularity, accessibility, and impressive topics generated by Sterckx's(2013) research, we decided to use the Latent Dirichlet Algorithm(LDA) to create our topic model.

The goal of LDA is to sort a collection of documents into topics. A topic is defined as a distribution amongst the vocabulary of the entire corpus. The main idea behind the algorithm is that a single document exhibits multiple topics. In this algorithm, we start with the first Dirichlet distribution. This distribution will assign topics to documents. The topics are scattered in different proportions amongst each document. A document is randomly chosen from this distribution. The document's distribution of topics will determine the multinomial distribution of topics. Topics are randomly selected from this distribution. From here words are selected from the second Dirichlet distribution that assigns topics to words. Each word selected corresponds to the topic. The next multinomial distribution is then created by using the words associated with topics from the second Dirichlet distribution. A topic from the first multinomial distribution is then associated with a word from the second multinomial distribution that lines up with that topic. This process creates the document that the LDA model will produce.

Gensim is a popular NLP library in python. It provides functions to implement and evaluate an LDA model. We used the Gensim library in python 3.9 to make the LDA model.

4 Data

We looked at MusixMatch and Genius API initially to get lyrics. MusixMatch dataset offered the lyrics in a preprocessed bag of words format which made it impossible for us to get corresponding chorus data for the songs. We worked with Genius API for a while. Genius API provided the lyricsgenius python library which helped us collect songs from specific artists. Unfortunately, Genius provides labels for parts of songs that resulted in redundant words like "Chorus", "Intro" making it into the data. Furthermore, Genius had live versions, remixes, speech, poems, etc. which needed to be filtered out manually. We worked with the Genius API until we found the Musci4all dataset. We used lyrics from the Music4all dataset to make our topic model. The Music4all dataset provides data for 109269 songs

from 16269 unique artists. The songs are in 46 unique languages. The most common language is English (Santana et al, 2020). While other metadata and 853 unique genre tags were provided with the Music4all dataset, we were only interested in the lyrics for the songs (Santana et al, 2020). We separated a randomly selected subset of songs for testing and evaluation.

4.1 Data Selection

We were only interested in English songs with enough lyrics for the model. Therefore, we got rid of the songs that are less than only two lines long. This got rid of instrumentals and songs without enough lyrics for the model. We used the LangDetect library to detect the language of songs. LangDetect is a python library that provides functions to give a language probability to the given text. This helped us get rid of songs that are in languages other than English. After this step, we were left with 83,825 non-empty songs in English. Although we got rid of almost all songs in other languages, some lyrics are in other languages as they are half in English and half in a non-English language. We weren't able to remove them.

4.2 Data Processing

We used different functions from python's NLTK library to process the songs. Similar to Gensim, NLTK is a prominent python library that offers different functions for NLP. After reading the files, we tokenized each song into a bag of words format using a regex tokenizer created using the NLTK library. The regex tokenizer kept words with apostrophe(‘). This allowed us to keep words like “can't”, “i'm”, etc. Certain words had the “\n” character attached. We removed those characters from all words. An LDA model works the best when provided words are meaningful. Words that occur frequently and other meaningless words are not useful for us. Therefore, after case-folding, we decided to filter stopwords. We acquired the stopwords list from the NLTK library. We added some meaningless words like “ohh”, “mmm”, etc. to the stopwords list to filter them as well. Most words under two characters' length were useless so we decided to filter all the words that are less than length two. Furthermore, we removed numeric tokens as most of the time they don't have a meaning. After trial and error, we decided to use the NLTK library's WordNet lemmatizer to lemmatize the songs. We avoided stemming as popular stemmers often result in unwanted meaningless words.

4.3 Unigrams vs Bigrams

We initially created models using unigrams. Those models had many topics with noise words. Based on manual evaluations, those models didn't have many rich and meaningful topics. Therefore, we tried using bigrams. Bigrams yielded much better results. Topics were more meaningful. Thus, we decided to use bigrams for the model that we ended up using for evaluations. Bigrams were created using the Gensim library's phrases function. In addition to the corpora, the function required two parameters: min_count and threshold. The min_count is the number of times a bigram should appear in the corpora for it to be considered a valid bigram. The threshold parameter is also a number. Increasing the threshold results in a lower number of bigrams. After trial and error, we used 20 for min_count and 10 for threshold.

4.4 Dictionary and Corpus

In order to pass the bag of words to the Gensim library's LDA function, we had to create an id2word dictionary object and a corpus object.

Id2word is a dictionary with a unique id for each unique word mapped to the frequency of the word in the corpus. We created the id2word using the Gensim library's provided Dictionary function. Another function we used was the filter_extremes function. Similar to Kleefdorfer(2008), we got rid of words that appear in only 3 songs. Similarly, we also removed words that have a frequency greater than 0.6(60 percent documents).

Corpus object is created by applying id2word mapping to all the words. Corpus is a bag of words representation with each word having a unique id and a frequency mapping for how many times it appears in the song. Table 1 provides quantitative data about the corpus after preprocessing. Table 2 lists the top ten words in the corpus.

Data Type	**Number**
Total Songs	83825
Total Tokens	4716295
Average Tokens per Song	56.26358
Unique Tokens	36225
Type Token Ratio	0.00768

Table 1: Information about the corpus after data processing .

Top Ten Words	Frequency
i'm	193109
know	132592
love	130978
like	111262
time	79728
get	78237
got	74718
never	73085
one	72752
yeah	68439

Table 2: Top 10 words in the corpus after data processing.

5 Making the LDA model

We had to decide the optimal number of topics for the LDA model. In order to do that, we looked at two evaluation metrics: perplexity and coherence. We made 49 models of 2,3,4,5…50 topics and collected their perplexities and coherences. These numbers helped us make our decision about the number of topics we want.

5.1 Perplexity

Perplexity is the assessment of the model on how it reacts to unseen information. A lower perplexity means a better model. What makes a good perplexity differs for each dataset and task. The goal for us was to lower it as much as possible while having a reasonable number of topics to work with. Figure 1 shows the effect of the number of topics on perplexity. The seems to get better for more topics.

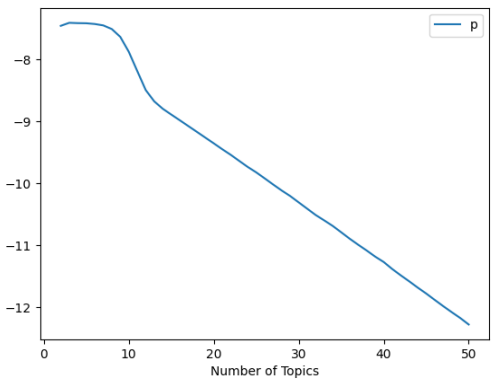

Figure 1: Perplexity over the number of topics.

5.2 Coherence Score

A coherence score is a semantic evaluation of words that score the highest in a given topic to see if the word really belongs in that particular topic that it's assigned to. The Gensim library provides its own functions to log these evaluation metrics. The coherence function takes different coherence parameters to do different kinds of semantic evaluations. The most accurate coherence parameter is the "c_v". In the paper "Conversational Structure Aware and Context-Sensitive Topic Model for Online Discussions", researchers Sun, Laparo, and Kolaicinski worked on different topic models and evaluation techniques. According to the researchers, the "c_v" parameter uses cosine similarity and normalized pointwise mutual information(NPMI) to evaluate the top words of each topic(Sun et al.,2020). While it is accurate, it is very slow. The fastest coherence parameter is the "u_mass". This parameter is based on the document "co-occurrence counts, and a logarithmic conditional probability as confirmation measure"(Sun et al.,2020). It is not as accurate as "c_v", but significantly faster. We used "u_mass" to check the coherence of the 50 models. Figure 2 shows the relationship between "u_mass" and the number of topics based on our data. The coherence seems to get worse as the number of topics increases since more topics mean more words getting distributed.

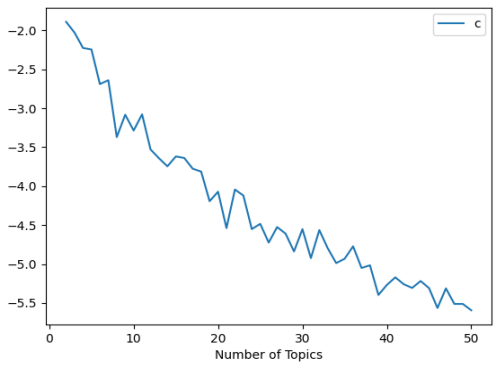

Figure 2: "u_mass" over the number of topics.

5.3 Number of Topics

Based on the graphs in Figure 1 and Figure 2, we decided to choose a model from 15 to 20 topics range. In addition to the perplexity and coherence scores, we also manually looked at the topics created by the models to decide whether or not the topics are meaningful. Based on our evaluation, we

decided to go with the 20 topics model. Although we based our model on these graphs, the model we made had different parameters than the ones used to make the graphs in Figure 1 and Figure 2.

5.4 Trial and Error

A lot of the numbers and decisions that decide how the model is made can depend on the data that is being used. We had to decide the optimal number of topics, the best way to tokenize the songs, frequency of words to keep, whether or not to apply lemmatization, bigrams vs unigrams, threshold and min_count for bigrams, what words are meaningful, and minimum words per song. A lot of these decisions have been based on trial and error. It is very likely that we can make a better model than the one that is used for evaluation.

6 Evaluation

The 20 topics model that we created has a perplexity of -8.12719. This is way better than the very first model we created which had a perplexity around -6. The "u_mass" coherence of this model is -2.73911 which is also better than earlier models that had a "u_mass" around -4.

6.1 Topics

To evaluate the quality of the topics, we manually looked at the words for each topic and decided whether or not the topics are meaningful. Meaningfulness is judged based on the relationships between words that are picked by the model. Based on our evaluation, all of the 20 topics have some meaning. Some topics have more noise words than others which makes them less meaningful. This is why it is very important to decide which words to keep and which words to get rid of during data processing. As expected, there are some overlapping topics as certain topics like love and heartbreak are more popular and related. Table 3 shows each topic with the top 15 words and Figure 3 shows the overlap between the topics. Figure 3 is created using the python library pyLDAvis which provides functions to visualize LDA models.

Topic	Meaning	Words
1	Yes	get, like, fuck, bitch, shit, everybody, got, hit, back, fucking, party, high, rock, damn, i'ma
2	Yes	night, light, eye, sun, fire, sky, see, star, dark, high, rain, blue, cold, morning, sea
3	Yes	watch, sound, half, three, mother, four, radio, ring, thank, family, called, five, seven, hero, doo_doo
4	Yes	head, around, hand, turn, put, face, room, back, house, cut, bed, body, foot, door, wall
5	Yes	ain't, ni**a, got, money, that's, keep, man, goin', gon', work, mama, nothin', bout, lookin', comin'
6	Yes	life, world, dream, live, without, new, sleep, living, save, today, wake, lonely, tomorrow, whole, dreaming
7	Yes	know, say, can't, way, see, tell, thing, think, cause, right, get, always, there's, nothing, something
8	Yes	come, we're, little, tonight, man, back, we'll, let's, well, old, he's, young, sing, ride, town
9	Yes	boy, dance, song, play, goodbye, mine, beat, music, friend, fine, dancing, summer, singing, hear, rhythm
10	Yes	could, would, said, better, never, thought, i'd, maybe, please, remember, still, wish, ever, knew, forget
11	Yes	black, fly, kill, white, red, gun, lay, people, ghost, king, dog, magic, shot, bone, wing
12	Yes	let, i'll, heart, keep, hold, inside, find, fall, mind, lost, break, still, end, waiting, can't
13	Yes	like, want, feel, make, girl, good, bad, real, free, happy, feeling, cause, doe, guy, need
14	Yes	i'm, gonna, i've, cause, going, coming, feeling, back, running, ready, getting, trying, looking, sorry, falling
15	Yes	soul, god, blood, die, death, dead, fear, born, war, lie, must, power, earth, child, pain
16	Yes	time, one, day, every, long, stop, last, cry, another, wait, forever, together, first, miss, two
17	Yes	love, give, need, kiss, enough, somebody, sweet, lover, loving, touch, heart, someone, true, heaven, darling

18	Yes	yeah, baby, got, gotta, hey, right, move, crazy, let, body, hey_hey, get, babe, night, honey
19	Yes	take, away, run, walk, far, road, throw, place, follow, air, walking, angel, somewhere, mile, lead, steal
20	Yes	never, wanna, home, alone, gone, stay, leave, call, name, anymore, die, phone, i'd_rather, another, anybody

Table 3: LDA Model Topics

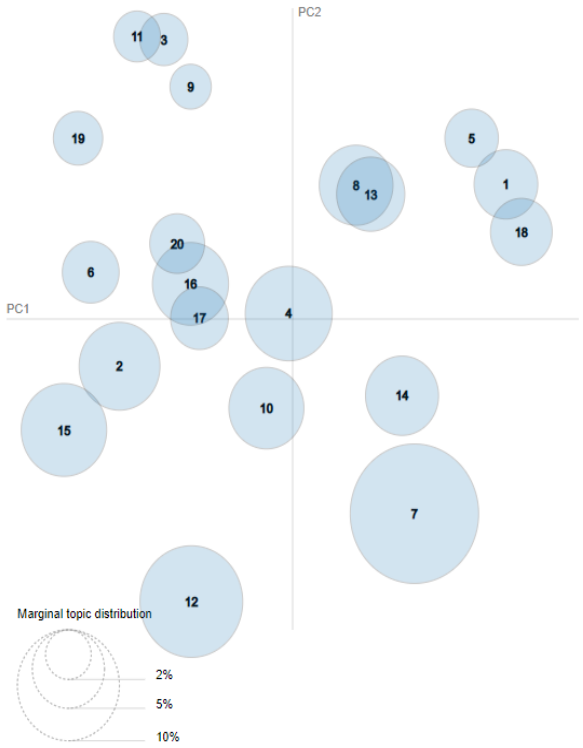

Figure 3: Topic Overlap. Each circle represents a topic. The size of the circle represents how common that topic is.

Based on Table 3 and Figure 3, the most popular topics are Topics 4, 7, 12 and 15. Topic 4 has words about human body parts. These words can be found in many different topics. For example, in a love song, an artist can be talking about their significant other's features. Topic 7 has words that can apply to many different topics as well. For example, a song about an artists past can have those words. Topic 12 has words that are mostly found in songs about heartbreak. Topic 15 has words that are mostly found in songs about religion.

6.2 Chorus and Full Song Evaluation

There are a few problems with choruses. Some songs don't have a chorus or hook. Many songs have very small choruses and hooks. Picking the accurate topic for those songs seems unlikely. For each song, its corresponding chorus/hook file had the chorus and hook as many times as it appeared in the song. This gave the model a bit more compared to just having it only once. The choruses and hooks of the songs were manually collected as we couldn't find a dataset for just the chorus/hook of songs. Therefore, we lacked the manpower to label and collect a significant amount of choruses for testing.

To check the accuracy of the model, we randomly separated 100 songs from the training data. Those songs were manually labeled with 3 likely topics from the ones that are created by the LDA model. We decided to go with three because only one person labeled them and the task of labeling songs with a topic is more difficult than it sounds! Those 3 labels were then compared to the top three topics picked by the model for each song and its corresponding chorus/hook. Thus, we had three sets of labels for each song with three labels in each set. One set was for chorus/hook, one set was for human labels, and the last set was for the full song. If any of the topic labels in those sets overlapped, we counted it as a successful match.

For example, if the human label for a song is topic 1,2,3 and the model picks topic 3,4,5 for the chorus; then the chorus agrees with the human as both picked topic 3. Similarly, we also checked if the chorus agreed with the full song, and if the full song agreed with the human labels.

7 Results

Based on the 100 test songs and choruses that we've collected, Table 4 shows the agreement probabilities between human labels, chorus topics, and full song topics. Table 5 shows the frequency of dominant topics on the test data.

Agreement	Probability
Full song – Human	0.45
Chorus – Human	0.37
Chorus – Full song	0.77
Chorus – Full song dominant topic	0.27

Table 4: Agreement results based on evaluation

Topic	Full	Chorus
1	0.05	0.03
2	0.09	0.07
3	0.03	0
4	0.06	0.12
5	0.02	0.06
6	0	0.01
7	0.07	0.03
8	0	0.02
9	0	0
10	0.01	0.03
11	0	0
12	0.03	0.13
13	0.02	0.03
14	0.01	0.03
15	0.6	0.36
16	0.02	0.02
17	0.01	0.02
18	0	0.01
19	0	0.01
20	0	0.02

Table 5: Frequency of the dominant topics in test data

8 Conclusion

We did not have enough chorus and labeled data to make any decisive conclusions yet. However, based on the 100 test songs and the LDA model we used, Table 4 shows that the chorus/hook agrees with the full song with a probability of 0.77 which means out of the 100 songs, 77 songs had overlapping top three topics picked by the model for both chorus/hook and full songs. In addition to that, Table 5 shows that most of the topics picked from the full songs and the choruses/hooks have similar frequencies. In both cases, the most popular pick is topic 15. However, the dominant topics picked by the model only agree 27 out of 100 times which could suggest there are frequent general words that are increasing the probability of certain topics. We can improve the stopwords list to fix it.

The chorus/hook agrees with the human labels with a probability of .37 which is less than 0.45, the full song – human agreement. This suggests that the LDA model has better accuracy in picking topics from full songs compared to just the chorus/hook. This makes sense because some choruses/hooks are very short and meaningless. However, 0.37 is still pretty close to 0.45. The human agreement in both cases is very low which means the LDA model we made can be further improved. Another possibility could be that using only one annotator resulted in biased/inaccurate song labels. However, 77 percent agreement between the top three chorus/hook and full song topics picked by the model suggests that there is a correlation between the topic in the full song and the chorus.

9 Future work

With a human agreement of 0.37 and 0.45, we believe that a better model can be built using different parameters. As stated earlier, building the best model for a particular dataset involves trial and error. Changing one or two parameters slightly can have drastic effects on the model performance. We plan to gather more choruses and test data to further evaluate our LDA model. Furthermore, we plan to use multiple annotators in the future to label the test data. We used “u_mass” to decide the number of topics for the model. We plan to use “c_v” coherence in the future. Although it takes significantly longer to get the “c_v”, it is more accurate compared to “u_mass”. Once we further verify the correlation we noticed between the chorus and the full song, we will try training a model using just the chorus of songs. Less text to train would significantly reduce the time it takes to make the model.

If we manage to find a stronger correlation between the chorus and the full song, we plan to make a music recommender system based on user-provided choruses. We plan to apply our topic model to different artists and genres to gather data on music trends. LDA is just one topic modeling technique and Music4all is just one dataset. We plan to explore other topic models and compare their performance on different datasets. Relying on just the lyrics can limit our research to only songs with lyrics. Therefore, we believe that combining the acoustic features and lyrics data can yield the best topic model for songs.

References

Alen Lukic. 2014. *A Comparison of Topic Modeling Approaches for a Comprehensive Corpus of Song Lyrics*. http://alenlukic.com/assets/docs/lyric_topic_modeling.pdf.

Florian Kleedorfer. 2008. *OH OH OH WHOAH! TOWARDS AUTOMATIC TOPIC DETECTION IN*

SONG LYRICS. https://archives.ismir.net/ismir2008/paper/000211.pdf.

Igor André Pegoraro Santana, Fabio Pinhelli, Juliano Donini, Leonardo Catharin, Rafael Biazus Mangolin, Yandre Maldonado e Gomes da Costa, Valéria Delisandra Feltrim and Marcos Aurélio Domingues. Music4All: A New Music Database and its Applications. In: 27th International Conference on Systems, *Signals and Image Processing (IWSSIP 2020)*, 2020, Niterói, Brazil. p. 1-6.

Lucas Sterckx. 2013. *Topic Detection in a Million Songs*. https://libstore.ugent.be/fulltxt/RUG01/002/033/229/RUG01-002033229_2013_0001_AC.pdf.

Markus Schedl, Emilia Gómez and Julián Urbano. 2014. Music Information Retrieval: Recent Developments and Applications. *Foundations and Trends® in Information Retrieval*, *Vol. 8*, No. 2-3, pp 127-261. https://dx.doi.org/10.1561/1500000042.

Yingcheng Sun, Kenneth Loparo and Richard Kolaicinski. 2020. Conversational Structure Aware and Context Sensitive Topic Model for Online Discussions. *Proceedings of the 14th IEEE international conference on semantic computing (ICSC)*. https://doi.org/10.1109/icsc.2020.00019.

Contributions

Sadab Hafiz: Coding, Testing, Abstract, Introduction, Previous Research, Data, Results, Evaluation, Conclusion, Future Work.

Zachary Motassim: Research, Chorus gathering, Song labeling, Abstract, Introduction, Previous Research, Method, Future Work.

The overlapping sections represent parts where we worked together.