**Data**

We collected data from Mendeley Data, which provides a list of lyrics from 1950 to 2019 describing music metadata as sadness, danceability, loudness, acousticness, etc., and also other information including artist name, music genre, and so on [1]. We selected both genre and lyrics from the original dataset and did natural language processing to clean the data. We deleted all non-English words in the lyrics, removed punctuation and double spaces, and changed all words to lower case in order to better encode them in our models. Since words can be grouped together based on their lemma and be analyzed as a single item or token, we lemmatized words in lyrics using the WordNet's built-in morphy function. In addition, words such as ‘the’, ‘a’, ‘in’ provide no useful information in lyric classification. Therefore, we used ‘stopwords’ packages from the NLTK library to remove all stop words in English. After data pre-processing, our final dataset contains 28,372 lyrics in seven genres: blues, country, hip hop, jazz, pop, reggae, and rock. For model training and testing, we split the training and testing data sets in an 80/20 ratio.

**Generative Model**

Describe the model in detail and develop a solution using parameter inference (and/or decoding).

**Apply to Real Data:** Highlight situations where each approach performs well and poorly.

We chose Latent Dirichlet Allocation (LDA) as our generative model and used the Latent Dirichlet Allocation with the online variational Bayes algorithm package from scikit-learn [2][3] to classify seven music genres. Since LDA is a type of unsupervised machine learning, we can only specify how many topics we want to find, which is seven. The seven classified topics that were given by the model were not necessarily the same as our original genres. After fitting the data into the model, we printed the top 50 words in each topic and used the pyLDAvis package to visualize the model results [4]. Table 1 shows the top 50 words in each topic. As shown in Figure 1, there are some overlaps between Topic1 and Topic2, and Topic0 and Topic4. Since for each document there is a topic distribution, we then used the argmax function on each document’s topic distribution to assign each document to a topic. In order to map the topics that were modeled by LDA to our original genres, we found the top 50 words in each original genre, computed the number of overlap words between each genre and each topic, and mapped each topic to the genre with the most overlapping words.

***Table 1:*** *Top 50 Words in Each Topic*

|  |  |
| --- | --- |
| **Topic** | **Top 50 Words** |
| 0 | like, fuck, bitch, shit, money, know, yeah, real, cause, smoke, need, wanna, time, tell, world, high, look, come, think, want, girl, feel, right, people, damn, life, talk, woman, gotta, fuckin, roll, stay, work, sell, check, hard, drink, city, bout, go, watch, face, pussy, fake, catch, game, hoe, live, ride, straight |
| 1 | heart, baby, hold, night, know, long, tonight, right, leave, kiss, like, believe, feel, want, come, sweet, time, stay, need, eye, go, dream, arm, wait, tell, girl, hand, cause, love, true, touch, break, close, darling, tear, yeah, promise, little, woman, start, apart, fall, wanna, look, dear, light, tight, lips, think, till |
| 2 | away, go, good, break, know, walk, lonely, home, gonna, feel, leave, come, heart, miss, fool, tell, tear, yeah, stand, wish, little, goodbye, cry, baby, morning, rain, look, cause, think, say, like, want, whoa, fade, stay, night, take, sleep, hurt, girl, place, time, dream, days, hide, belong, memory, yesterday, wonder, pain |
| 3 | fall, feel, come, inside, eye, like, head, know, dead, cold, blood, black, stand, burn, soul, face, leave, hand, fear, hear, lose, fight, pain, lie, death, wall, turn, hell, hide, live, save, kill, grind, devil, die, light, life, follow, tear, break, deep, word, mind, wind, world, look, call, alive, breathe, speak |
| 4 | like, yeah, know, better, cause, come, gonna, play, want, party, shoot, tell, wanna, gotta, right, head, bout, night, say, stop, start, talk, little, good, think, get, everybody, game, kick, baby, watch, girl, hear, look, ready, beat, turn, hand, goin, sound, go, drop, sick, work, black, feel, damn, throw, time, shake |
| 5 | sing, song, blue, hear, right, come, music, home, play, remember, hand, go, ring, bring, lord, like, sweet, songs, long, summer, kill, word, money, listen, people, say, white, write, roll, days, dance, woman, night, call, fight, land, time, know, little, yeah, guitar, band, moon, sound, swing, head, christmas, black, light, year |
| 6 | time, life, live, know, world, mind, change, come, think, want, need, things, cause, go, dream, look, lose, like, feel, give, today, love, leave, yeah, learn, hard, forget, long, days, take, tell, right, somebody, true, wanna, remember, tomorrow, turn, better, reason, gonna, believe, ready, thing, forever, matter, grow, make, hurt, years |

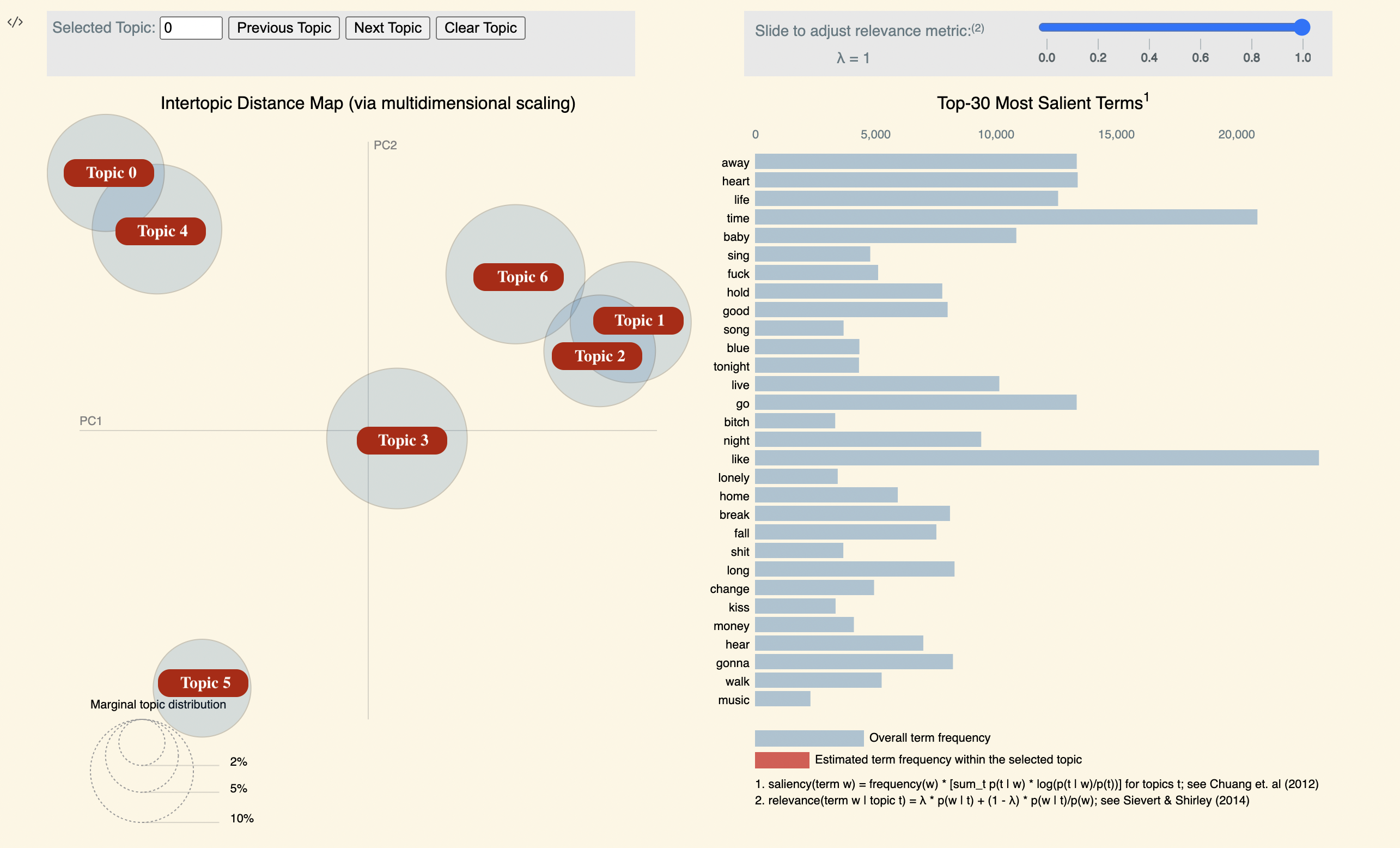
***Figure 1:*** *Visualization using pyLDAvis*

Table 2 shows the mapping results of topics to genres as well as the result of applying our LDA model to real data, which achieved an overall correctness of 22.68% with relatively higher correctness on rock and hip-hop and relatively lower correctness on reggae.

***Table 2:*** *Mapping Topics to Genre & Real Data Results*

|  |  |  |
| --- | --- | --- |
| **Topic** | **Genre** | **Correctness** |
| 0 | Hip-hop | 0.4688 |
| 1 | Jazz | 0.1692 |
| 2 | Country | 0.1650 |
| 3 | Rock | 0.3502 |
| 4 | Blues | 0.1370 |
| 5 | Reggae | 0.0888 |
| 6 | Pop | 0.2084 |

**Synthetic Data**

Evaluate the results qualitatively and quantitatively. Highlight situations where each approach performs well and poorly.

We used the LDA model that we trained to generate synthetic data. For each genre, we generated 4,000 lyrics so that the size of the synthetic data is similar to the size of the real data. The length of each lyric is randomly chosen from the length of lyrics in the real data[5], and the words in each lyric are chosen based on the distribution of words for each topic.

Table 3 shows the samples of generated data for each genre. Table 4 shows the results of applying our LDA model to the synthetic data that is generated by itself. Since the generated data meets all the assumptions of our LDA model, our model achieved correctness above 99% across all genres.

***Table 3:*** *Synthetic Lyrics for Each Genre*

|  |  |  |
| --- | --- | --- |
| **Topic** | **Genre** | **Synthetic Lyrics** |
| 0 | Hip-hop | movies machine phone undergo fuck girl wild cash repeat try fuck smoke real powder business high world bitch ones book week hair time disgrace classroom fuck crime counter exact chain extra mouth rhyme |
| 1 | Jazz | bugger different time tear remind stop straight spell smart hand love believe moment heart hold hand sweet close stay mechanism tell heart baby wait mean deep |
| 2 | Country | heart throne bring away basketball float home hearts miss heart lucid home leave go yeah wrong gonna cry away bring home away wish home away stand piece memory break lose lonely away walk change slip home get avenue make |
| 3 | Rock | trail white burn power side grave dress fight dead try sit take say fly pull point hide vein fall come death choose hear wall wide face need fall fall expose toll misery thoughts color years quick star bleed break room feast push |
| 4 | Blues | beat fool shit shots round steady wreck like gonna know sympathy bull want casually tell stand like mouth stop roll hell relax mister beat recognize come yeah know about radio doubt airport look leave home heat |
| 5 | Reggae | tower write right say know music soon say dance roll country rest sing sing blue dance blue ring blue dish magic song praise blue emergency heroes think yeah right tell halt song note beer play hand detroit write people crazy |
| 6 | Pop | soul need live live come listen child rest yesterday wanna need nerve things make look butterfly wrong wanna travel apart turn say dream life slip right say sundays wish want wear baby make need |

***Table 4:*** *Synthetic Data Results*

|  |  |  |
| --- | --- | --- |
| **Topic** | **Genre** | **Correctness** |
| 0 | Hip-hop | 0.9994 |
| 1 | Jazz | 0.9994 |
| 2 | Country | 0.9994 |
| 3 | Rock | 0.9994 |
| 4 | Blues | 0.9982 |
| 5 | Reggae | 0.9991 |
| 6 | Pop | 0.9994 |

**Discussion**

Discuss pros and cons of the two approaches. Consider:

• quality/correctness  
• data, time, and computational requirements

• interpretability

The "bag of-words" assumption of the LDA model lost information about the order of the words when trying to classify lyrics. Also, the LDA model performs worse at capturing rare topics due to data quality issues. Due to an increase in observations, our LDA model can be improved by training with a larger dataset. With the LDA, documents are represented as mixtures over latent topics, but when we assigned topic labels to lyrics, we simplified by choosing the topic with the highest probability, and mapping the pre-annotated genre of each lyric to a topic based on the number of overlapping words. A more reasonable approach can be taken to further improve the model's performance. During the process of generating data, since we randomly chose words based on the distribution of words for each topic, as we can see from Table 3, the synthesized lyrics are simply accumulations of words instead of meaningful sentences.

Table 5 shows the result of the two models on both the real and synthetic data. Compared to our discriminative model, the LDA model performed worse on the real data but better on the synthetic data that was generated by itself.

***Table 5:*** *Overall Correctness*

|  |  |  |
| --- | --- | --- |
| **Model** | **Correctness**  **(Real Data)** | **Correctness**  **(Synthetic Data)** |
| LDA | 0.2268 | 0.9992 |
| BERT |  |  |

The LDA model has fewer computational requirements as it can be trained locally within 10 seconds. With the algorithm and related distribution of a LDA model, it is more interpretable.

**Reference**

[1] Moura, Luan; Fontelles, Emanuel; Sampaio, Vinicius; França, Mardônio (2020), “Music Dataset: Lyrics and Metadata from 1950 to 2019”, Mendeley Data, V2, doi: 10.17632/3t9vbwxgr5.2

[2]“Online Learning for Latent Dirichlet Allocation”, Matthew D. Hoffman, David M. Blei, Francis Bach, 2010 <https://github.com/blei-lab/onlineldavb>

[3] “Stochastic Variational Inference”, Matthew D. Hoffman, David M. Blei, Chong Wang, John Paisley, 2013

[4] github repo: <https://github.com/bmabey/pyLDAvis>

[5] github repo: <https://github.com/dai-anna/Duke-NLP-FinalProject>