Topic Mapping of Song Lyrics Using Lda2vec

Megan Ardren, Spencer Dooley, Carter Ward

1. Introduction

It is an understatement to say there is an abundance of music in the world right now. Music is a product where more and more is created each day, and it is also something that true scientific analysis has been difficult to apply to. Between the vast amount of music and the inherent difficulty of interpreting it using scientific analysis, it can be difficult to visualize relationships between songs, artists, genres, and their meaning. Our goal with this is to show these relationships between these entities by using the machine learning concept of topic modeling.

A hypothesis we have, is that given the inherent complexity of song lyrics, they will be rather hard to form clear topics. To investigate this hypothesis we are going to also run the lda2vec algorithm on a dataset of news articles. Obviously news articles contain very topic-focused writing, so it will be interesting to explore the comparison between that and song lyrics which are much more "unfocused". We are going to use a topic modeling algorithm, lda2vec, which incorporates Latent Dirichlet Allocation (LDA) and word2vec. It is our goal to use lda2vec to analyze what topics and terms are most prevalent in our favorite songs, genres, discographies, and time periods.

Thus far, there have been works done using LDA for topic modeling of songs. Our goal is to use a newer, more deep-learning-based algorithm, lda2vec which is LDA incorporating word2vec, to make more presumably accurate and more interpretable topic models. From the final model, we will be able to calculate a coherence score which is a quantitative measure of how "accurate" our topics are as well as a visualization on a dimensionally reduced plane of how similar topics are to one another. To compare the effect the newer, deep-learning based algorithm lda2vec has on pure LDA, we will compare the coherence scores resulting from both algorithms and see if one produces more coherent topics.

We hope to learn if the change from LDA to LDA2vec results in an increase in accuracy and interpretability of results that warrants the increase in complexity and required training resources. We also hope to discern if topic modeling song lyrics is an inherently more difficult task than more topic focused types of text because of how songs are written.

2. Problem Definition and Algorithm

2.1 Task Definition

As alluded to in the introduction, there are currently a huge number of songs with that number only getting bigger. Given how universally popular music is, the demand for music

recommendations and personalized playlists is at an all-time high. With the results of our project, we can help advance these predictive algorithms used to suggest music, as our analysis will give a clear visualization of how similar songs are, topic-wise. Our input data set consists of about 10,000 songs. We ran the model using solely LDA first, then afterwards compared the effect word2vec has on it. For LDA specifically, our input data will consist of the text of the songs from the database mentioned previously.

We also built a topic model on a dataset of text from news articles to compare the effectiveness of topic modelling at the simplest level. This provided us an initial look at the differences between modeling news articles, which we assume to be topic-focused texts, and song lyrics which we assume to be non-topic-focused texts.

We then moved on to training LDA2vec models. Lda2vec takes the same text data LDA uses, except now a context vector will also be taken into account. This context vector is what makes lda2vec an improvement over pure LDA as it incorporates word2vec which models word-to-word relationships (local context). We got a list of interpretable topics and a list of terms which describe what the topic is. Essentially, we have a mapping of topics and to songs and then clusters of words describing these topics. From this, we will have a visualization of how "close" each topic is to each other using two principal components.

We then compared the LDA models to the LDA2vec models for both articles and song lyrics to see if there is an improvement in interpretability of the topics and coherence. Lastly, we compared the LDA2vec models trained on lyrics and articles. This analysis provided insight on if the difficulty in modeling topic-focused and non-topic-focused texts changes from the LDA to LDA2vec algorithm.

2.2 Algorithm Definition

The algorithm we used for the final topic model is lda2vec, which combines LDA and word2vec. These are two separate NLP algorithms that have been formed into one with the hopes of making topics more interpretable. LDA is a common unsupervised topic modeling algorithm that finds the distribution of topics a document belongs to based on the words in it. The algorithm makes a few key assumptions: there is a set number of topics across all documents (this is a hyperparameter in the lda2vec algorithm we will have to experiment with), each document is simply a bag of words (meaning order and grammatical context don't matter), and all topic assignments except for the current word are correct.

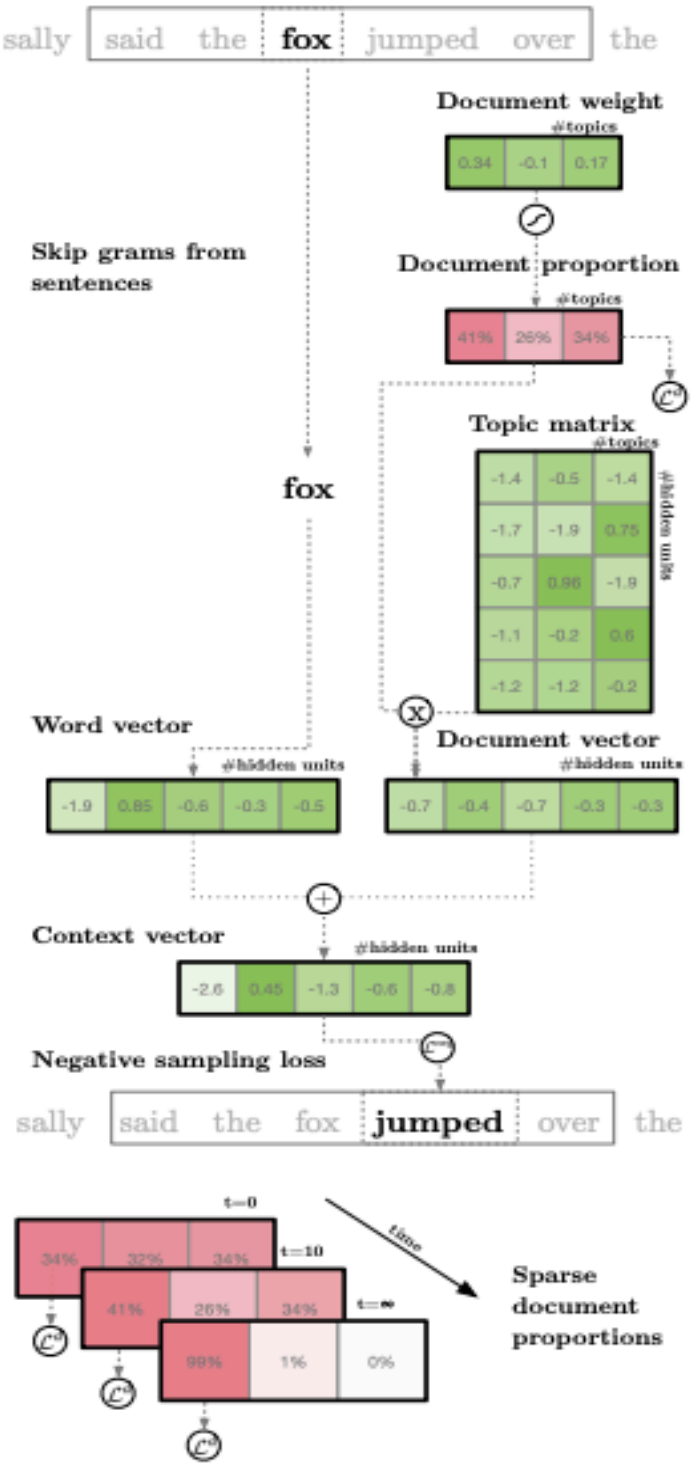
An overview of how the algorithm works is it starts with going through each document, randomly assigning each word to one of k pre-determined topics. Essentially by doing that, we took a document-term matrix for the corpus and converted it into two lower dimensional matrices (a document-topics matrix, and the other a topic-terms matrix). Then for each document, compute for each word the proportion *p1* of words in document *d* which are assigned

to topic *t,* and the proportion *p2* of assignments to topic *t* across all documents that come from word *w*. Then the current topic-word assignment is updated with a new topic probability which is the product of *p1* and *p2*. Essentially this is the probability topic *t* generated word *w*. After a large number of iterations, we reach a convergence point where the document-topic and topic-term distributions agree.

While LDA learns a document vector and then predicts the words inside the document, word2vec predicts neighboring words given a pivot word. Given a sentence, word2vec uses the pivot word to predict the surrounding context words. By training the model on our corpus, we will be able to obtain word embeddings that give vector representation to words. Word2vec is what allows us to essentially do math on words. While the LDA dimensions correspond to topics, these word vectors will simply be vectors of rather uninterpretable numbers representing the word in vector space that provide information about the words that are likely to surround it.

Implementing lda2vec from LDA and word2vec starts with generating a set of word embeddings (using the word2vec algorithm) and the document vector. The document vector is a weighted combination of the document weight vector (the weights of each topic in a document) and the topic matrix (represents each topic and its corresponding vector embedding). Lda2vec sums the document vector and the word vector to create context vectors for each word in the document. That is to say that lda2vec learns topic and document representations in addition to word and context vector embeddings. Consequently, the end-result of lda2vec is a set of sparse document weight vectors and easily interpretable topic vectors. A high level diagram of this algorithm can be seen below.

Figure 1: Diagram of LDA2vec algorithm (Moody 2006)



3. Experimental Evaluation

3.1 Methodology

To evaluate the performance of our topic model we had to assess how interpretable the topics created by the model are. Interpretability can be difficult to evaluate quantitatively because it is human judgement. The resulting topics that are interpretable will have to be manually assessed. The results that will be presented later in this report were trained on a corpus of around 10,000 songs from a variety of artists, genres, and time periods.

There are some metrics that evaluate topic interpretability such as coherence score, which assesses how coherent the words in each topic are. For our project we chose to focus on the extrinsic measure of coherence score (c_v) because of the interpretability of its results being on a 0-1 scale. The algorithm measures how similar the high scoring words in each topic are to each other using normalized pointwise mutual information and cosine similarity (Kapadia, 2019). We used coherence scores to optimize the number of topics by calculating it for each model with k topics and picking the model with the number of topics that have the maximum coherence score. Coherence score is not a perfect metric to base a model off of, as it is just telling us how related each word in the topic is to one another, but it is one of the few quantitative metrics accuracy of topic models.

We hypothesized that the topics created from our corpus of song lyrics will be less coherent than other corpuses such as news, literature, etc. This is because songs themselves often do not have as defined topics as pieces of written work. To test this hypothesis we trained lda2vec on a corpus of around 10,000 news articles. We compared the coherence scores from the resulting topic model trained on news articles to the model trained on song lyrics to see if there is any significant difference. Visually, the topic model can be presented on a simple xy-plane of the first 2 principal components. While there were no specific metrics to look at here, visualizing the topics shows the words in the topic and how much each topic overlapped, which shows how “different” topics are, and the most important words to each topic which provides a sense of interpretability. We use this “eye-test” to compare the results of LDA versus lda2vec and the results of song lyrics versus news articles.

3.2 Results

Figure 2: Coherence Score by Number of Topics for Song Lyrics

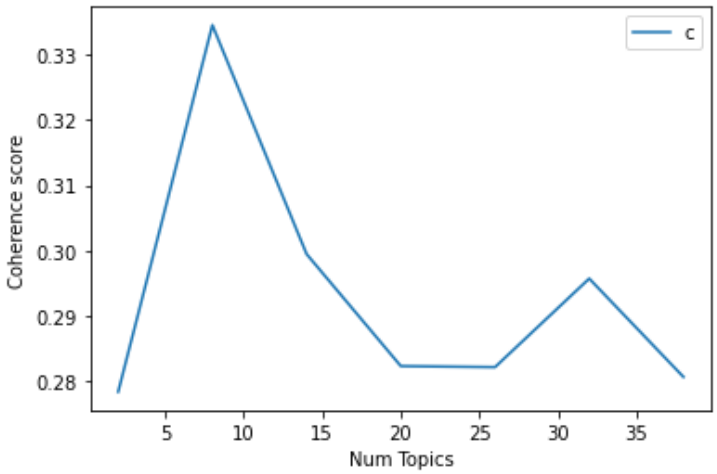


Figure 3: Coherence Score by Number of Topics for Articles

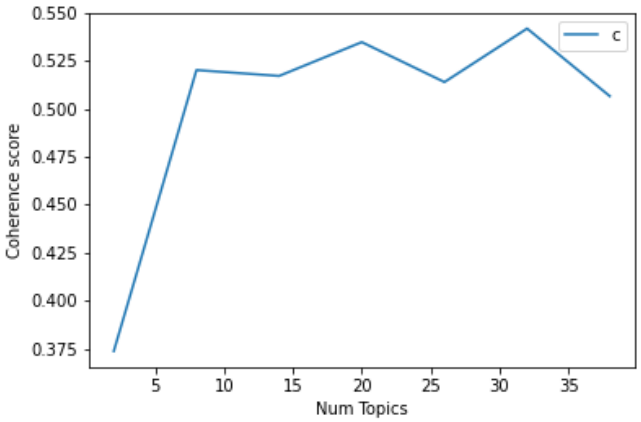


Table 1: Model Coherence Scores

Model	Coherence Score
Song Lyrics LDA	0.354
Articles LDA	0.588
Song Lyrics LDA2vec	0.373
Articles LDA2vec	0.59

Figure 4: LDA Visualization of Topic by Two PCs for Song Lyrics
Link to the HTML that renders the dashboard that can be downloaded and explored
https://github.com/carterwward/Topic-Mapping/blob/main/plots/large_lyrics_lda_.html

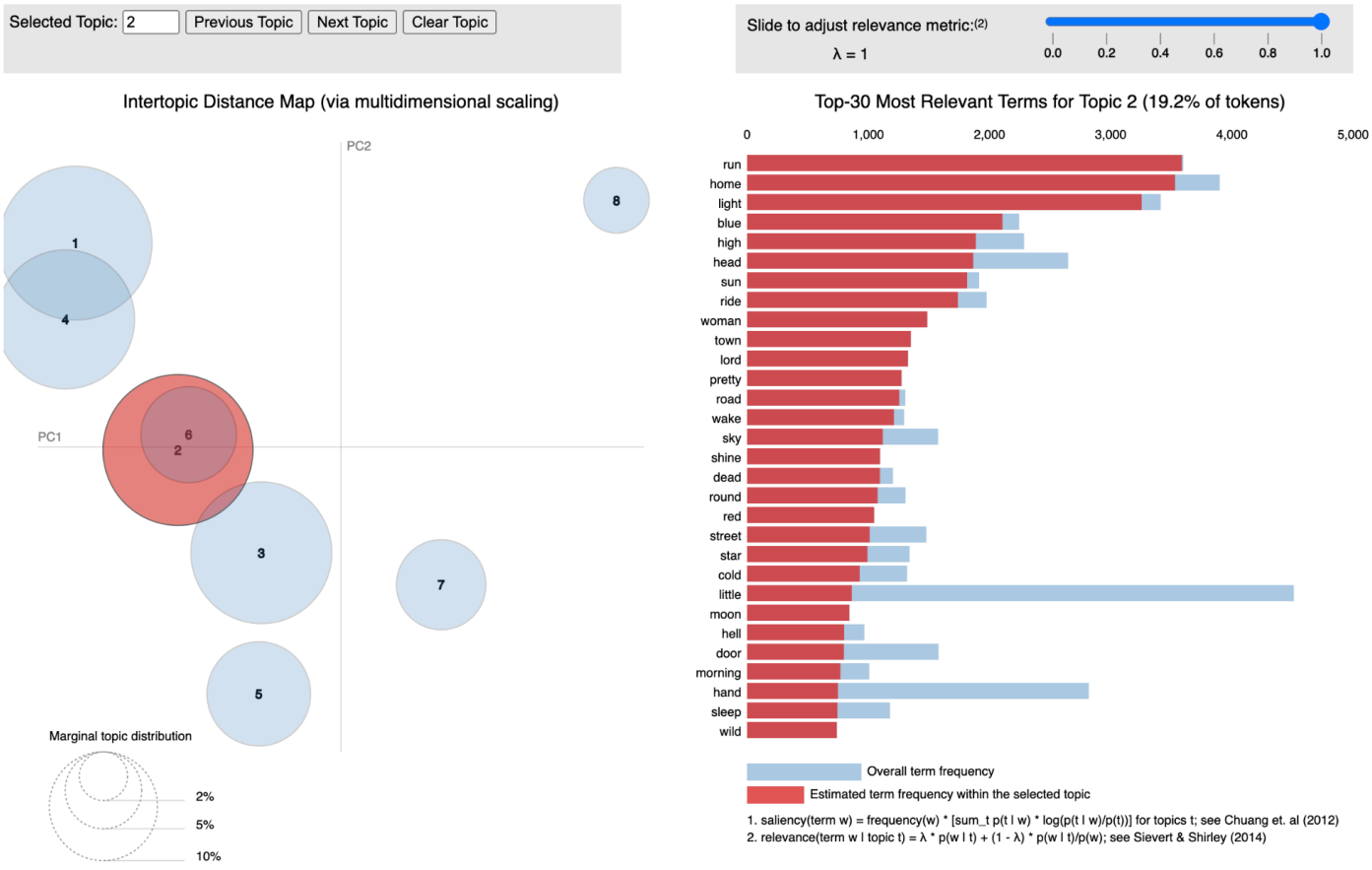


Table 2: LDA Article Topics (not all of them)

Topic #	Most important Words
Topic 1	World, dream, live, mind, fall, eye, die, lose, change, stand
Topic 7	Big, party, shake, people, jump, fun, city, eat, check, funk
Topic 8	Roll, na, ready, rock, hot, smoke, burn, dog, beat, heat

Figure 5: LDA Visualization of Topic by Two PCs for Articles
Link to the HTML that renders the dashboard that can be downloaded and explored
https://github.com/carterwward/Topic-Mapping/blob/main/plots/large_articles_lda.html

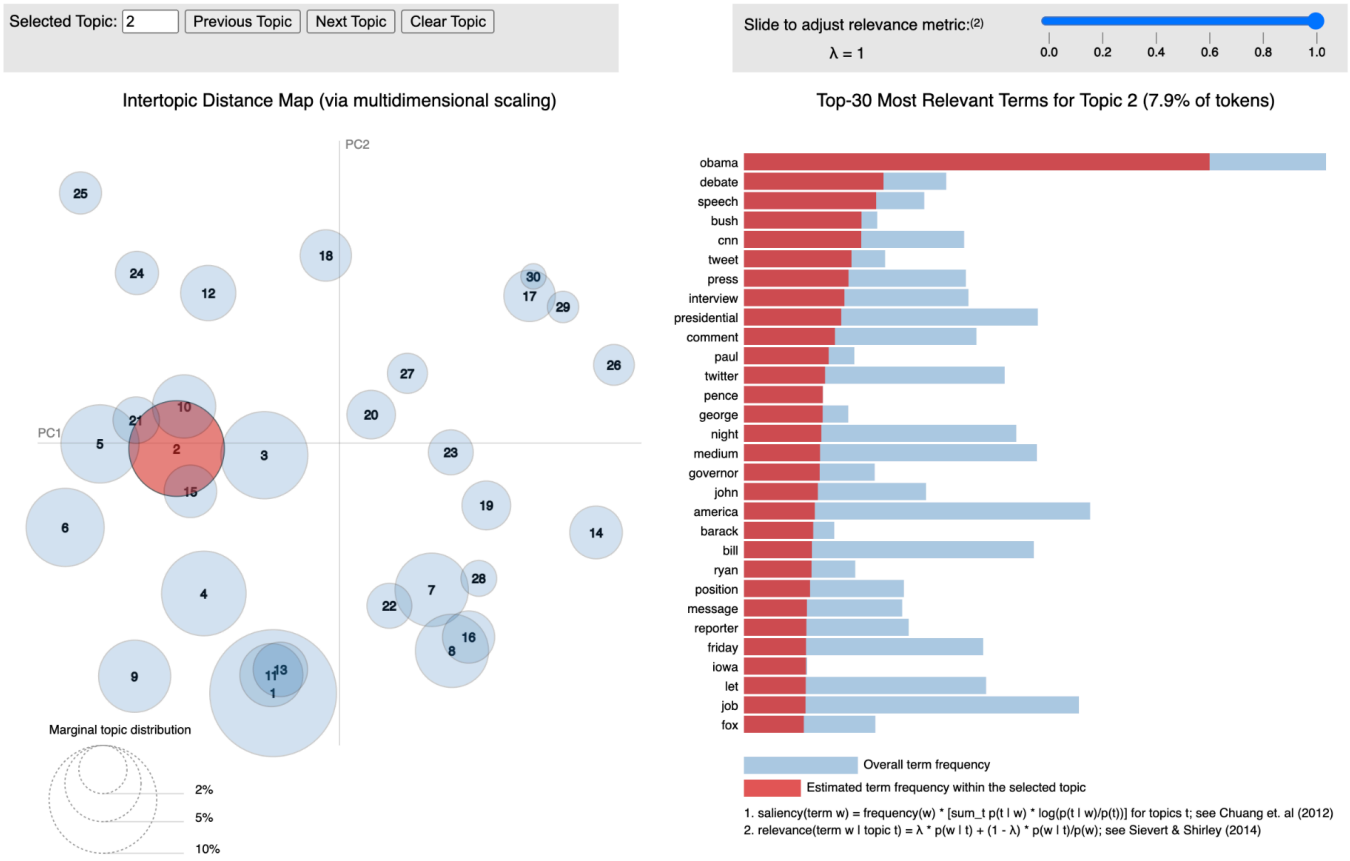


Table 3: LDA Article Topics (not all of them)

Topic #	Most important Words
Topic 3	Attack, police, kill, authority, security, fire, video, suspect, officer, arrest
Topic 12	Court, judge, justice, federal, attorney, refugee, lawyer, charge, prison, supreme
Topic 19	Game, team, play, season, second, episode, app, fan, album, coach

Figure 6: LDA2vec Visualization of Topic by Two PCs for Song Lyrics
Link to the HTML that renders the dashboard that can be downloaded and explored
https://github.com/carterwward/Topic-Mapping/blob/main/plots/lyrics_lda2vec.html

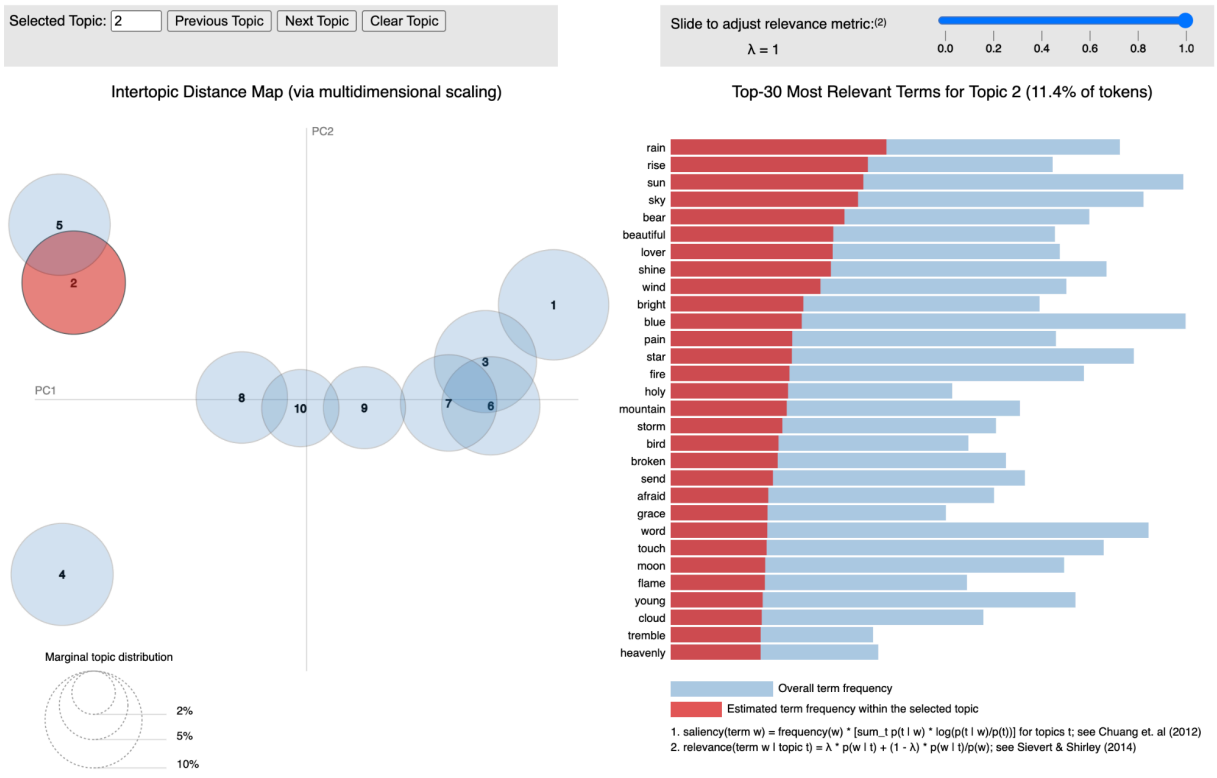


Table 4: LDA2vec Lyrics Topics (not all of them)

Topic #	Most important Words
Topic 2	Rain, rise, sun, sky, bear, beautiful, lover, shine, wind, bright
Topic 4	Goodbye, tear, honey, road, memory, wrong, apart, kiss, miss, ride
Topic 5	Joy, happy, jesus, sweet, water, sing, lord, christmas, child, smile

Figure 7: LDA2vec Visualization of Topic by Two PCs for Articles
Link to the HTML that renders the dashboard that can be downloaded and explored
https://github.com/carterwward/Topic-Mapping/blob/main/plots/articles_lda2vec.html

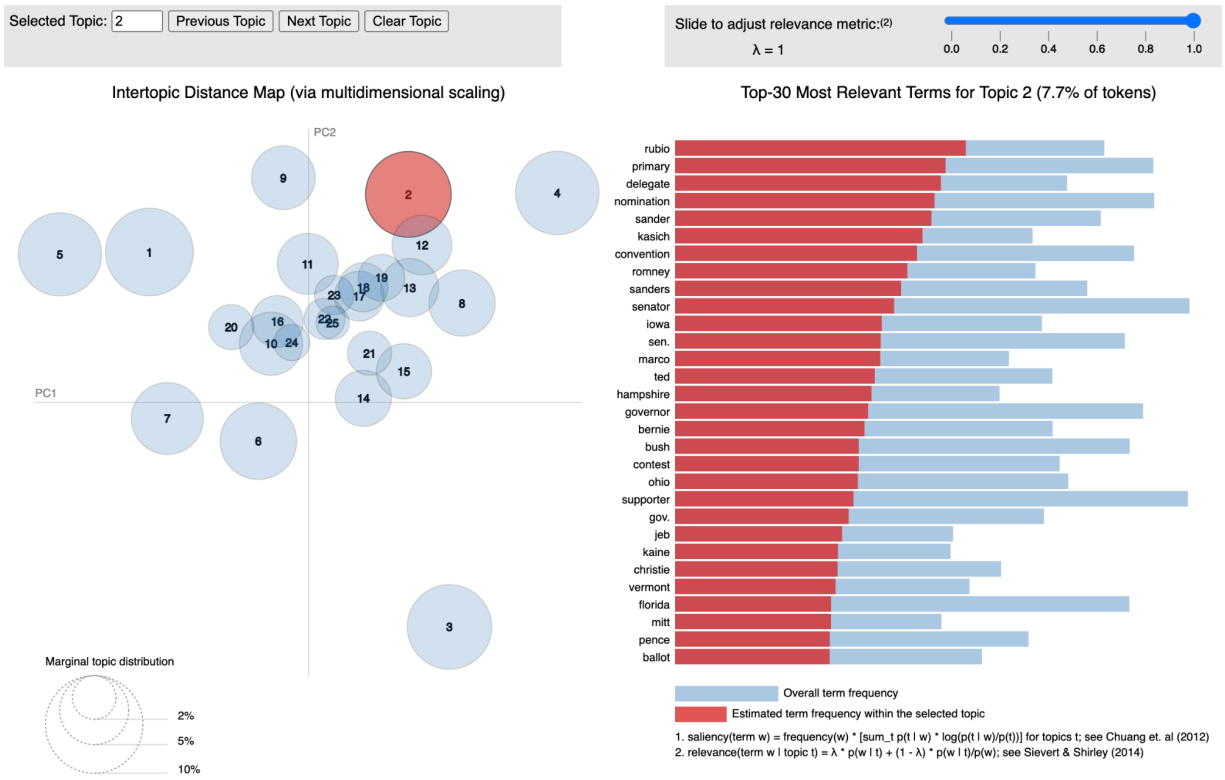


Table 5: LDA2vec Articles Topics (not all of them)

Topic #	Most important Words
2	Rubio, primary, delegate, nomination, sander, kasich, convention, romney, sanders, iowa
8	fbi, comey, intelligence, server, flynn, classified, nunes, leak, probe, hack
15	Eu, european, britain, brexit, germany, europe, merkel, labour, referendum, migrant

3.3 Discussion

For the LDA topic model of song lyrics we see in Figure 2, there was a maximum coherence score of 0.354 at 8 topics. While the plot in Figure 1 was very useful in seeing what topics maximized our coherence score, it also showed us that our initial model had a relatively low ceiling in terms of performance. A coherence score of 0.354 indicates poor performance as it means there was little continuity in the keywords in the documents that were grouped in each topic. Looking at the plot in Figure 4 we see exactly how this translates to the interpretability of the topics. The dashboard in the screenshot has Topic 2 loaded and the chart to the right of the graph shows the most important words to that topic. The top five most important words are run, home, light, blue, and high. These words do not give us a clear idea of what the theme of this topic is, hence the low coherence score. We saw that same trend of an unclear theme across most of the topics. There were at least two clear topics, however, which were songs focused on love and partying.

For the LDA topic model of news articles, we see in Figure 2 that the maximum coherence score of 0.588 at 30 topics. A coherence score of 0.588 indicates moderate performance and is significantly better than the coherence score for the LDA model for lyrics. Looking at the plot in Figure 5 we see exactly how this translates to the interpretability of the topics. The dashboard in the screenshot has Topic 2 loaded and the chart to the right of the graph shows the most important words to that topic. The top five most important words are obama, debate, speech, bush, and cnn. These words give us a much clearer idea of what the theme of this topic is (politics), which reflects the higher coherence score. We saw that same trend of much more interpretable themes across most of the topics.

Given the coherence scores for each model and the interpretability of the topics in the visualizations, there is evidence that modeling topics in song lyrics is more difficult than modeling topics in news articles. Not only was the coherence score for the news article model higher, but we found a higher number of interpretable topics in that model compared to the lyrics model. In the article we can identify a number of topics such as the police topic (Topic #3), judicial topic (Topic #12), the entertainment topic (Topic #19), and some others (Table 3). In the lyrics model we cannot identify any topic easily (Table 2). This supports our hypothesis that news articles are topic-focused text and song lyrics are non-topic-focused texts.

By implementing the lda2vec topic model we were able to raise the coherence score for our song lyrics corpus to 0.373 as can be seen in Table 1. A coherence score of 0.373 is better than 0.354 from LDA, but not by much. However, we do see an increase in interpretability of topics, which we can see in Figure 4. The dashboard in the screenshot has Topic 2 loaded and the chart to the right of the graph shows the most important words to that topic. As seen in Table 4, it seems we have topics for nature (Topic 2), breakups (Topic 4), and religion (Topic 5). While the labels for some of these topics are debatable and others are unclear, we still see a significant

increase in the interpretability of topics compared to those given by LDA (Table 2). In the case of song lyrics there seems to be a slight increase in coherence scores and a substantial increase in the interpretability of topics from LDA to lda2vec.

Implementing the lda2vec topic model for news articles had less of an effect on the coherence score, raising it from 0.588 to 0.59. However, there does seem to be an increase in the specificity of topics for articles from LDA to lda2vec. Looking at Table 5, we seem to have topics about the U.S. primary elections (Topic 2), the investigation surrounding Trump, Russia, and the FBI (Topic 8), and the European Union (Topic 15). While these topics are no more interpretable than those seen in the LDA model (Table 3, Figure 5), they do seem to be more specific, which may lead one to think that the lda2vec algorithm was better at creating more precise and nuanced topics. The increase in specificity here and the interpretability in the two model lyrics could be due in part to the fact that lda2vec looks at local relationships and can therefore pick up on more subtle relations between words, like fbi and comey.

When the results of the lyrics lda2vec model are compared to the results of the articles lda2vec model, it doesn't seem the algorithm "closed the gap" we noticed in the LDA models for each type of text. The coherence score for articles was still much larger than the coherence score for song lyrics. Also, the topics for the articles were still far more consistently and precisely interpretable than those for the lyrics (exploring the visualizations for each will clearly demonstrate this and can also be seen in Tables 4 and 5). This provides further evidence that news articles are topic-focused text and song lyrics are non-topic-focused texts and that non-topic-focused texts are more difficult to model. This makes sense given the point of an article is to inform or express an opinion about a specific topic while songwriting is not so focused or structured or focused on an explicit topic.

Our hypothesis that lda2vec would provide a more accurate topic model than LDA for song lyrics was supported by our results, albeit not to the extent that we had anticipated. This can be explained in terms of the underlying properties of the algorithm because lda2vec is adding word2vec used word embeddings that measure more localized relationships. This we assume would be more useful for songs because they are often written so the context and structure of the words probably have much more to do with the topic than the actual words used. Our hypothesis that topic modeling for lyrics would be less accurate than news articles is supported in our LDA and LDA2vec model results, as the models for songs consistently performed worse than the models for news articles. This offers evidence that, as we thought, songs do not necessarily have clear or concrete topics the way articles do and thus are harder to model.

4. Related Work

One related work is from *Semantic Analysis of Song Lyrics* (Logan et al., 2004). In this work, PLSA (probabilistic latent semantic analysis) was used to characterize semantic content and determine artist similarity. While PLSA performs a similar analysis as LDA, LDA is creating a topic distribution that can be easily adapted for new texts. The biggest difference between the algorithms is the flexibility of the final model given by LDA. We want a flexible model, because as our data set grows in size (we add more and more songs to it), we want the same model to be used. PLSA is prone to overfitting to the data set at hand. The incorporation of word2vec in our topic analysis will give us a straightforward way to put names to the topics themselves where in many other related works topics are vague (i.e. "Topic 24").

Alen Lukic explored different topic modeling approaches for song lyrics in his paper, *A Comparison of Topic Modeling Approaches for a Comprehensive Corpus of Song Lyrics* (2015). He uses LDA to measure word-to-document relationships and to group words into topics, which we are also planning on using. He also uses the topic modeling technique Pachinko allocation (PAM) which is a graph-based algorithm that models correlations between topics and therefore creates more specific topics known as subtopics. Instead of extending LDA with PAM we are going to extend it with word2vec which models word-to-word relationships. We think that this could produce a more robust model of topics because instead of trying to create subtopics from LDA created topics we are hoping to capture potentially unrelated topics that would originally be overlooked using only LDA or LDA with PAM.

5. Code and Dataset

Link to our Github: https://github.com/carterwward/Topic-Mapping

Datasets:

Link to the Song Lyrics Dataset:
https://github.com/carterwward/Topic-Mapping/blob/main/data/song_df.csv
This dataset is a collection of roughly 1000 observations of song data collected from the Spotify and Genius API.

Link to the News Articles Dataset:
https://github.com/carterwward/Topic-Mapping/blob/main/data/large_article_sample.csv
This is a dataset from Kaggle containing roughly 140,000 rows of data on news articles. For our model, we randomly sampled 1000 of these observations for our model. The original dataset can be found here: https://www.kaggle.com/snapcrack/all-the-news?select=articles3.csv

LDA Models:
To train a LDA model for song lyrics, clone the repository and run this notebook linked here:
https://github.com/carterwward/Topic-Mapping/blob/main/lda_analysis_updated_songs.ipynb

To train a LDA model for news articles, clone the repository and run this notebook linked here:
https://github.com/carterwward/Topic-Mapping/blob/main/article_lda_updated_analysis.ipynb

LDA2vec Models: Much of the implementation for the steps below have been borrowed from https://github.com/TropComplique/lda2vec-pytorch

First, go to the get_windows.ipynb file and specify which dataset you would like to use in the "Load dataset" code block. Then in the second block in the "Prepare initialization for document weights", specify the number of topics. Then run the notebook from the top.
https://github.com/carterwward/Topic-Mapping/blob/main/src/get_windows.ipynb

Next, go to the train.py file. Change the path_prefix in line 9 to correspond to the dataset you used in the above notebook. Then, in line 27, change the number of topics to match the number you used in the above notebook. Lastly, run the python file (and wait a really long time).
https://github.com/carterwward/Topic-Mapping/blob/main/train.py

Lastly, to analyze the results of the model, load either the lda2vec_analysis_articles.ipynb or the lda2vec_analysis_lyrics.ipynb depending on the data you used. Simply run this notebook starting with the first code block.
https://github.com/carterwward/Topic-Mapping/blob/main/lda2vec_analysis_lyrics.ipynb
https://github.com/carterwward/Topic-Mapping/blob/main/lda2vec_analysis_articles.ipynb

6. Conclusion

Our results show that LDA2vec increases interpretability of topic modeling for song lyrics from the standard model of topic analysis LDA. By implementing the LDA2vec topic model we were able to raise the coherence score for our lyrics corpus slightly (0.354 to 0.373) and the interpretability of the topics significantly. These results illustrate that measuring more localized relationships between words creates more interpretable topics for song lyrics by utilizing word2vec in addition to LDA for lyrics.

We found that for both LDA and LDA2vec the news article topics were more interpretable and had better coherence scores than the song lyric topics. This offers evidence that, as we thought, songs do not necessarily have clear or concrete topics the way articles do and thus are harder to model.

This difference in coherence and interpretability we demonstrate between topic models for song lyrics and news articles shows how topic models are ill equipped for texts where the topics are more nuanced such as song lyrics. Future research in this area could focus on creating topic models that are able to produce more coherent and interpetible topics for non-topic-focused texts.

Bibliography

Bansal, S. (2020, December 23). Beginners guide to topic modeling in python and feature selection. Retrieved March 19, 2021, from https://www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/

Kapadia, Shashank. "Evaluate Topic Models: Latent Dirichlet Allocation (LDA)." *Medium*, Towards Data Science, 29 Dec. 2020, towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0.

Kumar, K. (2018, May 03). Evaluation of topic modeling: Topic coherence. Retrieved March 19, 2021, from https://datascienceplus.com/evaluation-of-topic-modeling-topic-coherence/

Moody, C. (n.d.). Introducing our hybrid lda2vec algorithm. Retrieved March 19, 2021, from https://multithreaded.stitchfix.com/blog/2016/05/27/lda2vec/

Moody, Chris E. "Cemoody/lda2vec." *GitHub*, github.com/cemoody/lda2vec.

Moody, Chris E. "Mixing Dirichlet Topic Models and Word Embeddings to Make lda2vec." *ArxXiv*, 6 May 2016.

TropComplique. "TropComplique/lda2vec-Pytorch." GitHub, github.com/TropComplique/lda2vec-pytorch.

Xu, J. (2018, December 20). Topic modeling With LSA, PSLA, Lda & lda2Vec. Retrieved March 19, 2021, from

https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05