Genre Playlist Prediction through Lyrical Analysis

Jaehyeon Park

20 March 2024

Abstract

This project explores the possibility of classifying songs into genre-specific playlists based on their lyrical content. The Spotify Web API and Genius API were used to collect playlist data and song lyrics for various genres, including English and Korean songs. Natural Language Processing (NLP) techniques were employed to preprocess the lyrics, and the Word2Vec model was used to embed the lyrics into vector representations. The Word2Vec model, a single-layer neural network, learns the semantic and syntactic relationships between words by considering their context, enabling the capture of meaningful similarities between words based on their contextual usage. The results reveal that while distinguishing between genre-specific playlists across different languages remains challenging, the classification accuracy within the same language (English) is notably high. This project demonstrates the potential of leveraging lyrical content for genre-based playlist prediction and highlights the importance of language-specific analysis. The findings suggest that lyrics-based playlist classification is promising, mainly when applied to songs within the same language. Further research can explore integrating multi-lingual approaches and incorporating additional features to enhance classification performance across different languages.

1 Introduction

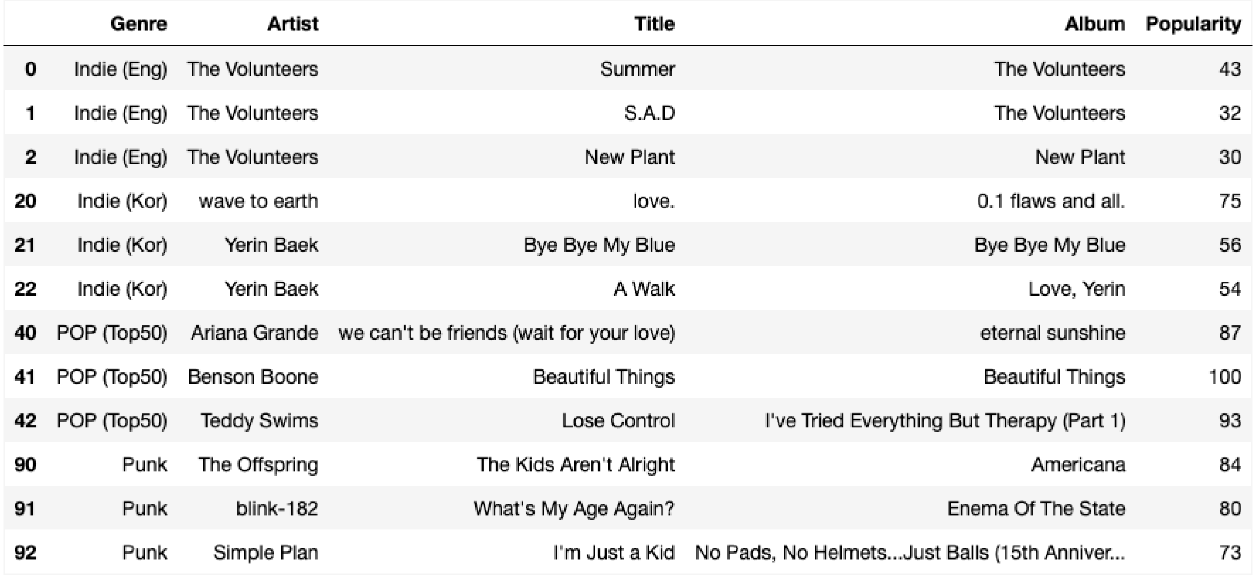
The inspiration for this project stems from a personal experience at a Yerin Baek concert in Berkeley, USA, last year. Surprisingly, the audience at the concert of the Korean indie singer consisted of more Americans than Koreans, which sparked curiosity about the reasons behind this phenomenon. While the melody undoubtedly plays a significant role in attracting listeners regardless of language barriers, the abundance of American indie singers singing in American English suggests that other factors might contribute to the appeal of Korean indie singers, even with imperfect English translations in their lyrics. This observation led to exploring the possibility of classifying music genres based on lyrics alone. This project investigates whether songs written in English can be accurately classified into their respective genres solely based on their lyrics and if this approach can be extended to English lyrics written by artists from other countries, such as Korean indie musicians. Furthermore, the project will explore whether precise genre classification is possible when comparing English songs to those in other languages. With the growing trend of music platforms automatically playing or recommending similar songs based on a given track, an effective classification method could enable more personalized and satisfactory song recommendations, transcending language barriers.

To assess the effectiveness of the classification, this project involves collecting songs and their corresponding lyrics and data preprocessing to prepare the information for analysis. During the Natural Language Processing (NLP) stage, the Word2Vec model is employed to learn and explore meaningful similarities between words based on their contextual usage. Finally, the trained model is tested using new songs from each genre that are not present in the training data.

The structure of this research begins by describing the data collection process, which includes the sources and methods used to gather songs and lyrics from various genres and languages. Next, the data preprocessing steps are detailed, focusing on cleaning and formatting the collected information to prepare it for analysis. The methodology section follows, explaining the Word2Vec model and its application in learning word embeddings from the preprocessed lyrics. Subsequently, the results of the genre classification experiments are presented, evaluating the approach's effectiveness across different genres and languages. Finally, the paper concludes by discussing the implications of the findings, limitations of the study, and potential future research directions.

2 Data Pre-processing

To obtain the songs for this project, the Spotify Web API (https://developer.spotify.com/dashboard) was used, which is accessible to anyone with a Spotify account. Intuitively distinguishable playlist genres were selected. For the pop genre playlist, the “Today’s Top Hits” (id: 37i9dQZF1DXcBWIGoYBM5M) playlist created by Spotify was used. The second genre chosen was Punk, which has a slightly different feel, and the “Punk Rock Mix” (id: 37i9dQZF1EIfCD6nwYB5OF) playlist, also created by Spotify, was used. The following two genres are Korean indie, with one playlist featuring English lyrics (id: 5JOQpjGZiLyekQnMKJ7rgV) and the other with Korean lyrics (id: 2jcSrBGxoJxS4FyRHOgn8f). The variable names for each genre were written as ‘POP (Top50)’, ‘Punk,’ ‘Indie (Eng),’ and ‘Indie (Kor),’ respectively. The variables extracted from each song included ‘Genre,’ ‘Artist,’ ‘Title,’ ‘Album,’ and ‘Popularity.’ The following table shows an example of three songs from each genre.

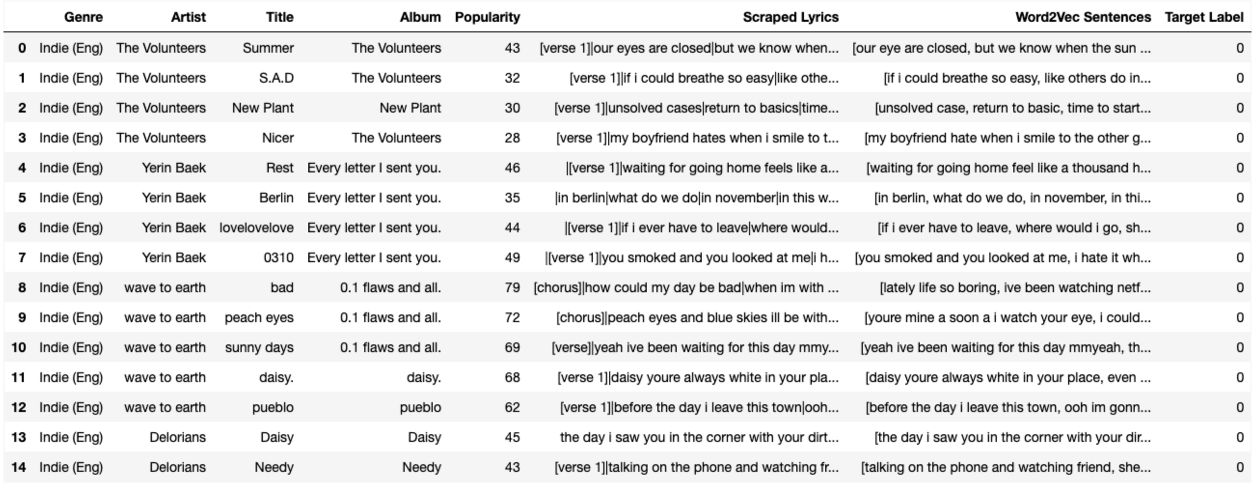


	Genre	Artist	Title	Album	Popularity
0	Indie (Eng)	The Volunteers	Summer	The Volunteers	43
1	Indie (Eng)	The Volunteers	S.A.D	The Volunteers	32
2	Indie (Eng)	The Volunteers	New Plant	New Plant	30
20	Indie (Kor)	wave to earth	love.	0.1 flaws and all.	75
21	Indie (Kor)	Yerin Baek	Bye Bye My Blue	Bye Bye My Blue	56
22	Indie (Kor)	Yerin Baek	A Walk	Love, Yerin	54
40	POP (Top50)	Ariana Grande	we can't be friends (wait for your love)	eternal sunshine	87
41	POP (Top50)	Benson Boone	Beautiful Things	Beautiful Things	100
42	POP (Top50)	Teddy Swims	Lose Control	I've Tried Everything But Therapy (Part 1)	93
90	Punk	The Offspring	The Kids Aren't Alright	Americana	84
91	Punk	blink-182	What's My Age Again?	Enema Of The State	80
92	Punk	Simple Plan	I'm Just a Kid	No Pads, No Helmets...Just Balls (15th Anniver...	73

Unfortunately, the Spotify Web API does not provide song lyrics, so the lyrics had to be obtained separately through web scraping. Due to variations in how lyrics are written on different sites, including annotations, retrieving the lyrics from a single source was preferable. The chosen website was Genius, a lyrics support site that fortunately also provides a web API (https://genius.com/api-clients). However, lyrics for less popular songs were sometimes unavailable, particularly for Korean indie songs, resulting in a smaller sample size for these genres than the others. Consequently, 50 samples were obtained for each of the ‘POP (Top50)’ and ‘Punk’ genres, while 20 samples were obtained for each of the ‘Indie (Eng)’ and ‘Indie (Kor)’ genres.

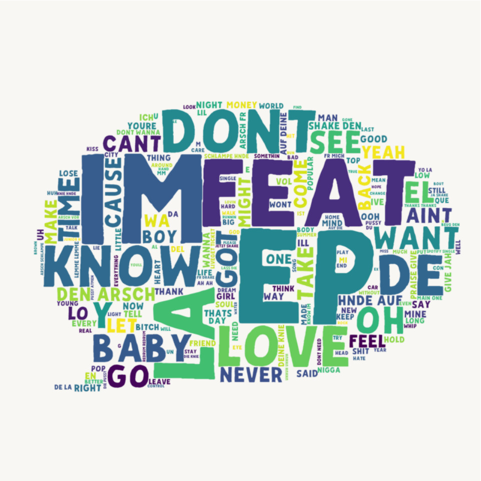
The song lyrics provided by the Genius API followed specific patterns. They appeared after the contributors and lyrics sections, and annotations such as [verse] or [chorus] were present in almost all cases. Recognizing these patterns, the lyrics were separated by ‘|’ whenever a new line started and added to the DataFrame in a column named ‘Scraped Lyrics’.

One of the key objectives of the data preprocessing phase was to adapt the acquired lyric information to be compatible with the Word2Vec model. To accomplish this, several steps were undertaken. Firstly, irrelevant lines such as “Line repeat” or “Chorus,” along with any numbers and symbols present in the lyrics, were eliminated. Subsequently, all words were converted to lowercase to ensure consistency. The next step involved lemmatizing the words, which was favored over stemming. While both methods yielded comparable outcomes in the project, lemmatization has the advantage of transforming words into their base form, resulting in more comprehensible results. For instance, the word ”running” would be reduced to “run” through lemmatization, whereas stemming would truncate it to “runn.” Interestingly, stopwords, which are typically removed in NLP preprocessing, were retained in this project. Their removal yielded no improvements in the results, likely due to the relatively small size of the vocabulary and the Word2Vec model's ability to utilize them in context evaluation. Finally, word tokenization was carried out during the model training phase. The resulting data frame after these preprocessing steps is shown in the following table.

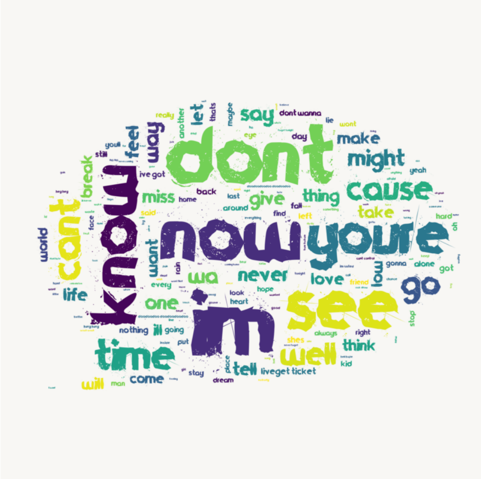


	Genre	Artist	Title	Album	Popularity	Scraped Lyrics	Word2Vec Sentences	Target Label
0	Indie (Eng)	The Volunteers	Summer	The Volunteers	43	[verse 1]|our eyes are closed|but we know when...	[our eye are closed, but we know when the sun ...	0
1	Indie (Eng)	The Volunteers	S.A.D	The Volunteers	32	[verse 1]|if i could breathe so easy|like othe...	[if i could breathe so easy, like others do in...	0
2	Indie (Eng)	The Volunteers	New Plant	New Plant	30	[verse 1]|unsolved cases|return to basics|time...	[unsolved case, return to basic, time to start...	0
3	Indie (Eng)	The Volunteers	Nicer	The Volunteers	28	[verse 1]|my boyfriend hates when i smile to t...	[my boyfriend hate when i smile to the other g...	0
4	Indie (Eng)	Yerin Baek	Rest	Every letter I sent you.	46	|[verse 1]|waiting for going home feels like a...	[waiting for going home feel like a thousand h...	0
5	Indie (Eng)	Yerin Baek	Berlin	Every letter I sent you.	35	|in berlin|what do we do|in november|in this w...	[in berlin, what do we do, in november, in thi...	0
6	Indie (Eng)	Yerin Baek	lovelovelove	Every letter I sent you.	44	|[verse 1]|if i ever have to leave|where would...	[if i ever have to leave, where would i go, sh...	0
7	Indie (Eng)	Yerin Baek	0310	Every letter I sent you.	49	|[verse 1]|you smoked and you looked at me|i h...	[you smoked and you looked at me, i hate it wh...	0
8	Indie (Eng)	wave to earth	bad	0.1 flaws and all.	79	[chorus]|how could my day be bad|when im with ...	[lately life so boring, ive been watching netf...	0
9	Indie (Eng)	wave to earth	peach eyes	0.1 flaws and all.	72	[chorus]|peach eyes and blue skies ill be with...	[youre mine a soon a i watch your eye, i could...	0
10	Indie (Eng)	wave to earth	sunny days	0.1 flaws and all.	69	[verse]|yeah ive been waiting for this day mmy...	[yeah ive been waiting for this day mmyeah, th...	0
11	Indie (Eng)	wave to earth	daisy.	daisy.	68	[verse 1]|daisy youre always white in your pla...	[daisy youre always white in your place, even ...	0
12	Indie (Eng)	wave to earth	pueblo	pueblo	62	[verse 1]|before the day i leave this town|ooh...	[before the day i leave this town, ooh im gonn...	0
13	Indie (Eng)	Delorians	Daisy	Daisy	45	the day i saw you in the corner with your dirt...	[the day i saw you in the corner with your dir...	0
14	Indie (Eng)	Delorians	Needy	Needy	43	[verse 1]|talking on the phone and watching fr...	[talking on the phone and watching friend, she...	0

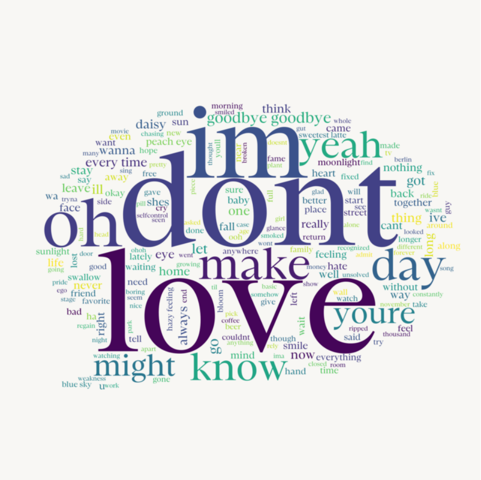
Finally, the Word cloud were generated to provide an overall impression of the word atmosphere for each genre playlist. Different fonts were used for each genre to match the vibe of each playlist: Emilo for 'Indie (Eng),' Joseon Myungjo for 'Indie (Kor),' Plamer sans for 'POP (Top50)', and Funkrocker for 'Punk.' The generated images are as follows.



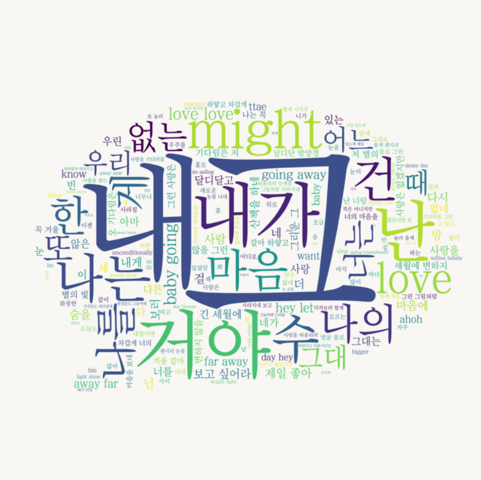
POP (Top50)



Punk



Indie (Eng)



Indie (Kor)

The word clouds were created by merging the lyrics of all songs into two variables, one per playlist, and then generating the word cloud using masks. These charts are based on the occurrences of words, highlighting the ones that occur most frequently in each playlist. The words appearing in the Word cloud can later be compared to the words extracted, as they are similar to the centroids of each playlist.

3 Methodology

In this project, the Word2Vec model and logistic regression were employed to classify songs into genre-specific playlists based on their lyrical content. The Word2Vec model, developed by Mikolov et al. (2013), is a neural network-based approach that learns word embeddings by considering the context in which words appear. It captures semantic and syntactic relationships between words, allowing for meaningful representations in a high-dimensional vector space.

The specific variant of the Word2Vec model used in this project is the skip-gram architecture. The objective of the skip-gram model is to predict the context words given a target word. Given the current word, it maximizes the average log probability of observing the context words. The skip-gram model maximizes the following average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t)$$

where T is the size of the training corpus, c is the context window size, and w_t is the target word at position t.

The conditional probability $p(w_{t+j}|w_t)$ is computed using the softmax function, where v denotes the input and output vector representations of the word w within the neural network architecture:

$$p(w_O|w_I) = \frac{\exp(u_{w_O}^T v_{w_I})}{\sum_{w=1}^{W}\exp(u_w^T v_{w_I})}$$

In this equation, v_{w_I} represents the input vector corresponding to the target word w_I, u_{w_O} represents the output vector corresponding to the context word w_O, and W denotes the size of the vocabulary, i.e., the total number of unique words in the dataset.

The model architecture consists of an input layer, a hidden layer, and an output layer. The input layer represents the target word, the hidden layer learns the word embeddings, and the output layer predicts the context words. The final output will be vectors (one per word in the vocabulary) of a desired dimension (another model hyperparameter).

These vectors should represent the relationships among words with high accuracy. For instance, Mikolov et al. (2013) obtained vectors that could be used to perform linear operations, such as <King> - <Man> + <Woman> would give a result that is very close to the vector corresponding to <Queen>.

Two techniques were employed to train the Word2Vec model efficiently: hierarchical softmax and negative sampling. Hierarchical softmax approximates the softmax function using a binary tree representation of the output layer, reducing the computational complexity. Negative sampling, on the other hand, distinguishes between the target word and randomly sampled "noise" words, simplifying the training process.

The hyperparameters of the Word2Vec model, such as the embedding dimension, context window size, and minimum word frequency, were tuned to achieve optimal performance on the lyrical data. The resulting word embeddings capture the semantic relationships between words in the context of song lyrics.

Once the word embeddings were obtained, they were used as features for a logistic regression classifier. Logistic regression is a popular choice for binary classification tasks, and in this case, it was extended to handle multiple genre classes. The input to the logistic regression model was the average of the word embeddings for each song, representing the overall lyrical content.

The logistic regression model was trained using a supervised learning approach, with the genre labels as the target variable. The dataset was split into training and testing sets, and cross-validation was performed to assess the model's performance and prevent overfitting. The hyperparameters of the logistic regression model, such as the regularization strength, were tuned using grid search to find the optimal configuration.

To evaluate the effectiveness of the proposed methodology, several metrics were employed. Accuracy was used to measure the overall correctness of the genre classifications. Precision, recall, and F1-score were calculated for each genre class to assess the model's performance in correctly identifying songs belonging to specific genres. Additionally, confusion matrices were generated to visualize the model's predictions and identify any common misclassifications.

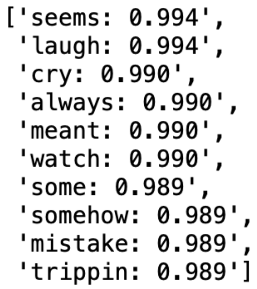
Combining the Word2Vec model for learning word embeddings and logistic regression for classification provides a powerful approach for genre playlist prediction based on lyrical content. The Word2Vec model captures the semantic relationships between words, while logistic regression leverages these embeddings to make genre predictions. By tuning the hyperparameters and evaluating the model's performance, the methodology aimed to achieve accurate and reliable genre classifications.

4 Data Interpretation

The output vectors for each word were set to have 300 coordinates (vector size=300), which is the same dimension used by word2vec models trained on the entire Google news corpus. The context window was set to 4 words (window=4), as this was found to be suitable for the project, which included both English and Korean lyrics. Words appearing less than three times were excluded from the corpus (min count=3) to filter out rare words and reduce the vocabulary size.

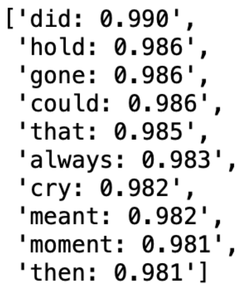
After training, a vector is obtained for each word in the playlists, allowing for the computation of playlist centroids and the identification of the most representative words in each playlist. To achieve this, the vectors corresponding to each word in a playlist were extracted and averaged to obtain the centroids. Subsequently, words similar to the centroid were extracted, representing the most characteristic words of each playlist.

The "top" 10 words for each playlist are as follows:



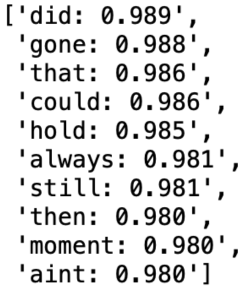
```
['seems: 0.994',
 'laugh: 0.994',
 'cry: 0.990',
 'always: 0.990',
 'meant: 0.990',
 'watch: 0.990',
 'some: 0.989',
 'somehow: 0.989',
 'mistake: 0.989',
 'trippin: 0.989']
```

POP (Top50)



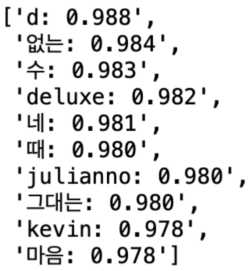
```
['did: 0.990',
 'hold: 0.986',
 'gone: 0.986',
 'could: 0.986',
 'that: 0.985',
 'always: 0.983',
 'cry: 0.982',
 'meant: 0.982',
 'moment: 0.981',
 'then: 0.981']
```

Punk



```
['did: 0.989',
 'gone: 0.988',
 'that: 0.986',
 'could: 0.986',
 'hold: 0.985',
 'always: 0.981',
 'still: 0.981',
 'then: 0.980',
 'moment: 0.980',
 'aint: 0.980']
```

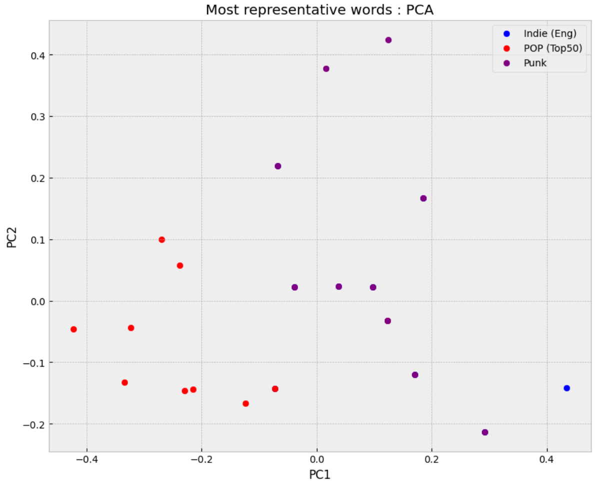
Indie (Eng)

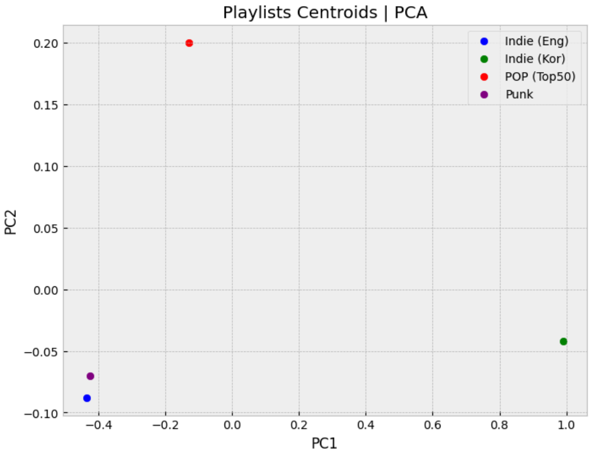


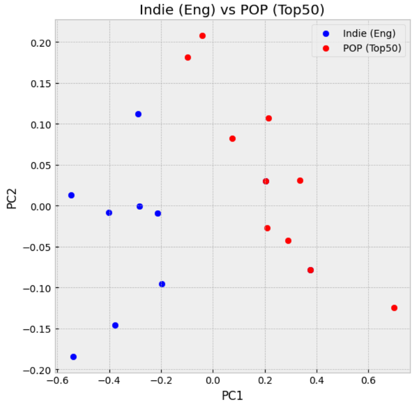
```
['d: 0.988',
 '없는: 0.984',
 '수: 0.983',
 'deluxe: 0.982',
 '네: 0.981',
 '때: 0.980',
 'julianno: 0.980',
 '그대는: 0.980',
 'kevin: 0.978',
 '마음: 0.978']
```

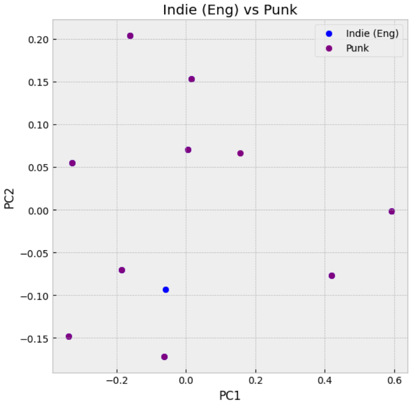
Indie (Kor)

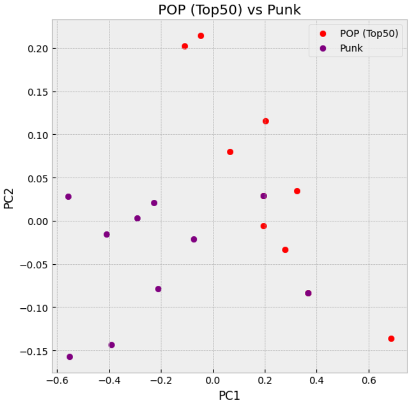
At this point, the project encountered its first challenge: the unusual results for the Indie (Kor) playlist. This issue will be addressed in more detail in the upcoming Challenges section. Due to this problem, it became evident that more complex complementary measures are necessary for analyzing Korean lyrics. To ensure a more efficient project, the decision was made to proceed with the remaining three genres, excluding Indie (Kor). Examining the "top" 10 words, it is clear that they are not identical to the word cloud images generated earlier in the Data pre-processing step. PCA was used to present the 300-coordinate vectors in a 2D space to visualize these words. The results are shown below, with the three genres plotted together and separately. Although the genres are not 100% separated, they can still be distinguished to a certain extent. After briefly discussing the limitations in the next section, the results will be revisited.

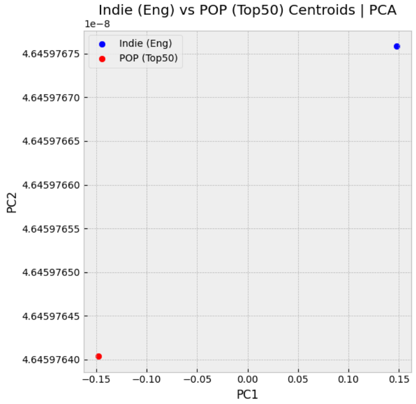


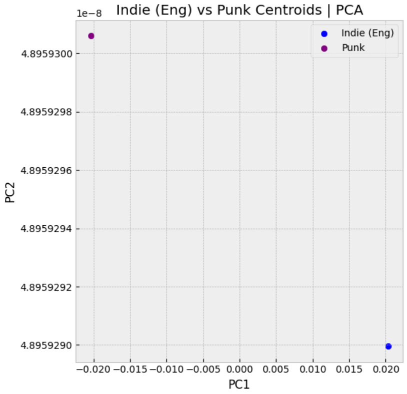


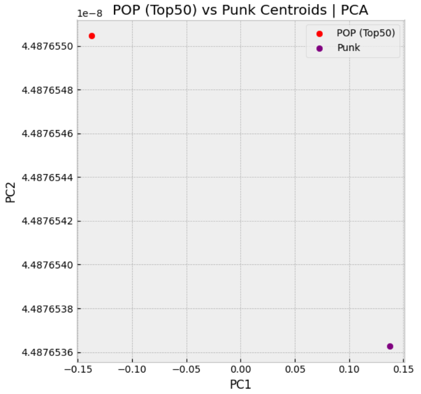








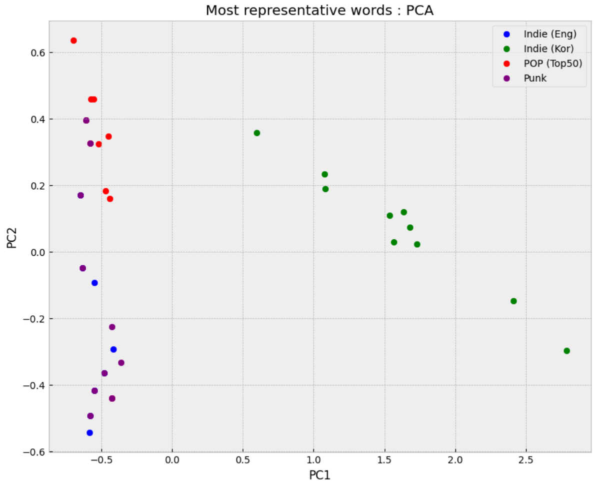


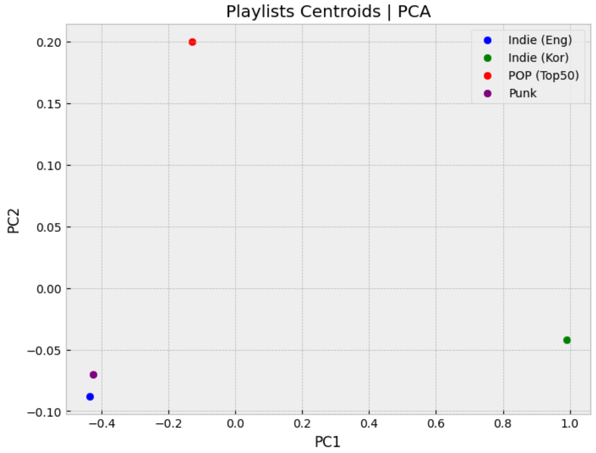


5 Challenges and Solutions

As briefly mentioned in the previous section, one of the project's goals, which was to classify playlists across different languages, encountered a problem. The biggest obstacle is the structural difference between Korean and English. Using the method employed in this project, Korean words are separated into morphological units, which are likely to be particles without any meaning. Moreover, even if the words are separated into meaningful units, the nuance of a Korean word can easily change with just one particle despite having the same core meaning. As a result, as evident from the top 10 words in the previous section, many meaningless words were extracted. Additionally, songs from the Korean indie scene rarely contain English at all. Even if a small portion of the song includes English, the classification becomes more complicated when English and Korean are mixed together. Furthermore, as seen in the graphs, using Korean is inefficient for classification purposes.

It may seem obvious, but when only one genre uses a different language, it is clearly distinguished from the others in the graph. This is not effective for classifying the other three genres.



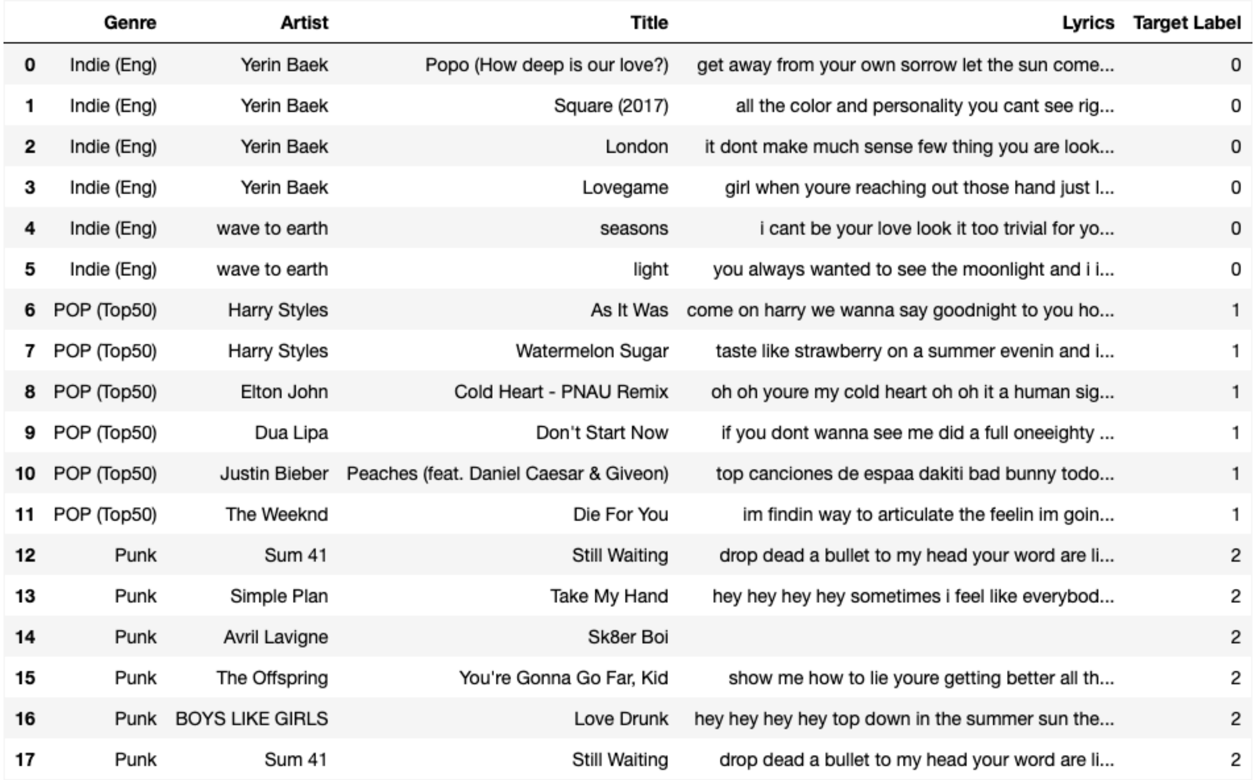


To address this issue, the project attempted to classify the words into minor meaningful units and then translate them into English for reapplication. However, the criteria for translation were ambiguous, and the results were expected to vary significantly depending on the translation. Therefore, the project decided to exclude Korean songs from the indie genre. Of course, given these results, the project could have restarted from the beginning without including Korean. However, this section is included in case it might be helpful for someone working on a similar playlist classification project.

The second challenge is the genre-specific issue with indie music. The term ”indie” is derived from the word ”independent,” which refers to artists who handle all aspects of music production, creation, and distribution without being affiliated with a record label. Naturally, indie artists often face financial constraints, making indie a minor genre. The indie genre can encompass all major genres, albeit on a smaller scale. Therefore, when attempting to classify songs based solely on lyrics, there is a high probability of overlap with other major genres. To obtain better results, it would be advisable to use playlist samples from more clearly defined genres.

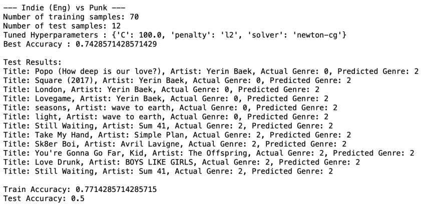
6 Results and Visualization

To summarize the process for the results, lyrics from the training set were used to compute an average vector per song, which served as the train set. Six songs from each genre not present in the train samples were selected for the test set to avoid overfitting. Vectors were computed for each test song using the same word2vec model. The test samples are as follows:



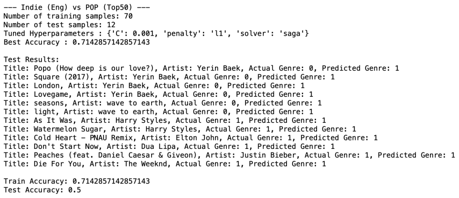
	Genre	Artist	Title	Lyrics	Target Label
0	Indie (Eng)	Yerin Baek	Popo (How deep is our love?)	get away from your own sorrow let the sun come...	0
1	Indie (Eng)	Yerin Baek	Square (2017)	all the color and personality you cant see rig...	0
2	Indie (Eng)	Yerin Baek	London	it dont make much sense few thing you are look...	0
3	Indie (Eng)	Yerin Baek	Lovegame	girl when youre reaching out those hand just l...	0
4	Indie (Eng)	wave to earth	seasons	i cant be your love look it too trivial for yo...	0
5	Indie (Eng)	wave to earth	light	you always wanted to see the moonlight and i i...	0
6	POP (Top50)	Harry Styles	As It Was	come on harry we wanna say goodnight to you ho...	1
7	POP (Top50)	Harry Styles	Watermelon Sugar	taste like strawberry on a summer evenin and i...	1
8	POP (Top50)	Elton John	Cold Heart - PNAU Remix	oh oh youre my cold heart oh oh it a human sig...	1
9	POP (Top50)	Dua Lipa	Don't Start Now	if you dont wanna see me did a full oneeighty ...	1
10	POP (Top50)	Justin Bieber	Peaches (feat. Daniel Caesar & Giveon)	top canciones de espaa dakiti bad bunny todo...	1
11	POP (Top50)	The Weeknd	Die For You	im findin way to articulate the feelin im goin...	1
12	Punk	Sum 41	Still Waiting	drop dead a bullet to my head your word are li...	2
13	Punk	Simple Plan	Take My Hand	hey hey hey hey sometimes i feel like everybod...	2
14	Punk	Avril Lavigne	Sk8er Boi		2
15	Punk	The Offspring	You're Gonna Go Far, Kid	show me how to lie youre getting better all th...	2
16	Punk	BOYS LIKE GIRLS	Love Drunk	hey hey hey hey top down in the summer sun the...	2
17	Punk	Sum 41	Still Waiting	drop dead a bullet to my head your word are li...	2

Hyperparameter tuning was performed using grid search with cross-validation to identify the best model hyperparameters. After finding the optimal hyperparameters, the logistic regression model was trained on the train set and tested by classifying the new songs. The results of comparing two genres at a time are as follows:



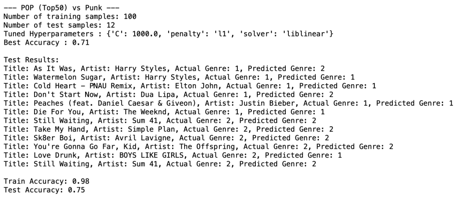
```
--- Indie (Eng) vs Punk ---
Number of training samples: 70
Number of test samples: 12
Tuned Hyperparameters : {'C': 100.0, 'penalty': 'l2', 'solver': 'newton-cg'}
Best Accuracy : 0.7428571428571429

Test Results:
Title: Popo (How deep is our love?), Artist: Yerin Baek, Actual Genre: 0, Predicted Genre: 2
Title: Square (2017), Artist: Yerin Baek, Actual Genre: 0, Predicted Genre: 2
Title: London, Artist: Yerin Baek, Actual Genre: 0, Predicted Genre: 2
Title: Lovegame, Artist: Yerin Baek, Actual Genre: 0, Predicted Genre: 2
Title: seasons, Artist: wave to earth, Actual Genre: 0, Predicted Genre: 2
Title: light, Artist: wave to earth, Actual Genre: 0, Predicted Genre: 2
Title: Still Waiting, Artist: Sum 41, Actual Genre: 2, Predicted Genre: 2
Title: Take My Hand, Artist: Simple Plan, Actual Genre: 2, Predicted Genre: 2
Title: Sk8er Boi, Artist: Avril Lavigne, Actual Genre: 2, Predicted Genre: 2
Title: You're Gonna Go Far, Kid, Artist: The Offspring, Actual Genre: 2, Predicted Genre: 2
Title: Love Drunk, Artist: BOYS LIKE GIRLS, Actual Genre: 2, Predicted Genre: 2
Title: Still Waiting, Artist: Sum 41, Actual Genre: 2, Predicted Genre: 2

Train Accuracy: 0.7714285714285715
Test Accuracy: 0.5
```

Indie (Eng) vs Punk



```
--- Indie (Eng) vs POP (Top50) ---
Number of training samples: 70
Number of test samples: 12
Tuned Hyperparameters : {'C': 0.001, 'penalty': 'l1', 'solver': 'saga'}
Best Accuracy : 0.7142857142857143

Test Results:
Title: Popo (How deep is our love?), Artist: Yerin Baek, Actual Genre: 0, Predicted Genre: 1
Title: Square (2017), Artist: Yerin Baek, Actual Genre: 0, Predicted Genre: 1
Title: London, Artist: Yerin Baek, Actual Genre: 0, Predicted Genre: 1
Title: Lovegame, Artist: Yerin Baek, Actual Genre: 0, Predicted Genre: 1
Title: seasons, Artist: wave to earth, Actual Genre: 0, Predicted Genre: 1
Title: light, Artist: wave to earth, Actual Genre: 0, Predicted Genre: 1
Title: As It Was, Artist: Harry Styles, Actual Genre: 1, Predicted Genre: 1
Title: Watermelon Sugar, Artist: Harry Styles, Actual Genre: 1, Predicted Genre: 1
Title: Cold Heart - PNAU Remix, Artist: Elton John, Actual Genre: 1, Predicted Genre: 1
Title: Don't Start Now, Artist: Dua Lipa, Actual Genre: 1, Predicted Genre: 1
Title: Peaches (feat. Daniel Caesar & Giveon), Artist: Justin Bieber, Actual Genre: 1, Predicted Genre: 1
Title: Die For You, Artist: The Weeknd, Actual Genre: 1, Predicted Genre: 1

Train Accuracy: 0.7142857142857143
Test Accuracy: 0.5
```

Indie (Eng) vs POP (Top50)

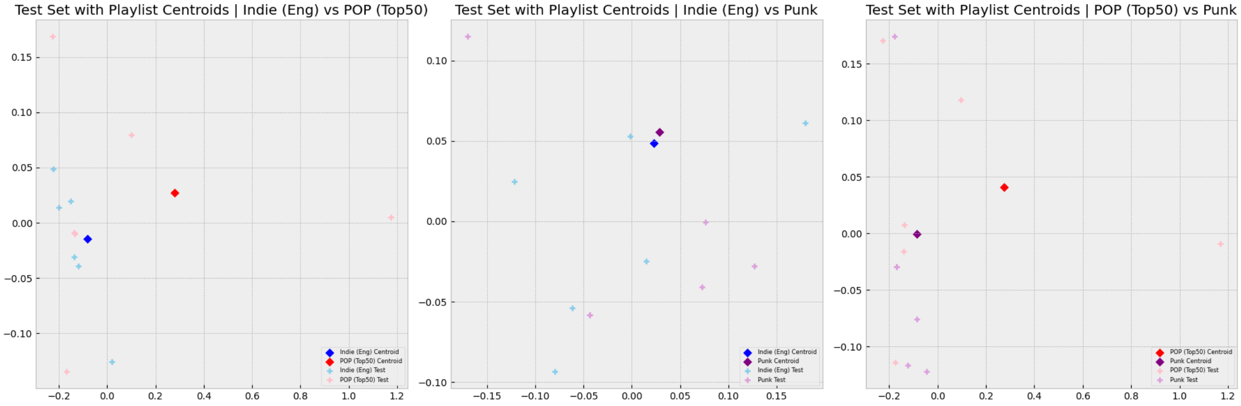


```
--- POP (Top50) vs Punk ---
Number of training samples: 100
Number of test samples: 12
Tuned Hyperparameters : {'C': 1000.0, 'penalty': 'l1', 'solver': 'liblinear'}
Best Accuracy : 0.71

Test Results:
Title: As It Was, Artist: Harry Styles, Actual Genre: 1, Predicted Genre: 2
Title: Watermelon Sugar, Artist: Harry Styles, Actual Genre: 1, Predicted Genre: 1
Title: Cold Heart - PNAU Remix, Artist: Elton John, Actual Genre: 1, Predicted Genre: 1
Title: Don't Start Now, Artist: Dua Lipa, Actual Genre: 1, Predicted Genre: 2
Title: Peaches (feat. Daniel Caesar & Giveon), Artist: Justin Bieber, Actual Genre: 1, Predicted Genre: 1
Title: Die For You, Artist: The Weeknd, Actual Genre: 1, Predicted Genre: 1
Title: Still Waiting, Artist: Sum 41, Actual Genre: 2, Predicted Genre: 2
Title: Take My Hand, Artist: Simple Plan, Actual Genre: 2, Predicted Genre: 2
Title: Sk8er Boi, Artist: Avril Lavigne, Actual Genre: 2, Predicted Genre: 2
Title: You're Gonna Go Far, Kid, Artist: The Offspring, Actual Genre: 2, Predicted Genre: 2
Title: Love Drunk, Artist: BOYS LIKE GIRLS, Actual Genre: 2, Predicted Genre: 1
Title: Still Waiting, Artist: Sum 41, Actual Genre: 2, Predicted Genre: 2

Train Accuracy: 0.98
Test Accuracy: 0.75
```

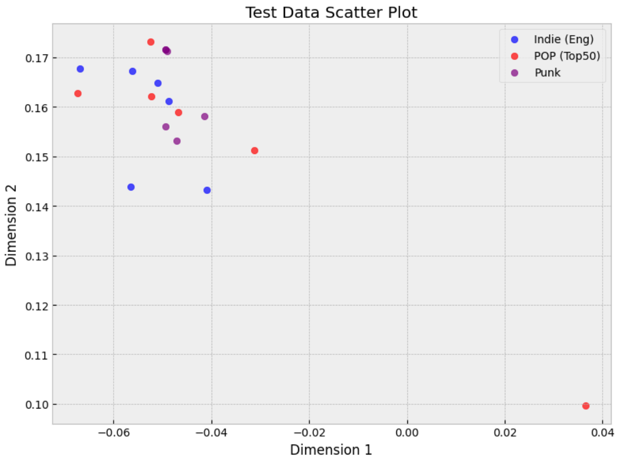
POP (Top50) vs Punk

As evident from the results, the performance is not exceptionally good, particularly when the Indie genre is involved. This could be attributed to the fact that the Indie genre is more likely to encompass songs that belong to other genres, as mentioned in the limitations section. Additionally, the limited number of samples contributed to the suboptimal performance.

The test data points were plotted along with the training data using PCA to visualize the results further.



Although the classification is relatively good, some songs need to be more qualified. However, an encouraging point is that although the distribution is not well-classified, most of the data points are distributed around the appropriate centroids. This suggests that for playlists where the classification is more distinct, the classification will be clearer.



The scatter plot confirms that the selected samples need to be more clearly distinguishable for specific genres, indicating the challenging nature of the classification task. In other words, it shows that the three genres used in the project are difficult to classify. When receiving automatic recommendations on music apps, certain songs may appear in the recommended lists for different genres, as is the case here. The more similar the genres are, the more additional conditions or classification methods beyond just lyrics seem necessary.

7 Conclusion

This project explored the feasibility of classifying songs into genre-specific playlists based solely on their lyrical content. The results demonstrated that while distinguishing playlists across different genres remains challenging, classification accuracy within the same language (English) is notably high. The findings suggest that a lyrics-based approach holds promise for genre prediction, mainly when applied to songs within the same language. However, the limited number of samples and the inherent overlap between specific genres, such as the indie genre encompassing elements from other genres, sometimes contributed to suboptimal performance.

Integrating multi-lingual approaches and incorporating additional features beyond lyrics, such as musical elements or artist information, could potentially enhance classification performance across different languages. Despite the challenges encountered, this project has piqued my interest in the topic. I plan to periodically update and refine the approach, as there is significant potential in leveraging lyrical content for personalized music recommendations and genre classification.

8 references

1. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013, September 7). Efficient estimation of word representations in vector space. arXiv.org. https://arxiv.org/abs/1301.3781

2. GfG. (2024, January 3). Word embedding using word2vec. GeeksforGeeks. https://www.geeksforgeeks.org/python-word-embedding-using-word2vec/

3. OpenAI. (2024). ChatGPT. Retrieved March 20, 2024, from https://openai.com/chatgpt