Music as economic barometer

Does song lyric embody the economic condition of an era?

Dingan Chen

Abstract

This paper studies the relationship between popular songs' lyrics and the economic condition of their respective era. The goal is to build a classification model where a song's lyrics are passed into the model, and the model can predict if the song is popular during the times of economic prosperity or economic depression. This study might attract scholars from multiple disciplines such as sociology, economics, and humanities. With the classification model, I can extract the "mood" of a nation in a certain period of time. In addition, I might also predict the economic growth from the national "mood" reflected in the music. I can also use the classifier to judge if a new song is "retro" or not.

The study tested a combination of 105 word embedding methods and machine learning model, including word2vec models trained on Twitter, Google News, and Wikipedia articles for 100, 200, or 300 dimensions and machine learning models such as DNN, CNN, and LSTM. For generating representations of each individual song's lyrics, two methods were used: 1. taking the average of the word vectors 2. padding the word vectors to a large matrix. The result shows that the correct combination of word embedding method and machine learning model (word2vec model trained on Twitter data and CNN) could perform better at the classification task better than a random guessing model (F1 Score of 0.63 vs 0.44). However, the author also recognizes the possibility of the model over-fitting the data set due to confounding variables.

1 Introduction

Each era has its distinctive genre of music. For example, the rock music in the 70s, the electronic music in the 80s, and the hip-hop music in the 90s. While enjoying a diverse set of music from many eras and different countries, I came up with the theory that popular song lyrics can reflect a country's economic health in that certain period. For example, most of the "City Pop" genre reflects the booming economy and celebrates consumerism of the late 80s in Japan. Thus I wish to use the project to study that if popular music can be used as a country's economy barometer. Specifically, I will use lyrics of Billboard Top 100 songs since 1950s as the corpus, divide the corpus into two categories based on the economic cycles as defined by National Bureau of economic research, extract textual features from the lyrics within each category, and build a classification model where it can categorize a song into the four pre-defined categories. With this classification model, I can classify the era or economic cycle when the music is written. Because the economic cycle data set is generated in hindsight, if we use this model on the latest popular music, we might be able to detect changes in the economic cycle for a certain country.

For economists, if we continue running this model on the latest popular music, we can catch the moment that the "national mood" has shifted. Ideally, we can use this model to analyze popular music and use it as an indicator for a country's economic health.

For humanities scholars, this classification algorithm can identify "outliers" in each era. For example, during the time of economic prosperity when everyone else is singing about prosperity, there might be an artist whose music is rather melancholic. Thus humanities scholars can find these "outlier artists" and study their work closely.

For further study, I can incorporate other features of music such as tempo and beats into the study. My hypothesis is that upbeat and bright tones are correlated with prosperity. However, at this time, I do not have a reliable source to scrape audio files. In addition, I have to think about the best way to represent tempo and beats numerically,

Singles from 1969		
January 4	"Soulful Strut"	Young-Holt Unlimited
	"Hooked on a Feeling"	B. J. Thomas
January 11	"Crimson and Clover"	Tommy James and the Shondells
January 18	"The Worst That Could Happen"	The Brooklyn Bridge
	"Touch Me"	The Doors
	"Son of a Preacher Man"	Dusty Springfield
January 25	"Everyday People"	Sly and the Family Stone
	"I Started a Joke"	Bee Gees

Figure 1: Tables that have merged cells

such the data can be used to train models.

2 Problem Definition and Data

The goal of this project is to create a classification model, where the input is a song's lyrics, and the output is one of the classes Prosperity or Recession, which represent the economic cycle of the year that the song made to Billboard Top 100 chart.

1. Song titles and artists information for Billboard Top 100 songs since the 1950s

   For this task, I used this Wikipedia page:

   https://en.wikipedia.org/wiki/List_of_Billboard_top-ten_singles(wik, 2019)

   One challenge I encountered during this process is the inconsistent formatting of Wikipedia tables. Between all pages for each year, the column name might not be exactly the same. For example, the column "artist" might be spelled as "artists" in some other pages. Also, the Wikipedia table is known for the practice of stacking column names together as shown in Figure. This makes it harder for me to parse the table.

   The Billboard Top 100 songs' titles and artists information has been scraped and stored in a local SQLite database. During the scraping process, I encountered a major problem in parsing the date column. For songs with the same entry date, the Wikipedia page will group these songs and merge the entry date cells, as shown in the Figure.

   To solve this problem, I used a code snippet(JoshuaJoshua) to flatten the rows and assign a date of entry for each song.

2. Scraping song lyrics based on song titles and artists information

   For this task, I used this website:

Clock speeds			Fillrate			
Base core clock (MHz)	Boost core clock (MHz)	Memory (MT/s)	Pixel (GP/s)[f]	Texture (GT/s)[g]	Size (GiB)	Ba
1151	1379	2100	18.41	27.6	2	
1227	1468	6000	19.6	29.4		
1354	1455		43.3	54.2		
1392	1518	7000	33.4	66.8	3	

Figure 2: Tables that have stacked column heads

https://www.lyrics.com/(lyr)

While this website seems to have a simplistic design, the challenge for this task is to identify if the resulting song is indeed the song I am looking for. There are plenty of songs with the same name, see Figure. Thus it becomes challenging to pick the right lyric that matches the song I am looking for, see Figure.

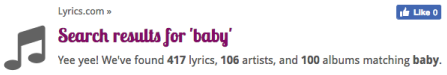

Figure 3: So many songs named "Baby"

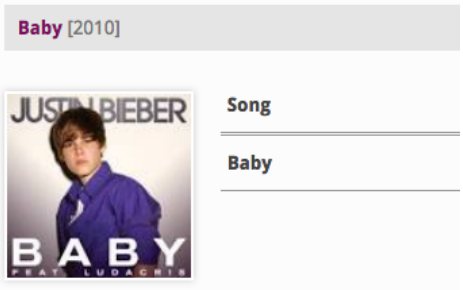

Figure 4: Yet I are looking for this song by Justin Bieber

The Billboard Top 100 songs have been scraped and stored in a local SQLite database. The database structure is shown as the above Figure. After manual inspection, it seems using title+artist name is sufficient enough to capture the correct song on lyrics.com that matches the song on Wikipedia. In addition, I saved the HTML of the search results page and encoded the HTML with base64. Thus if the future inspection finds out some songs were mis-

Title	Artist	SourceURL	YearDate	Lyrics	ArtistLyricsdotcom	AllURL
Poor Littl...	Ricky Nelson	https://en.wikipe...	August 4 19...	I used to play ...	Artist: Ricky Nel...	WylviIwgI90b3Atc29uZ...
Patricia	Pérez Prado	https://en.wikipe...	August 4 19...	Kiss her and y...	Artist: Perez Pra...	WylviIwgI90b3Atc29uZ...
Splish Spl...	Bobby Darin	https://en.wikipe...	August 4 19...	Splish splash, ...	Artist: Bobby Da...	WylviIwgI90b3Atc29uZ...
Hard Hea...	Elvis Presley	https://en.wikipe...	August 4 19...	Well a hard he...	Artist: Elvis Pres...	WylviIwgI90b3Atc29uZ...
When	Kalin Twins	https://en.wikipe...	August 4 19...	When, when y...	Artist: Kalin Twins	WylviIwgI90b3Atc29uZ...
Rebel 'Ro...	Duane Eddy	https://en.wikipe...	August 4 19...	Alisha's Attic J...	Artist: Duane Ed...	WylviIwgI90b3Atc29uZ...
Yakety Yak	The Coasters	https://en.wikipe...	August 4 19...	Take out the p...	Artist: The Coas...	WylviIwgI90b3Atc29uZ...
My True L...	Jack Scott	https://en.wikipe...	August 4 19...	I prayed to the...	Artist: Jack Scott	WylviIwgI90b3Atc29uZ...
Willie and...	The Johnny...	https://en.wikipe...	August 4 19...	I know a cat n...	Artist: The John...	WylviIwgI90b3Atc29uZ...
Fever	Peggy Lee	https://en.wikipe...	August 4 19...	Never know h...	Artist: Peggy Lee	WylviIwgI90b3Atc29uZ...

Figure 5: Song Database

matched, I have a locally-stored data to correct the mismatches.

3. US economic cycle data from National Bureau of economic research

   For this task, I used the data from this web page https://www.nber.org/cycles.html(nbe, 2010) NBER provided an Excel spreadsheet, listing each year and the year's corresponding economic cycle. I used Python's Pandas package to read this spreadsheet and extrapolate data.

With all the data collected, I have lyrics for Billboard Top 100 songs in every year since 1958, totaling 6658 songs. I also have U.S. economic cycle information for every year since 1958. Joining the two tables together, I have each song's lyrics-economic cycle at the time when the song is on Billboard Top 100 data set. The economic cycle will be the "class" for each lyric, and the model is supposed to predict the economic cycle given a song's lyrics.

Song	Artist	Lyrics	Year
99 Revolutions	Green Day	There's a trouble	2012
Uptown Funk	Mark Ronson	This hit, that ice	2014
Baby	Justin Bieber	Ooh whoa, ooh	2010

Figure 6: Lyrics data set

Year	Economic Cycle
1990	Expansion
1991	Expansion
1992	Peak
1993	Peak
1994	Recession

Figure 7: Economic cycle data set

Prior to building the machine learning model, I built word clouds to visually inspect some unique

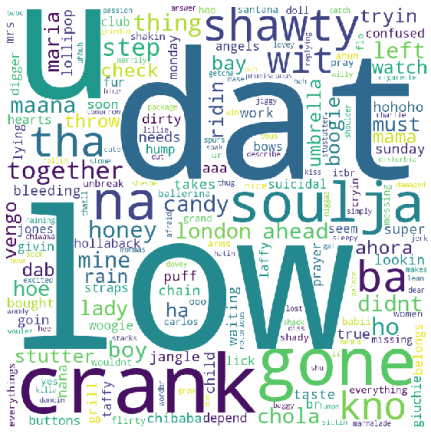

Figure 8: Unique Words in Boom Songs

words and artists to the songs in the boom economic cycle and bust economic cycle. The visualization (Amueller, 2019) results are somewhat counter-intuitive. For example, the unique words that only appear in boom songs include words such as low, gone, and crank, which are usually associated with pessimistic content. The unique words that only appear in bust songs include words such as yeah, like, boom, and love, which are usually associated with optimistic content.

One possible explanation could be that song lyrics are simply not correlated to the economic trend. Therefore, this visualization paints a gloomy picture that implies the predictive model I are building will perform poorly. However, this implication is overthrown by the results in the next section, where the best performing model actually outperforms a random guess model by a significant margin and indicates that the lyrics might be correlated with the economic trend. One caveat of this graph is that the graph is generated based on word tokens rather than n-gram. Using word tokens might lose some nuances in the text. For example, the token approach will overlook negations in the text. Therefore. this visualization, although gives us a general idea of the word composition in the lyrics, fails to accurately represent the nuances in the lyrics.

3 Related Work

1. https://ieeexplore.ieee.org/abstract/document/1394328(Logan et al., 2004) This paper primarily focuses on predicting music genre based on the lyrics. I think besides using song lyrics as features, I

Figure 9: Unique Words in Bust Songs

could also use the music genre composition of a specific era as the features for predicting the economic cycle of the time. One caveat may be that certain genre is only popular in a certain period, such as the rock music in the 80s and rap music in the 2000s.

2. https://ieeexplore.ieee.org/abstract/document/4725050(Laurier et al., 2008) This paper uses lyrics to extract the mood of the music. In addition to the genre, I could also use the music of the mood and compose a mood composition of a specific era and use this mood as a feature for predicting economic cycle.

3. https://ieeexplore.ieee.org/abstract/document/5363083(Yang and Lee, 2009) Similarly, this paper also uses lyrics to extract the mood of the music.

4 Methodology

For creating a representation of each song, I plan to use the following pre-trained Gensim models:

1. glove-wiki-gigaword-100
2. glove-wiki-gigaword-200
3. glove-wiki-gigaword-300
4. glove-twitter-100
5. glove-twitter-200
6. word2vec-google-news-300
7. fasttext-wiki-news-subwords-300

All of the above embedding methods are word-based(Mikolov et al., 2013), thus I used further processing techniques to generate a vector representation of the entire song, which is made of many words. Two methods are used:

1. Averaging method. For each word in the song, I retrieve their vector representation from the embedding model. Then I calculate the average of all word vectors within a song and use this averaged vector as the song's representation in the vector space. For example, if I use a 300 dimension Wikipedia embedding, my representation for any song is a vector with a dimension of 1X300.

2. Padding method. First I scan the entire song database, find the song with the longest list of words, and store the number of words for this song as Q. For each word in the song, I retrieve their vector representation from the embedding model. I store the word vectors as a list, then I add vectors with 0s to the front of the list, so the list contains Q elements for each song, regardless of how many words are actually in the song. For example, I use a 300 dimension Wikipedia embedding, my representation for any song is a matrix with a dimension of QX300.

After creating the representation for each song, I used the following framework for building the classification model (Chollet et al., 2015a):

1. Dummy Model (for benchmark)
2. Support Vector Machine
3. Naive Bayes
4. Logistic Regression
5. KNN Grouping
6. Decision Tree
7. Random Forest
8. XGBoost
9. Single-layer Neural Network
10. Deep Neural Network
11. Deep Neural Network that updates the embedding

12. Convolutional Neural Network

13. Convolutional Neural Network that updates the embedding

14. LSTM Neural Network

15. LSTM Neural Network that updates the embedding

In total, there are be 7*15, 105 combinations of models and embedding methods. Model 1 through 9 used the song representation produced by the averaging method. Model 10 through 15 used the song representation produced by the padding method.

I classify each song as being in "economic expansion" or in "economic contraction" by the following methodology:

1. Store all the years of the economic peak as defined in the National Bureau of Economic Research data set as list A

2. Store all the years of the economic bottom as defined in the National Bureau of Economic Research data set as list B

3. For each song in the data set, do the following:

   (a) Parse the year of the song

   (b) Try to plug the song's year in list A by calculating the absolute value of peak year - song year, index the smallest absolute value, use this index to retrieve which year is the closest to the song year, this year is named "the song's closest boom year," calculate the distance between the song year and closest boom year, and store the distance as "how close is this song to an economic peak?" (dBoom)

   (c) Repeat the process for list B and store the distance as "how close is this song to an economic bottom? (dBottom)

   (d) Compare the distances. If dBoom is smaller than dBottom, then the song is labeled as "popular in economic prosperity". If dBoom is larger than dBottom, then the song is labeled as "popular in economic depression".

F1-Scores				Av
WordVector	dummy	logistics	naïve bayes	support vector
WIKI100	0.430	0.410	0.521	0.000
WIKI200	0.435	0.444	0.561	0.000
WIKI300	0.440	0.479	0.569	0.000
TWITTER100	0.443	0.399	0.560	0.000
TWITTER200	0.443	0.437	0.564	0.000
GOOGLE300	0.440	0.385	0.552	0.000
FTEXT300	0.457	0.116	0.570	0.000

Figure 10: Results Part 1

F1-Scores	Average Vector			
WordVector	knn	decision tree	random forest	xgboost
WIKI100	0.533	0.577	0.533	0.519
WIKI200	0.502	0.604	0.560	0.536
WIKI300	0.542	0.566	0.516	0.555
TWITTER100	0.559	0.589	0.544	0.510
TWITTER200	0.532	0.584	0.523	0.520
GOOGLE300	0.548	0.572	0.536	0.526
FTEXT300	0.527	0.575	0.542	0.543

Figure 11: Results Part 2

5 Evaluation and Results

This problem is a standard supervised classification problem, where the output labels are provided. Thus to evaluate the accuracy of the model, I will be using the F1 score to compare the prediction with the ground truth. The F1 score is a metric that takes account of false positives and false negatives. A higher F1 score means better performance.

The figures shown here are the F1 scores for the 105 combinations of models and embedding methods. The dummy model is based on a random guessing principle, thus any model with a higher F1 score than the dummy model indicates that the model is performing better than random chance. Among all the combinations, a CNN model trained on word embedding generated from a 100-dimension Twitter word2vec model performs the best.

There are also other noteworthy results. Logistics regression performed the worst across all the embedding methods, and it performs even worse than a random guess classifier (the dummy) model. The support vector machine gave an F1 score of 0 across all the embedding methods. Upon a closer examination, the support vector machine model fails to generate any classification result. For any input, the support vector machine model will generate the same "negative" prediction. Thus the F1 score for the support vector machine model is 0.

F1-Scores	erage Vect			Per-toke
WordVector	1 layer nn	dnn	dnn improv	cnn
WIKI100	0.606	0.625	0.592	0.595
WIKI200	0.610	0.578	0.622	0.601
WIKI300	0.623	0.552	0.620	0.591
TWITTER100	0.596	0.593	0.611	0.634
TWITTER200	0.621	0.590	0.553	0.581
GOOGLE300	0.572	0.612	0.616	0.614
FTEXT300	0.512	0.600	0.627	0.568

Figure 12: Results Part 3

F1-Scores	Per-token Vector		
WordVector	cnn improv	lstm	lstm improv
WIKI100	0.632	0.577	0.575
WIKI200	0.580	0.599	0.575
WIKI300	0.627	0.588	0.608
TWITTER100	0.577	0.514	0.583
TWITTER200	0.613	0.597	0.628
GOOGLE300	0.620	0.601	0.629
FTEXT300	0.616	0.545	0.607

Figure 13: Results Part 4

Specs.				
	1 layer nn Specs.		dnn Specs.	
WordVector	Dense Layer Size	Drop Out Prob.	Dense Layer Size	Drop Out Prob.
WIKI100	200	n/a	256	0.25
WIKI200	200	n/a	128	0.25
WIKI300	200	n/a	128	0.25
TWITTER100	200	n/a	256	0.25
TWITTER200	200	n/a	128	0.25
GOOGLE300	200	n/a	128	0.25
FTEXT300	200	n/a	128	0.25

Figure 14: Specification Part 1

Specs.			
	cnn Specs.		
WordVector	Dense Layer Size	Cnn Layer Specs.	Drop Out Prob.
WIKI100	256	f=100,k=2	0.25
WIKI200	128	f=100,k=2	0.25
WIKI300	128	f=100,k=2	0.25
TWITTER100	256	f=100,k=2	0.25
TWITTER200	128	f=100,k=2	0.25
GOOGLE300	128	f=100,k=2	0.25
FTEXT300	128	f=100,k=2	0.25

Figure 15: Specification Part 2

Specs.			
	lstm Specs.		
WordVector	Dense Layer Size	Lstm Layer Size	Drop Out Prob.
WIKI100	50	100	0.25
WIKI200	50	100	0.25
WIKI300	50	100	0.25
TWITTER100	50	100	0.25
TWITTER200	50	100	0.25
GOOGLE300	50	100	0.25
FTEXT300	50	100	0.25

Figure 16: Specification Part 3

6 Discussion

Across all the machine learning models, the Twitter embedding seems to perform the best. I think this is because the Twitter corpus is mostly informal English, with a lot of informal spellings that are in a similar style as song lyrics. This similarity makes the embedding generated from Twitter corpus more relatable and makes it a better representation of the text. In contrast, Wikipedia embedding generate the worst results in general. I think this is because Wikipedia articles are mostly written and edited informal English, which has a different vocabulary and lexical meanings than informal English in song lyrics.

Across all the embedding methods, the convolutional neural network performs the best, even better than LSTM neural network in some cases. I think this is because the convolutional neural network takes account of the entire ”picture” of the lyrics, where the LSTM network gets the information by chunks. This lyrics classification task is actually similar to an image classification task, where neighboring features usually conveys important information.

In contrast, the logistics model and support vector machine model performed very poorly. The latter model cannot even make a prediction. I think this is caused by the high-dimensional feature for each train/test example(red). Each word in the lyrics is embedded to a vector with 100+ dimensions. This sparse dimensionality means the data points are hard to be separated by a line or plane. Another possible cause for this issue is how I generate the representation of the lyrics. In this project, two approaches were used. The first approach is to vectorize each word within a song’s lyrics, take the average of the vectors, and use the average vector as the representation of the en-

tire song. The second approach is to vectorize each word within a song's lyrics, store the vectors to a larger vector, and pad the larger vector to the size of the song with the most word tokens(kelvankelvan 12618 et al.). The first method loses the vector for each individual token, and the SVM model is trained and tested using data generated by the first method. The neural network models are trained and tested using data generated by the second method. This difference in treatment might explain why the SVM model performed poorly.

This data has demonstrated that there is some information embedded in song lyrics that are helpful for predicting the economy cycle of when the song became popular. Several models are able to achieve a significantly higher F1 score than the random guess estimator. However, there is some caveat in this study that might compromise the claim that song lyrics can predict the economic cycle. The U.S. economic cycle has a long interval, which means the economic cycle won't change in 10+ years. Each decade has its own distinctive music, which can be identified by the unique vocabulary in the songs. For example, popular songs in the 1950s and 1960s might be using phrases such as "flutter bum" or "gringles." (wor)The model could have picked up these features and used these features to guess a song's year's publication. This becomes problematic as the model also learns each year's economic cycle status from the training set. For example, the model could have learned that the presence of word token "gringles" indicates a song became popular in the 1950s. The model could also learn that the economic status in the 1950s is a boom. Then each time the model sees "gringles" in the test set, it will predict the label as "boom," which is correct. However, the model is not making a prediction using the data I would like it to use, the lyrics. Therefore, there is a possibility that the model is over-fitting the data instead of actually learning the relationship between lyrics and economic status. One potential solution is to split the test and train data set by years rather than randomly. For example, use the 1950s to 1980s as the training data and the rest as test data. Such configuration will guarantee that if the model picks up "short-cut" features in the training set, the model won't perform well in the test set since the unique word token will not exist in the test set.

7 Conclusion

Based on the results, it appears that some models are able to predict the classification given the lyrics. Specifically, the neural network model performed much better than models such as random forest or decision tree. All machine model performs better than the benchmark, which is a random guessing model that achieved an F1 score of 0.44. This result concludes that there is some key information in the songs' lyrics that could reflect the general economy status when the song became popular. However, this result might be generated by over-fitting the models, as mentioned in the discussion section. To conclude, this project served as an exploration of using NLP techniques to study social, cultural, and economy phenomenons.

8 Other Things I Tried

While reading the related work paper, I proposed the possibility to include acoustics features into the data set. For example, songs during economy prosperity might have a brighter tone and quicker tempo. However, two problems arise that prevented us from including this feature into the data set. First, it is hard to find a source that has the music files for all the songs in the data-set. Music recordings are generally copyrighted, thus there is not a convenient online website to batch download the audio files. Second, I could not find a way to convert audio files into numerical representation, similar to the process of converting words into numeric vectors via word2vec model. These two problems stopped us from including audio features and made us focus on using the textual features to build the model.

Another challenge I ran into is the scale of the model and limited computing resources I have on hand. For training the model, I used an RTX 2060 graphics card with 6GB of VRAM. However, some of the embeddings I tried has a dimension of 300, and the padding method further exacerbates this issue by creating a matrix representation for each song rather than a vector representation (Chollet et al., 2015b). As a result, I often ran into an out-of-memory error when training the model with embedding models with higher dimensions. To solve this issue, I changed the structure of the neural network model and tuned down the number of nodes in each layer.

9 What You Would Have Done Differently or Next

After the project concludes, I realize that there are many features that might be included to improve the performance of the model. For example, parts of speech information might be useful for understanding the structure of each lyric. Also, most songs have repeated verses. Thus it'd be interesting to see how different the model will perform if the repeated verses are removed.

In addition, I could also give more thoughts on how to split the test and training data set. As explained in the discussion section, the model might gradually learn distinctive features that can be used to predict the year of the songs. Since the economy cycle usually doesn't change in a decade, the model could also gradually learn the economy status of each year. Therefore, the model might not be using the textual information to directly predict the economy status. This flaw is caused by the fact that the test and train data set are split randomly, thus exposing all the years' songs to the model. A remedy is to use the pre-1970s songs to train the model and use the post-1970s songs to test model. This configuration will make sure that the model doesn't learn any "short-cut" for predicting the economy status label.

10 Workplan

One challenge in this step is having enough data. Songs are shorter than the typical text corpus such as newspaper. Thus I'm worried about not having enough tokens for building the model. Also, songwriters choose their vocabulary not only based on the idea they want to express but also based on rhymes and beats. Thus there might be some noise in the vocabulary. A solution for this is to scrape more songs and expand the song list beyond Billboard Top 100. Another solution might be testing the project in a different language set. For example, using UK Top 100 songs to classify UK's economic status.

11 Group Effort

This project is done by Dingan Derek Chen.

12 Making your project report public

The codes and data used in this project can be found at https://github.com/dinganc/630FP

Acknowledgments

In this project, I wish to thank Professor Jurgens for the discussion on the implementation of the project, specifically how to frame the project as a text classification problem. I also wish to thank Brandon Punturo for suggesting several machine learning classification frameworks for this project.

References

???? 17 slang terms from the 1950s we want everyone to start using again. https://www.metv.com/lists/17-slang-terms-from-the-1950s-we-want-everyone

???? r/machinelearning - where do support vector machines perform badly? https://www.reddit.com/r/MachineLearning/comments/3zqwbc/where_do_support_vector_machines_perform_badly/.

???? Welcome to lyrics.com. https://www.lyrics.com/.

2010. https://www.nber.org/cycles.html.

2019. List of billboard top-ten singles. https://en.wikipedia.org/wiki/List_of_Billboard_top-ten_singles.

Amueller. 2019. amueller/word$_c$*loud*. https://github.com/amueller/word_cloud.

François Chollet et al. 2015a. Keras. https://github.com/fchollet/keras.

François Chollet et al. 2015b. Keras. https://keras.io.

JoshuaJoshua. ???? How to parse table with rowspan and colspan. https://stackoverflow.com/questions/48393253/how-to-parse-table-with-rowspan-and-colspan/48451104.

kelvankelvan 12618, Nate RawNate Raw 1816, and aneesh joshianeesh joshi 288210. ???? Embedding vs inserting word vectors directly to input layer. https://stackoverflow.com/questions/53837088/embedding-vs-inserting-word-vectors-directly-t

C. Laurier, J. Grivolla, and P. Herrera. 2008. Multimodal music mood classification using audio and lyrics. In *2008 Seventh International Conference on Machine Learning and Applications*. pages 688–693. https://doi.org/10.1109/ICMLA.2008.96.

B. Logan, A. Kositsky, and P. Moreno. 2004. Semantic analysis of song lyrics. In *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE*

Cat. No.04TH8763). volume 2, pages 827–830 Vol.2. https://doi.org/10.1109/ICME.2004.1394328.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. Curran Associates Inc., USA, NIPS'13, pages 3111–3119. http://dl.acm.org/citation.cfm?id=2999792.2999959.

D. Yang and W. Lee. 2009. Music emotion identification from lyrics. In *2009 11th IEEE International Symposium on Multimedia*. pages 624–629. https://doi.org/10.1109/ISM.2009.123.