Brazilian Lyrics-Based Music Genre Classification Using a BLSTM Network

A Preprint

Raul de Araújo Lima
Department of Informatics, PUC-Rio
Rua Marques de Sao Vicente 225
Rio de Janeiro, RJ, 22420-030, Brazil
rlima@inf.puc-rio.br

Rômulo César C. de Sousa
Department of Informatics, PUC-Rio
Rua Marques de Sao Vicente 225
Rio de Janeiro, RJ, 22420-030, Brazil
rsousa@inf.puc-rio.br

Simone Diniz Junqueira Barbosa
Department of Informatics, PUC-Rio
Rua Marques de Sao Vicente 225
Rio de Janeiro, RJ, 22420-030, Brazil
simone@inf.puc-rio.br

Hélio Cortês Vieira Lopes
Department of Informatics, PUC-Rio
Rua Marques de Sao Vicente 225
Rio de Janeiro, RJ, 22420-030, Brazil
lopes@inf.puc-rio.br

February 9, 2022

Abstract

Organize songs, albums, and artists in groups with shared similarity could be done with the help of genre labels. In this paper, we present a novel approach for automatic classifying musical genre in Brazilian music using only the song lyrics. This kind of classification remains a challenge in the field of Natural Language Processing. We construct a dataset of $138,368$ Brazilian song lyrics distributed in 14 genres. We apply SVM, Random Forest and a Bidirectional Long Short-Term Memory (BLSTM) network combined with different word embeddings techniques to address this classification task. Our experiments show that the BLSTM method outperforms the other models with an F1-score average of 0.48. Some genres like *gospel*, *funk-carioca* and *sertanejo*, which obtained 0.89, 0.70 and 0.69 of F1-score, respectively, can be defined as the most distinct and easy to classify in the Brazilian musical genres context.

***K*eywords** Music Genre Classification · Natural Language Processing · Neural Networks

1 Introduction

Music is part of the day-to-day life of a huge number of people, and many works try to understand the best way to classify, recommend, and identify similarities between songs. Among the tasks that involve music classification, genre classification has been studied widely in recent years [1] since musical genres are the main top-level descriptors used by music dealers and librarians to organize their music collections [2].

Automatic music genre classification based only on the lyrics is considered a challenging task in the field of Natural Language Processing (NLP). Music genres remain a poorly defined concept, and boundaries between genres still remain fuzzy, which makes the automatic classification problem a nontrivial task [2].

Traditional approaches in text classification have applied algorithms such as Support Vector Machine (SVM) and Naïve Bayes, combined with handcraft features (POS and chunk tags) and word count-based representations, like bag-of-words. More recently, the usage of Deep Learning methods such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) has produced great results in text classification tasks.

Some works like [3], [4] [5] focus on classification of mood or sentiment of music based on its lyrics or audio content. Other works, like [2], and [6], on the other hand, try to automatically classify the music genre; and the work [7] tries to classify, besides the music genre, the best and the worst songs, and determine the approximate publication time of a song.

In this work, we collected a set of about 130 thousand Brazilian songs distributed in 14 genres. We use a Bidirectional Long Short-Term Memory (BLSTM) network to make a lyrics-based music genre classification. We did not apply an elaborate set of handcraft textual features, instead, we represent the lyrics songs with a pre-trained word embeddings model, obtaining an F1 average score of 0.48. Our experiments and results show some real aspects that exist among the Brazilian music genres and also show the usefulness of the dataset we have built for future works.

This paper is organized as follows. In the next section, we cite and comment on some related works. Section 3 describes our experiments from data collection to the proposed model, presenting some important concepts. Our experimental results are presented in Section 4, and Section 5 presents our concluding remarks and future work.

2 Related Works

Several works have been carried out to add textual information to genre and mood classification. Fell and Sporleder [7] used several handcraft features, such as vocabulary, style, semantics, orientation towards the world, and song structure to obtain performance gains on three different classification tasks: detecting genre, distinguishing the best and the worst songs, and determining the approximate publication time of a song. The experiments in genre classification focused on eight genres: Blues, Rap, Metal, Folk, R&B, Reggae, Country, and Religious. Only lyrics in English were included and they used an SVM with the default settings for the classification.

Ying *et al.* [1] used Part-of-Speech (POS) features extracted from lyrics and combined them with three different machine learning techniques – k-Nearest-Neighbor, Naïve Bayes, and Support Vector Machines – to classify a collection of 600 English songs by the genre and mood.

Zaanen and Kanters [8] used the term frequency and inverse document frequency statistical metrics as features to solve music mood classification, obtaining an accuracy of more than 70%.

In recent years, deep learning techniques have also been applied to music genre classification. This kind of approach typically does not rely on handcraft features or external data. In [6], the authors used a hierarchical attention network to perform the task in a large dataset of nearly half a million song lyrics, obtaining an accuracy of more than 45%. Some papers such as [9] used word embedding techniques to represent words from the lyrics and then classify them by the genre using a 3-layer Deep Learning model.

3 Methods

In this chapter we present all the major steps we have taken, from obtaining the dataset to the proposed approach to address the automatic music genre classification problem.

3.1 Data Acquisition

In order to obtain a large number of Brazilian music lyrics, we created a crawler to navigate into the *Vagalume*[1] website, extracting, for each musical genre, all the songs by all the listed authors. The implementation of a crawler was necessary because, although the *Vagalume* site provides an API, it is only for consultation and does not allow obtaining large amounts of data. The crawler was implemented using Scrapy[2], an open-source and collaborative Python library to extract data from websites.

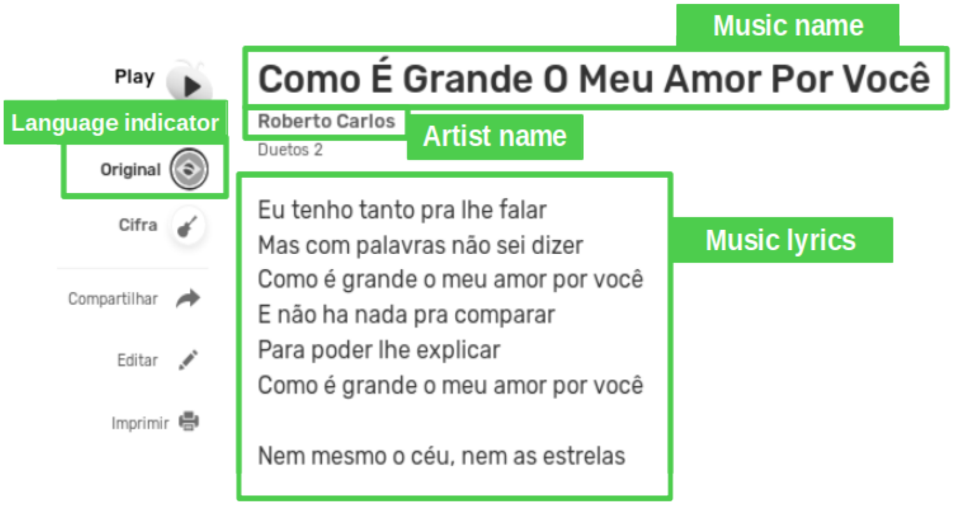
From the Vagalume's music web page, we collect the song title and lyrics, and the artist name. The genre was collected from the page of styles[3], which lists all the musical genres and, for each one, all the artists. We selected only 14 genres that we consider as representative Brazilian music, shown in Table 1. Figure 1 presents an example of the Vagalume's music Web page with the song *"Como é grande o meu amor por você"*[4], of the Brazilian singer Roberto Carlos. Green boxes indicate information about music that can be extracted directly from the web page. From this information, the language in which the lyrics are available can be obtained by looking at the icon indicating the flag of Brazil preceded by the *"Original"* word.

[1] https://www.vagalume.com.br/

[2] https://scrapy.org/

[3] `https://www.vagalume.com.br/browse/style/`

[4] `https://www.vagalume.com.br/roberto-carlos/como-e-grande-o-meu-amor-por-voce-letras.html`



Como É Grande O Meu Amor Por Você

Roberto Carlos

Duetos 2

Eu tenho tanto pra lhe falar
Mas com palavras não sei dizer
Como é grande o meu amor por você
E não ha nada pra comparar
Para poder lhe explicar
Como é grande o meu amor por você

Nem mesmo o céu, nem as estrelas

Figure 1: An example of a Vagalume's song web page

After extracting data, we obtained a set of $138,368$ songs distributed across 14 genres. Table 1 presents the number of songs and artists by genre. In order to use the data to learn how to automatically classify genre, we split the dataset into tree partitions: training ($96,857$ samples), validation ($27,673$ samples), and test ($13,838$ samples). The total dataset and splits are available for download[5].

Table 1: The number of songs and artists by genre

Genre	#songs	#artists
Gospel	33,344	800
Sertanejo	27,417	543
MPB	16,785	282
Forró	11,861	191
Pagode	8,199	174
Rock	8,188	396
Samba	6,221	111
Pop	4,629	338
Axé	4,592	63
Funk-carioca	4,557	279
Infantil	4,550	70
Velha-guarda	3,179	24
Bossa-nova	3,105	38
Jovem-guarda	1,741	18

3.2 Word Embeddings

Word embeddings is a technique to represent words as real vectors, so that these vectors maintain some semantic aspects of the real words. Basically, vectors are computed by calculating probabilities of the context of words, with the intuition that semantically similar words have similar contexts, and must therefore have similar vectors.

Word2Vec[6], by Mikolov *et al.* [10], is one of the first and most widely used algorithms to make word embeddings. It has two architectures to compute word vectors: Continuous Bag-Of-Words (CBOW) and Skip-gram. CBOW gets a context as input and predicts the current word, while Skip-gram gets the current word as input and predicts its context.

[5] https://drive.google.com/open?id=1b681ChByK737CpASYImFB4GfuqPdrvBN

[6] https://code.google.com/archive/p/word2vec/

In this work, we use the Python Word2Vec implementation provided by the Gensim[7] library. The Portuguese pre-trained word embeddings created by [11] and available for download[8] was used to represent words as vectors. We only used models of dimension 300 and, for Word2Vec, Wang2Vec, and FastText, skip-gram architectured models.

3.3 Bidirectional Long Short-Term Memory

Long Short-Term Memory (LSTM) is a specification of Recurrent Neural Network (RNN) that was proposed by Hochreiter and Schmidhuber [12]. This kind of network is widely used to solve classification of sequential data and is designed to capture time dynamics through graph cycles. Figure 2 presents an LSTM unity, which receives an input from the previous unit, processes it, and passes it to the next unit.

The following equations are used to update C_t and h_t values.

$$
\begin{gathered}
f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \\
i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i) \\
\widetilde{C}_t = tanh(W_C h_{t-1} + U_C x_t + b_C) \\
C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t \\
o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \\
h_t = o_t \times tanh(C_t)
\end{gathered}
$$

where W_f, W_i, W_C, W_o are the weight matrices for h_{t-1} input; U_f, U_i, U_C, U_o are the weight matrices for x_t input; and b_f, b_i, b_C, b_o are the bias vectors.

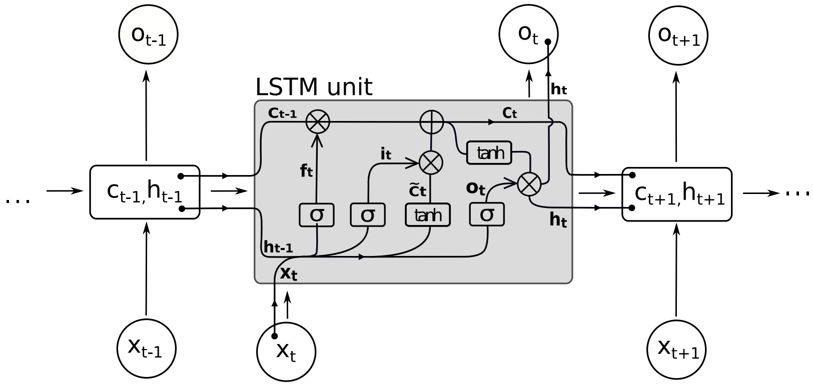


Figure 2: The Long Short-Term Memory unit.

Basically, a Bidirectional LSTM network consists of using two LSTM networks: a forward LSTM and a backward LSTM. The intuition behind it is that, in some types of problems, past and future information captured by forward and backward LSTM layers are useful to predict the current data.

3.4 Proposed Approach

Our proposed approach consists of three main steps. Firstly, we concatenate the title of the song with its lyrics, put all words in lower case and then we clean up the text by removing line breaks, multiple spaces, and some punctuation (,!.?). Secondly, we represent the text as a vector provided by a pre-trained word embeddings model. For classical learning algorithms like SVM and Random Forest, we generate, for each song, a vectorial representation by calculating the average of the vectors of each word in the song lyrics that can be can be expressed by the equation below:

$$
vector(L) = \frac{1}{n} \sum_{w \in L} vector(w)
$$

where L is the song lyrics, w is a word in L, and n is the number of words in L. If a word does not have a vector representation in the word embeddings model, it is not considered in the equation. For the BLSTM algorithm, the representation was made in the format of a matrix, as shown in Figure 3, where each line is a vector representation of a word in the lyrics. In the third step, we use as features the generated representation for the genre classification tasks using SVM, Random Forests, and BLSTM.

[7] https://radimrehurek.com/gensim/index.html

[8] http://nilc.icmc.usp.br/embeddings

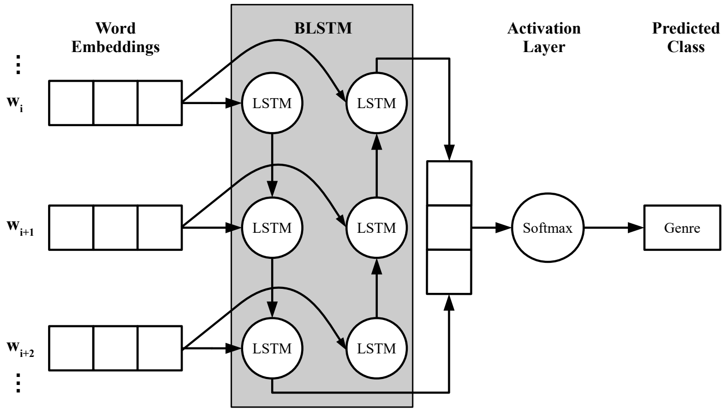


Figure 3: Our BLSTM model architecture

4 Experimental Results

In this section, we describe our experiments. We used the Linear SVM and Random Forest Scikit-learn[9] implementations and Keras[10] on top of TensorFlow[11] for the BLSTM implementation. In this study, we did not focus on finding the best combination of parameters for the algorithms, so that for SVM we used the default parameters, and for Random Forest we used a number of 100 trees. Our BLSTM model was trained using 4 epochs, with Adam optimizer, and 256 as the size of the hidden layer.

As we can see in Table 2, our BLSTM approach outperforms the other models with an F1-score average of 0.48. In addition, we can note that the use of Wang2Vec pre-trained word embeddings made it possible to obtain better F1-score results in BLSTM, which is not necessarily noticed in other cases, since for SVM and Random Forest, Glove and FastText, respectively, were the techniques that obtained better F1-scores.

Table 2: Classification results for each classifier and word embeddings model combination

System	Model	Precision	Recall	F1-score
SVM	Word2Vec	0.290	0.143	0.120
	Wang2Vec	0.297	0.144	0.121
	FastText	0.278	0.144	0.120
	Glove	0.296	0.144	**0.124**
Random Forest	Word2Vec	0.388	0.197	0.212
	Wang2Vec	0.386	0.197	0.207
	FastText	0.404	0.203	**0.215**
	Glove	0.394	0.199	0.210
BLSTM	Word2Vec	0.492	0.454	0.465
	Wang2Vec	**0.515**	**0.460**	**0.477**
	FastText	0.417	0.350	0.360
	Glove	0.492	0.460	0.470

Table 3 shows the BLSTM classification results for each genre. We can see that the genres *gospel*, *funk-carioca* and *sertanejo* have a greater distinction in relation to the other genres, since they were better classified by the model. In particular, *funk-carioca* obtained a good classification result although it did not have a large number of collected song lyrics.

In gospel song lyrics, we can identify some typical words, such as *"Deus"* (God) , *"Senhor"* (Lord), and *"Jesus"* (Jesus); in *funk-carioca*, songs have the words *"bonde"* (tram), *"chão"* (floor) and *"baile"* (dance ball), all used as slang; in *sertanejo*, some of the most common words are *"amor"* (love), *"coração"* (heart) and *"saudade"* (longing). The occurrence of these typical words could contribute to the higher performance of F1-scores in these genres.

[9] https://scikit-learn.org/

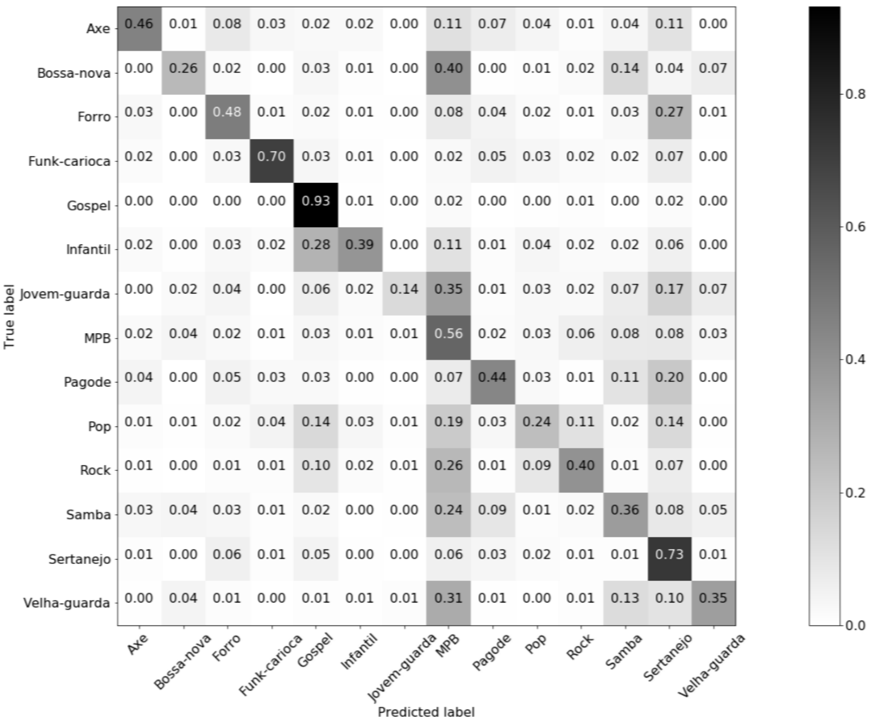
[10] https://keras.io/

[11] https://www.tensorflow.org/

Table 3: Detailed result of BLSTM

Genre	**F1-score**
Gospel	0.89
Funk-carioca	0.70
Sertanejo	0.69
Forró	0.53
Axé	0.49
MPB	0.49
Pagode	0.48
Infantil	0.47
Rock	0.46
Velha-guarda	0.38
Samba	0.35
Bossa-nova	0.31
Pop	0.26
Jovem-guarda	0.19
Average	*0.481*

The *bossa-nova* and *jovem-guarda* genres, which have few instances in the dataset, are among the most difficult ones to classify using the model. The *pop* genre, by contrast, has a small distribution between the number of songs and the number of artists, and could not be well classified by our model. This may indicate that our model was unable to identify a pattern due to the low number of songs per artist, or that the song lyrics of this genre cover several subjects that are confused with other genres.



	Axe	Bossa-nova	Forro	Funk-carioca	Gospel	Infantil	Jovem-guarda	MPB	Pagode	Pop	Rock	Samba	Sertanejo	Velha-guarda
Axe	0.46	0.01	0.08	0.03	0.02	0.02	0.00	0.11	0.07	0.04	0.01	0.04	0.11	0.00
Bossa-nova	0.00	0.26	0.02	0.00	0.03	0.01	0.00	0.40	0.00	0.01	0.02	0.14	0.04	0.07
Forro	0.03	0.00	0.48	0.01	0.02	0.01	0.00	0.08	0.04	0.02	0.01	0.03	0.27	0.01
Funk-carioca	0.02	0.00	0.03	0.70	0.03	0.01	0.00	0.02	0.05	0.03	0.02	0.02	0.07	0.00
Gospel	0.00	0.00	0.00	0.00	0.93	0.01	0.00	0.02	0.00	0.00	0.01	0.00	0.02	0.00
Infantil	0.02	0.00	0.03	0.02	0.28	0.39	0.00	0.11	0.01	0.04	0.02	0.02	0.06	0.00
Jovem-guarda	0.00	0.02	0.04	0.00	0.06	0.02	0.14	0.35	0.01	0.03	0.02	0.07	0.17	0.07
MPB	0.02	0.04	0.02	0.01	0.03	0.01	0.01	0.56	0.02	0.03	0.06	0.08	0.08	0.03
Pagode	0.04	0.00	0.05	0.03	0.03	0.00	0.00	0.07	0.44	0.03	0.01	0.11	0.20	0.00
Pop	0.01	0.01	0.02	0.04	0.14	0.03	0.01	0.19	0.03	0.24	0.11	0.02	0.14	0.00
Rock	0.01	0.00	0.01	0.01	0.10	0.02	0.01	0.26	0.01	0.09	0.40	0.01	0.07	0.00
Samba	0.03	0.04	0.03	0.01	0.02	0.00	0.00	0.24	0.09	0.01	0.02	0.36	0.08	0.05
Sertanejo	0.01	0.00	0.06	0.01	0.05	0.00	0.00	0.06	0.03	0.02	0.01	0.01	0.73	0.01
Velha-guarda	0.00	0.04	0.01	0.00	0.01	0.01	0.01	0.31	0.01	0.00	0.01	0.13	0.10	0.35

Figure 4: Normalized confusion matrix

Figure 4 shows the confusion matrix of the results produced by our BLSTM model. We can notice that many instances of class *forró* are often confused with class *sertanejo*. Indeed, these two genres are very close. Both *Forró* and *sertanejo* have as theme the cultural and daily aspects of the Northeast region of Brazil. Instances of class *infantil* are often confused with class *gospel*: in *infantil* we have music for children for both entertainment and education. In some of the songs, songwriters try to address religious education, which could explain the confusion between those genres. The

MPB (Brazilian Popular Music) genre was the most confused of all, which may indicate that song lyrics of this genre cover a wide range of subjects that intersect with other genres.

5 Conclusion and Future Works

In this work we constructed a dataset of $138,368$ Brazilian song lyrics distributed in 14 genres. We applied SVM, Random Forest, and a Bidirectional Long Short-Term Memory (BLSTM) network combined with different word embeddings techniques to address the automatic genre classification task based only on the song lyrics. We compared the results between the different combinations of classifiers and word embedding techniques, concluding that our BLSTM combined with the Wang2Vec pre-trained model obtained the best F1-score classification result. Beside the dataset construction and the comparison of tools, this work also evidences the lack of an absolute superiority between the different techniques of word embeddings, since their use and efficiency in this specific task showed to be very closely related to the classification technique.

As future work, it is possible to explore the dataset to identify genre or artist similarities, generating visualizations that may or may not confirm aspects pre-conceived by the consumers of Brazilian music. It is also possible to perform classification tasks by artists of a specific genre.

References

[1] Teh Chao Ying, Shyamala Doraisamy, and Lili Nurliyana Abdullah. Genre and mood classification using lyric features. In *2012 International Conference on Information Retrieval & Knowledge Management*, pages 260–263. IEEE, 2012.

[2] Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, 23(2):133–141, 2006.

[3] Cyril Laurier, Jens Grivolla, and Perfecto Herrera. Multimodal music mood classification using audio and lyrics. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 688–693. IEEE, 2008.

[4] Yajie Hu, Xiaoou Chen, and Deshun Yang. Lyric-based song emotion detection with affective lexicon and fuzzy clustering method. In *ISMIR*, pages 123–128, 2009.

[5] Xiao Hu and J Stephen Downie. When lyrics outperform audio for music mood classification: A feature analysis. In *ISMIR*, pages 619–624, 2010.

[6] Alexandros Tsaptsinos. Lyrics-based music genre classification using a hierarchical attention network. *arXiv preprint arXiv:1707.04678*, 2017.

[7] Michael Fell and Caroline Sporleder. Lyrics-based analysis and classification of music. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 620–631, 2014.

[8] Menno Van Zaanen and Pieter Kanters. Automatic mood classification using tf* idf based on lyrics. In *ISMIR*, pages 75–80, 2010.

[9] Akshi Kumar, Arjun Rajpal, and Dushyant Rathore. Genre classification using word embeddings and deep learning. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2142–2146. IEEE, 2018.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[11] Nathan Hartmann, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jessica Rodrigues, and Sandra Aluisio. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *arXiv preprint arXiv:1708.06025*, 2017.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.