UNIVERSIDADE DE SÃO PAULO

Ian Carvalho

A CASE STUDY IN MUSIC STYLE'S SIMILARITY MEASUREMENT USING DEEP LEARNING

SÃO PAULO
2017

IAN CARVALHO

A CASE STUDY IN MUSIC STYLE'S SIMILARITY MEASUREMENT USING DEEP LEARNING

Undergraduate dissertation thesis submited to *Universidade de São Paulo* as a requirement for obtaining the Computer Science Bachelor Degree

Supervisor: Prof. Dr. Alfredo Goldman

São Paulo, 2017

To my parents, the gave me the encouragement and support necessary to get where I am. To my life partner, Tatiana, that was there in the best and worst moments in the monograph trajectory. To my supervisor, mentor and friend, Alfredo Goldman, whom without its help would not be possible to complete the work.

"Music gives a soul to the universe,
wings to the mind, flight to the imagination
and life to everything.
(Plato)

Abstract

Resumo em Inglês This work tries to establish a similarity measure for music genres in

Portuguese utilizing a deep neural network. It shows the necessary steps in order to gather and process the lyrics with a crawler and them proceeds to use this dataset in order to train two neural networks utilizing Long Short-Term Memory and Hierarchical Temporal Memory architectures. From the confusion matrices generated by those classifiers, this works then calculates a similarity matrix that indicate the likeness between two genres.
Keywords: music classification, deep learning, music similarity

List of Figures

List of Tables

Contents

1 Introduction

This work is motivated by the interest in how can Deep Neural Networks used in order to classify music titles according to genre using their lyrics. As said in section 1.3, there are many studies regarding the use of audio for genre classification, but lyrics are an important aspect of modern music.

It may seem counter intuitive to use lyrics as it seems genre are more related to the harmonic and rhythmic part of a song. Nevertheless, our approach considers that there may be textual factors the are common within the genres and can, therefore, be used to classify the lyrics.

1.1 Problem description

Measure the similarity between music genres accurately is a problem with many real world applications as music recommendation models, automatic playlists generation and musical search engines [Knees e Schedl 2013]. According to [Downie e Cunningham 2002] and [Lee e Downie 2004], music gender has an important role in Music Information Retrieval (MIR) theory. It is one of the most common parameters people use to either search or discover new song. Music genre classification is not an easy task, even for humans. [Perrot e Gjerdigen 1999] says people are able to classify only 70% of the 3 seconds samples correctly according to their style.

In order to precisely define the problem some definitions are important.

1.1.1 Music Genre

There is a debate whether genres are intrinsic or extrinsic property of music titles, but they are one form of music classification extensively used in music industry and in academic studies. [Aucouturier e Pachet 2003] suggests that genre is an extrinsic music's property as the classification of titles in genres depends more on cultural habits rather than on its specifics features. The study conducted an extensive analysis on different genre taxonomies and proposed three main approaches to classify songs according to their gender: manual, prescriptive, and emergent genre classification.

Manual genre classification is an attempt to gather expert human knowledge in order to group song titles in categories. The total number of genres can vary within different taxonomic systems, as there is not an expert consensus on what those categories really are and sometimes it brings cultural and subjective biases into the categorization. In the scope of this work, we were not able to find an standard manual attempt to define the

music genres in Brazilian songs, even though there are taxonomies proposals (e.g. [Pereira et al. 2009]) they do not have the necessary amount of classified songs, so this work will adopt the same taxonomy as one of the biggest Brazilian lyrics website: <letra.mus.br>.

1.1.2 Similarity measurement

There are different ways one can measure the similarity between two genres. In [Esparza, Bello e Humphrey 2015] there is a rhythm-based approach to genre classification, other studies harmonic features are used as input to the classifier. This work will attempt to define the similarity between genres using their lyrics.

This work will define the similarity measurement between two genres as the normalized $L2$ distance between their respective row vector in confusion matrix generated by a deep learning classifier. This definition is built upon the idea that confusion matrices can be used for more than simply evaluating the performance of an classifier. It is also an indicator of similarity amongst classes (genres), as pointed in [Godbole 2002]. The details about how to derive an similarity measurement from the confusion matrix will be explained in chapter 2.

1.2 Objectives

The main objective of this paper is to calculate the similarity amongst Brazilian's music styles utilizing a deep neural network. In order to do so, it is necessary to define a dataset to work with. This dataset will be then split into training and validation sets. After that, a deep neural network will be trained to classify the lyrics according to their genres.

In practical terms, being $g_1, g_2, ..., g_n$ the set of all the music genres selected for the experiment, the $C_{n \times n}$ matrix, called confusion matrix, will be built with the following procedure: given any $0 < i, j \leq n$, the value of C_{ij} will be the number of times the classifier predicts genre g_i when the actual genre is g_j. After that, for each pair of rows (i, j) with $0 < i, j \leq n$, the $L2$ distance will be calculated to obtain d_{ij}. All the distances will be normalized so they are in the range $0 < d_{ij} \leq n$. The similarity between genders g_i and g_j, will be such $s_{ij} = 1 - d_{ij}$.

This work hopes to find that is even though the lyrics of a song are not necessarily tied to a genre, as it is possible to play a song in different music styles, there will be enough similar semantical features to classify the lyrics correctly, given a big enough dataset of lyrics and their genres for training and evaluation.

Also, there is an intuitive notion that some music styles are more closely related to each others compared with others. This can be related to year of composition, the

composers demographics, and target audience for the songs. While this notion and the reasons behind it are not in the scope of this work, it will compare the results of the experiment with the common sense and check if it agrees or not with it.

1.3 Related works

Text classification is a relevant problem for many fields of study both inside and outside computer science being researched by data mining, machine learning, database, and information retrieval communities alike according to [Aggarwal e Zhai 2012]. Spam filters, document organization, sentiment analysis are some of the applications of the problem.

When trying to classify song titles according to their genre, text classification is not the only approach. Is possible to categorize the studies regarding supervision:

- Supervised
- Semi-supervised
- Unsupervised

Respectively, [Yeh e Yang 2012], [Poria et al. 2013] and [Lee et al. 2009] show examples of each category.

The [Scaringella, Zoia e Mlynek 2006] survey compile some techniques used for music classification in 2006. Even though those techniques evolved a lot with the upcoming of DNNs, it presents valuable information on kind of features and technologies utilized for genre classification. The study is extensive but focus on audio features extraction in order to classify the title.

[Vieira et al. 2013] proposes a quantitative approach to evaluate and compare attributes between classical music and philosophy. The authors use this evaluation in order to analyze the temporal evolution of both humanities' areas and they obtain data by letting humans assign scores to each of eight selected attributes from objects.

In [Vieira et al. 2013], they use a human based approach for evaluation in arbitrarily defined parameters and uses multi-variable statistics in order to provide a quantitative analysis of evolution in music and philosophy. In this work, we try to suggest a more generic approach to evaluating and comparing text by utilizing deep learning. The reasoning behind is to let the neural network figures what the most important features are when comparing lyrics which makes the process more scalable. This comes at a trade-off of not knowing which features are being analyzed by the neural network.

1.4 Organization

This text is organized as follows:

The chapter 1 provides a description of the problem, objects and related works.

In chapter 2, this work presents a detailed approach of the necessary steps in order to obtain the dataset, it describes how data is preprocessed and fed into the deep neural network and how the results will be analyzed.

In chapter 3, the result analysis described in the previous chapter is applied to the experiment.

The chapter 4 summarizes the key learnings from the experiments and brings a subjective analysis about the experience of writing this paper.

2 Methodology

2.1 Collecting the data

Training DNNs usually requires a great amount of data. In order to develop a classifier with supervised learning it is important that each lyrics to be labeled with its genre. The first attempt to obtain a dataset was to reach the *vagalume.com.br* API, but the lack of querying parameters for music genre made this approach not viable.

There are some international APIs also available, but they either had a small amount of Brazilian's songs or would be very expensive to use.

So in order to gather the data, a crawler bot, a program that automatically captures a website information, was developed in Python for *letras.mus.br*. The criteria for choosing this website was the diversity of genres and songs and a structure that made the website easy to crawl. The crawler transverses the HTML structure utilizing XPath, a query language for consulting and selecting nodes in a XML document.

For a list of genres, the bot would access *letras.mus.br/mais-acessadas/<genero>*, and look for the most popular artists. The //**ol[@class="top-list_art"]**//**a** XPath queries the links for the artists and the program follows them. For each link, it is possible to select the links for every single music using the //**ul[@class="cnt-list"]**//**a** XPath. From there, the //**div[@class="cnt-letra"]**//**p** XPath is traversed in other to get all paragraphs from the lyrics. The data gathered by the robot is save in a CSV file, that shows for each line the song title, along with their lyrics, url and genre.

The list of genres chosen for this experiment was fairly arbitrary. But all choices followed some requirements:

- Only root genres would be considered. So the leaf genres *Samba enredo* and *Samba de raiz* would be classified as their root genre *Sampa*.
- Most of genre's songs had to be in Portuguese. In order for simplify filtering.
- At least 100 different artists.
- At least 5.000 different songs.

Using that criteria the following genders were chosen:

- Axé
- Bossa Nova

- Forró
- Funk
- Pagode
- Samba
- Sertanejo
- MPB

Even filtering by gender mostly containing Portuguese song, the downloaded dataset contained few songs in English and Spanish. The Python package *langdetect* was used to determine the language of the lyrics and filter those which were not in Portuguese. After that, all songs in English were totally filtered out of the dataset. A manual search filtered the few remaining Spanish songs. After that, songs were randomly chosen to make sure the lyrics were evenly distributed across the genres.

2.2 Picking a framework

Frameworks provide a layer of abstraction helping the developer focus on the higher-level implementation instead of the low-level details. There are many different Deep Learning frameworks available nowadays, so it is important to analyze their good and bad aspects in order to pick one that is more suitable for the experiment. The frameworks Keras, Tensorflow, Theano and CNTK were selected for this analysis because they are all open source, written in Python or have bindings available for Python, and have a huge community behind them.

Those criteria are important because a huge community provides quicker support, a broader access to documentation and learning materials and considering the frameworks are open source, a bigger community means more people developing and perfecting the code. Python was the language of choice for this project because of author's proficiency with the language.

2.2.1 Theano

One of the first frameworks of its kind and discontinued since 2017, it can be used for machine learning applications in general, not just for deep learning, as it provides a really low level interface. One positive aspect is Theano has a big *google user groups* and *github community*. Theano documentation is simple and the important topics are explained by an example, making it easier to grasp upon the framework.

2.2.2 Tensorflow

At this moment, TensorFlow by Google is the most used deep learning framework considering Github stars, forks and Stack Overflow activity. The very active community provides great support. The documentation is very through, but lacks some practical examples. The API is very low level as the framework can be used for most kind of numeric calculations. Their data flow graph allow flexibility and parallelism.

2.2.3 CNTK

The most recent player in deep learning frameworks. Is is built by Microsoft and provides a very good performance compared to Tensorflow. As it is newer, the framework is still gaining popularity due its good documentation and a slightly higher level API, comparing to Theano and Tensorflow.

2.2.4 Keras

According to their website "Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research."

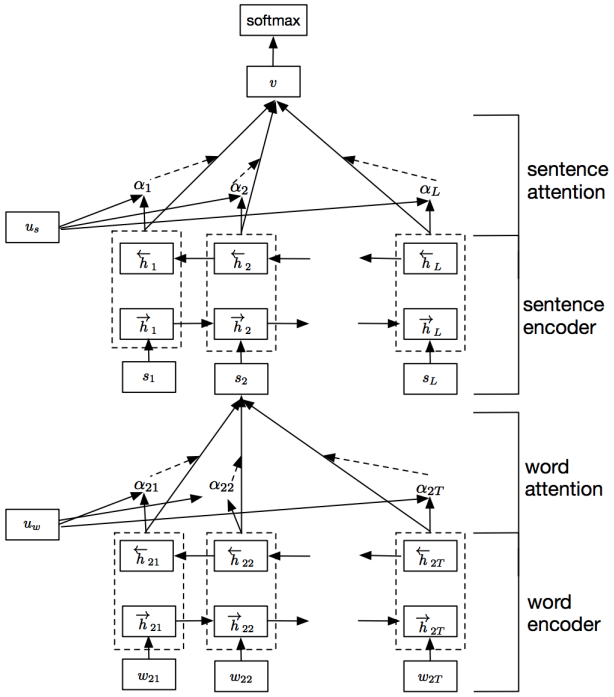
It has a great community both in github and stackoverflow allowing questions to be solved quickly. The documentation is simpler than the previous frameworks, but sometimes it is too simples, leaving important details out.

It was the choice of framework because of its gentle learning curve and high-level capabilities that allow prototypes to be quickly developed. Also, it leverages the strengths of the previous frameworks as they can used as back-end for the API.

2.3 Neural Network Architecture

In this work, two DNNs were built, each one with a different architecture, so results could be compared. The architecture were chosen according to their results in text classification tasks in scientific literature. A brief description of those networks will be presented here, while a more detailed definition for LTSM and HAT architectures is not in the scope of this work, refer to [Hochreiter e Schmidhuber 1997] and [Yang et al. 2016] for a deep understanding mechanics and mathematical reasoning behind those architectures.

Figure 1 – HAN Architecture



Source: [Yang et al. 2016, p. 02]

2.3.1 Hierarchical Attention Networks

HAN were used in [Yang et al. 2016], [Tsaptsinos 2017] and [Pappas e Popescu-Belis 2017] to successfully classify text documents into categories. The neural network tries to incorporate elements from the text structure in the architecture itself.

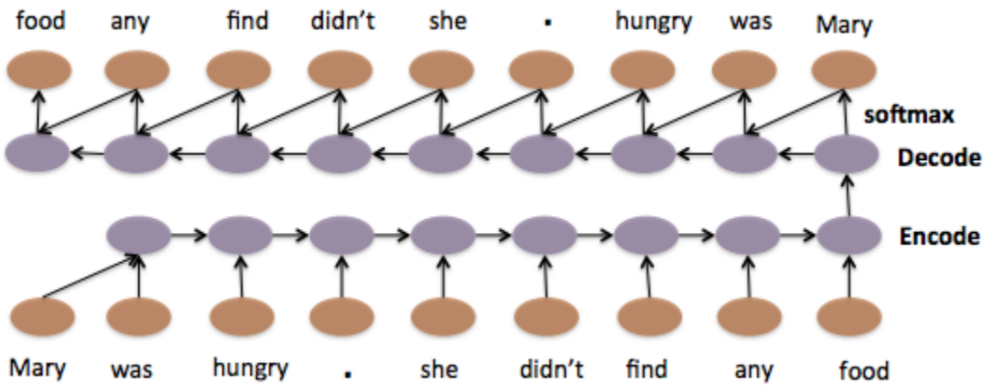
One of the elements is hierarchy - as sentences are formed from words and a number of sentences form a document). So, in order to construct a representation from a document the neural network first build representations of the sentences and then aggregates it to the document representation.

The other is that the importance of a word or a sentence depends a lot on their context. Because of that, the model includes two attention mechanisms one for words and other for sentences.

2.3.2 Long Short-Term Memory

The Long Short-Term Memory (LSTM) architecture was first proposed at [Hochreiter e Schmidhuber 1997] and since then it has been used for many applications; notably, text generation (e.g [Sutskever, Martens e Hinton], sequence to sequence modeling

Figure 2 – LSTM Architecture



Source: [Li, Luong e Jurafsky 2015, p. 04]

(e.g. [Sutskever, Vinyals e Le 2014]) and text categorization (e.g. [Li, Luong e Jurafsky 2015]). LSTM is specific type of Recurrent Neural Network (RNN) that address some limitations of previous RNNs architectures.

RNNs are neural networks built so that connections between neurons are modeled in such a way they store in their internal state the sequence of previous inputs. This internal memory allows it to excel at modeling sequence of inputs that have some kind of temporal relationship between them. In figure 2, it is possible to see how each individual input in encoded using information from the previous state. The same holds for decoding, the neural network is able to predict the next word based on the context of previous predicted words.

One of the difficulty faced gradient-based learning methods, specially in RNNs, is the vanishing gradient problem. In Recurrent Neural Networks the stochastic gradient points towards the local minimum of the objective function, in this case back-propagation of errors can lead to two bad scenarios, according to [Hochreiter e Schmidhuber 1997]. In the first scenario the gradient ends up exploding and the weights in the neural network start oscillating and in this case the objective function may not converge to the local minimum. Other possible scenario, is the vanishing of the gradient which would decrease the learning speed to the point the neural network stops learning.

To overcome those limitations, [Hochreiter e Schmidhuber 1997] propose a new type of recurrent neural network based architecture that enforces constant error flow through the internal state of new special units, called Long Short-Term Memory units. The constant error flow prevents the gradient to explode or vanish and allow the neural network to handle very noise, distributed representations, and continuous values.

Table 1 – One-hot encoding for selected genres

	Axé	Bossa Nova	Forró	Funk	Pagode	Samba	Sertanejo	MPB
Axé	1	0	0	0	0	0	0	0
Bossa Nova	0	1	0	0	0	0	0	0
Forró	0	0	1	0	0	0	0	0
Funk	0	0	0	1	0	0	0	0
Pagode	0	0	0	0	1	0	0	0
Samba	0	0	0	0	0	1	0	0
Sertanejo	0	0	0	0	0	0	1	0
MPB	0	0	0	0	0	0	0	1

2.4 Word Embeddings

When working with text classification, it is necessary to encode words so they can be transformed in a more meaningful representation for the neural network tasks [Hartmann et al. 2017]. A way to provide this representations is to use a vector of real valued numbers known as *word embeddings*. Those vectors are able to capture syntactic, semantic and morphological knowledge and, therefore, have become popular in the Natural Language Processing (NLP) community.

Word embeddings can be learned from a huge text corpus using different approaches. The choice of word embedding algorithm can impact on the performance of the classifier, so it is important to consider their particularities. There are mainly two kinds of word embedding algorithms: context-predicting and count-based. In [Baroni, Dinu e Kruszewski 2014], the authors show the context-predicting models have an advantage in tasks as relatedness, categorization, and analogy.

This work uses the models available at Interinstitutional Center for Computational Linguistics word embedding repository <http://www.nilc.icmc.usp.br/nilc/index.php/repositorio-de-word-embeddings-do-nilc>, as it has an extensive word embeddings collection with different dimensions and algorithms.

2.5 Experiment Description

As seen in section 2.4, it is important to use a representation for words that are suitable for inputing in a neural network. As [Hartmann et al. 2017] suggests, the word embeddings model *Wang2Vec* has good general performance for NLP task in Portuguese language, so it is the choice of embeddings for this work.

The lyrics datasets is loaded from a CSV file and manipulated as follow. First all-genres are encoded using a one-hot encoding. In this process the *genre* feature is transformed in a n-sized binary vector where n is the number of available genres. Each genre i, will be represented by a vector with element $v_i = 1$ and all the other equal 0 as

Table 2 – Confusion matrix example

	Axé	Bossa Nova	Forró	Funk
Axé	20	2	5	3
Bossa Nova	1	25	2	2
Forró	4	3	21	2
Funk	6	2	17	5

shows table 1.

After that, the lyrics are encoded utilizing the word embeddings. For better performance, small words (less than three characters) are removed and the 40.000 more frequent words in the whole corpus are selected.

The data is split equally in three creating the datasets for training, validation, and test. In order to maintain the distribution between genres equal in the sets, as an unbalanced may lead to some problems [Molinara, Ricamato e Tortorella 2007], each genre is divided in three randomly and assigned to one of the sets. At the end, each set would have about the same number of elements.

The training and the validation dataset will be used for the training of the deep neural network. The neural network will learn during 30 epochs - a full cycle across the training data - and for after epoch the progress will be assessed by running the classifier against the validation set. It is important to use a different dataset for validation to check if the classifier is able to generalize enough to correctly classify a unseen dataset.

In this phase, two problems can happen: over and underfitting. Overfitting can be seen when the classifier predicts accurately the genres of the lyrics within the training set, but performs poorly against unseen data. This shows the classifier has problems generalizing learning. Other sign of overfitting is that the after some epochs the objective function stops decreasing. The other possible problem is underfitting. In this case, the model cannot accurately predict the training dataset. It shows the model is unable to capture the relationship between inputs and their matching genres.

After the training, it is possible to obtain a classifier that performs well on training data and is able to correctly generalize to novel inputs. Then, for each lyric in the test set, the model is asked to predict its genre. The results are then represented as a confusion matrix. A matrix which every row represents a target class, and the columns represent the predicted value.

From that confusion matrix a similarity matrix is derived. Similarity matrix is the triangular matrix M which for every $i, j.i \leq j$ M_{ij} represents the similarity measure between g_i and g_j. This matrix is created by calculating the $L2$ distance d_{ij} between the rows associated with every pair on genres g_i, g_j, and setting $M_{ij} = 1 -_i^d j$, where ${}_i^d j$ is the normalized distance between g_i and g_j.

Table 3 – Similarity matrix for table 2

	Axé	Bossa Nova	Forró	Funk
Axé	1	0.0	0.24	0.38
Bossa Nova	-	1	0.02	0.06
Forró	-	-	1	0.81
Funk	-	-	-	1

3 Results

For this experiment, two neural networks were implemented and the procedures described in chapter 2 applied. The computer user for running the experiments was:

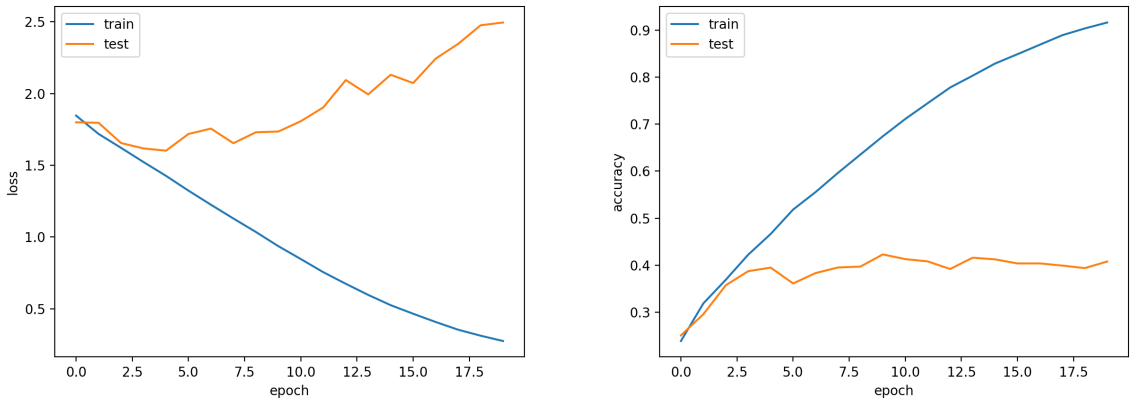
- Processor: Intel(R) Xeon(R) CPU E3-1230 V2 @ 3.30GHz
- RAM: $4x8$ GiB DDR3 @ 1333 MHz
- Graphic card: GeForce GTX TITAN X 12 GB GDDR5 VRAM @ 7 GHz

In order to properly calculate the similarity, it was first necessary to have a classifier that was good enough. There is no standard benchmark for lyric-based genre classification and previous similar results were around 45% accuracy ([Tsaptsinos 2017]). The 70% accuracy threshold was defined as it is a reasonable accuracy to achieve and, yet, it would be precise enough to indicate classes weren't being predicted by chance.

The previously cited neural network architectures (HAN and LSTM) were built and trained to obtain the results shown in **??** and 4. It is possible to see both architectures weren't able to score better than 50% accuracy. Also, the figure indicates some overfitting may be happen as accuracy in train dataset continues to grow as in test dataset is stalls.

In figures 7 and 6, there is a big improvement upon the previous attempt. The classifiers were able to reach the 70% accuracy and therefore could already be used for the similarity experiment. As the figure shows, there is still some room for improvement, but it is outside of the scope of this work to further improve the classifier accuracy as

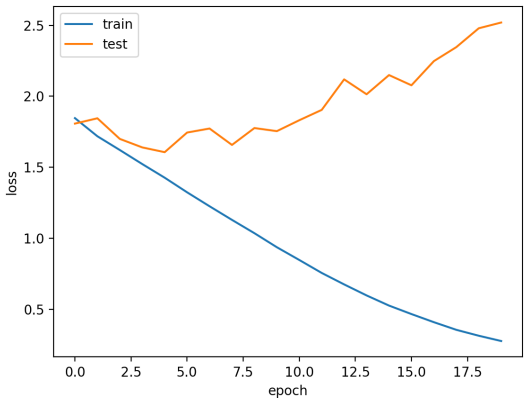
Figure 3 – LSTM Accuracy and Loss at First Attempt



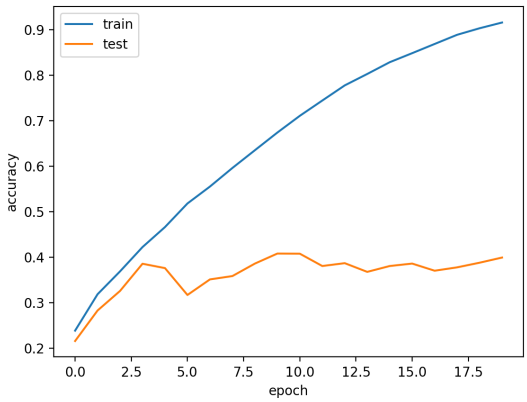
(a) LSTM Loss

(b) LSTM Accuracy

Figure 4 – HAN Accuracy and Loss at First Attempt

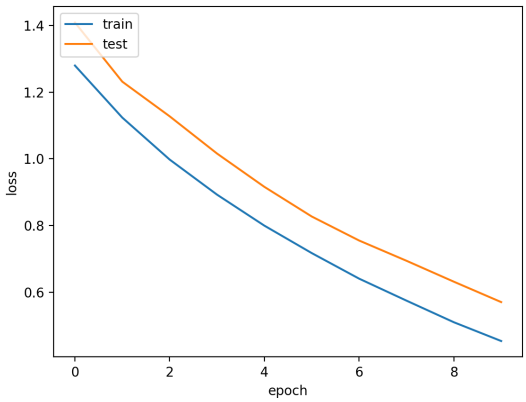


(a) HAN Loss

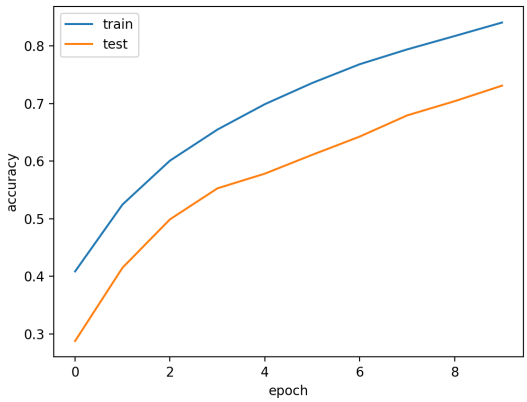


(b) HAN Accuracy

Figure 5 – LSTM Accuracy and Loss at Best Attempt

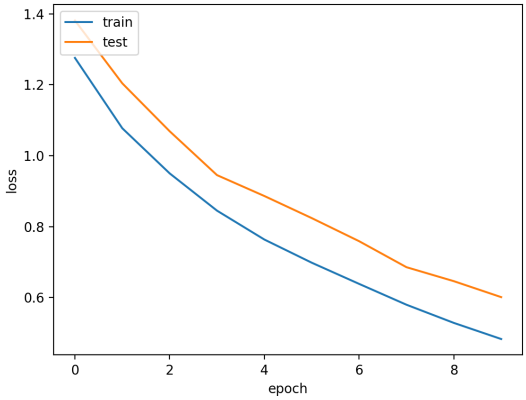


(a) LSTM Loss

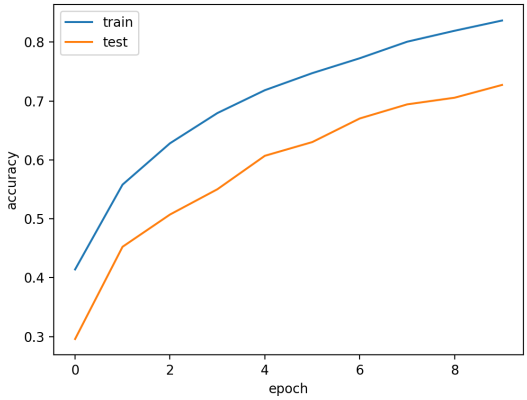


(b) LSTM Accuracy

Figure 6 – HAN Accuracy and Loss at Best Attempt

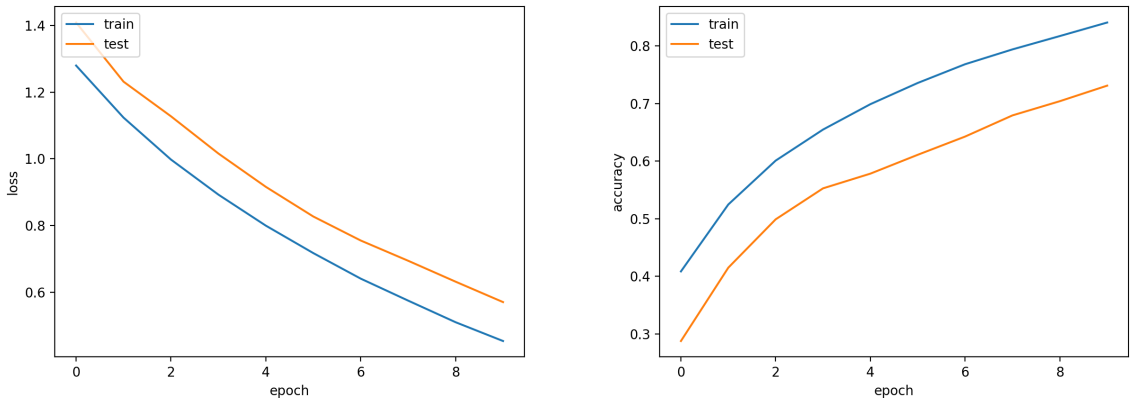


(a) HAN Loss



(b) HAN Accuracy

Figure 7 – LSTM Accuracy and Loss at Best Attempt



(a) LSTM Loss

(b) LSTM Accuracy

the genre similarities are a more important. At this point, the total time for training the neural network was about 2 hours, this limiting in a sense as any change in the model or hyper-parameters configuration would require a long time until the results were seem.

With the trained neural networks, it was now possible to use them to calculate similarity between the genres. In the figures 9 and 8 it is possible to see the similarity matrix for each NN.

In order to have a comparison baseline, three other classifiers were implemented using more traditional machine learning algorithms - Naive Bayes, K-Nearest Neighbors (KNN) and Support Vector Machines. Unfortunately, their accuracy was too low to provide any meaningful insight. On the other hand, the likeness of the similarity matrix in both DNNs, indicates that this results might be useful.

Figure 8 – Similarity Matrix - LSTM Architecture

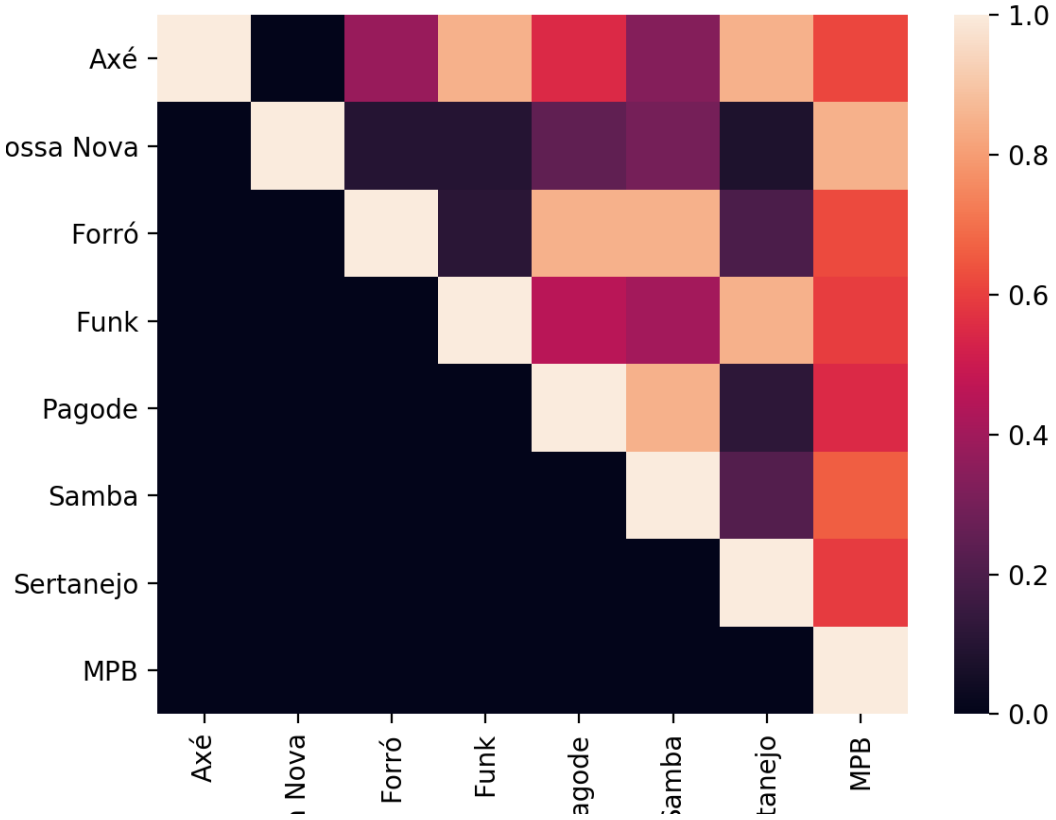
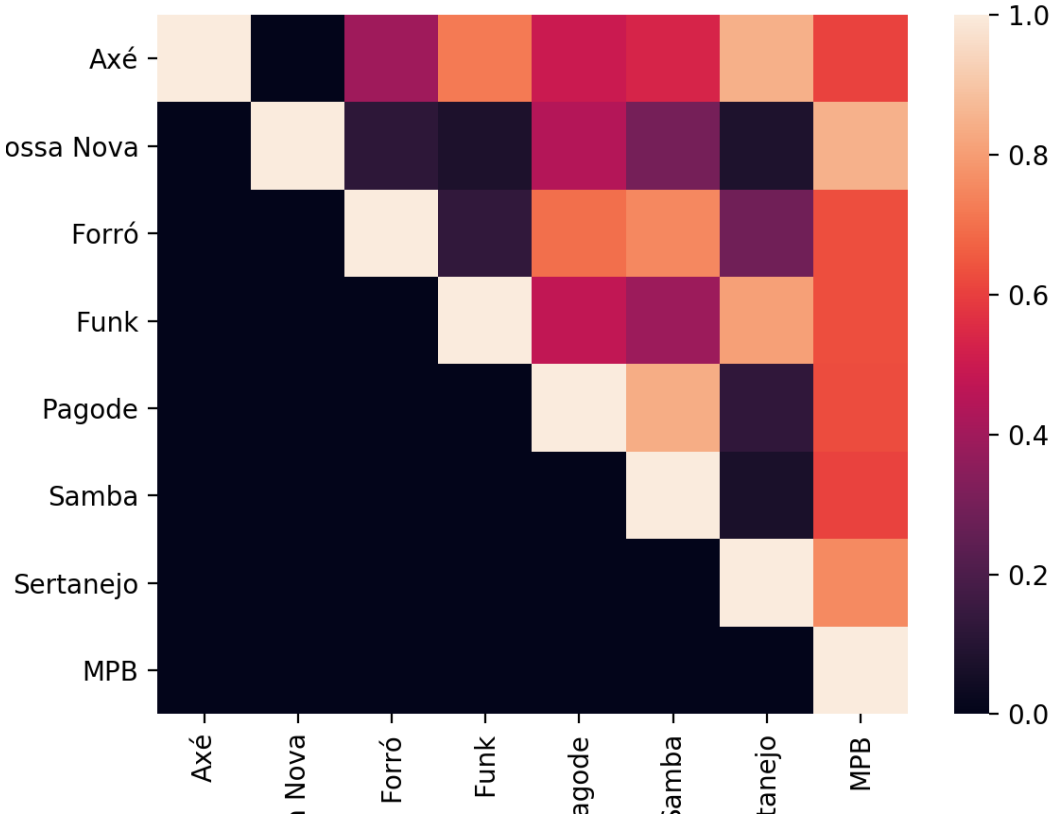


Figure 9 – Similarity Matrix - HAN Architecture



4 Conclusion

4.1 Experiment review

This work proposed to build an DNN classifier that was able to detect similarity among music genres. In order to do so, extensive research was made into the state-of-art algorithms for text classification.

During this process, the proposed experiment was successfully realized but the results were exactly the expected. That is because of the difficulty to establish a comparison baseline. The traditional machine learning algorithms were not able to classify accurately the lyrics in their genre and, therefore, did not produce an meaningful result. Regardless of this, the fact that both trained neural networks achieved similarity matrices that look alike is an indicator that this might be a method to calculate text classes similarity.

In the beginning of this work, the hypothesis assumed was that even though lyrics are not specifically tied to a genre - as it is possible to play a song in different styles - there is a group of common features within genres that can be learned and used for classification. The fact that the developed classifiers were able to accurately classify the lyrics suggests that the hypothesis is true.

4.2 Further Work

There are many possible lines of future works. for instance, attempt to parallelize the training of the neural network in order to make viable to train a larger dataset in a shorter amount of time. This would also allow more experimentation regarding the hyperparameter and model tweaking. The obvious one is the compare the results obtained in this text with obtained by different methods in order to better support the results here achieved.

It is possible to adopt the line used in [Vieira et al. 2013], using a quantitative approach to genre similarity. This way the survey results could the be compare to the ones obtained here. A broader analysis can be made considering not only text-based features but also include audio. A classifier can be built using audio features and used to compare if text similarity would also means audio similarity and otherwise.

In [Bogdanov 2013], the author uses music similarities in order to create a recommendation system for songs using only audio in order to establish music similarity. Further work can be developed by utilizing text similarity of the songs and following the cited approach to build the recommendation system.

4.3 Subjective analysis

I've started my graduation at BCC in 2013, after studying Electric Engineering for 2 years in the Santa Catarina's Federal University. The decision of changing my area of study and future professional life came after a long process of maturing, growth and self-knowledge that oriented my relationship with the Computer Science's graduation course. I knew from the beginning the academic challenges that I could and would face during the next 5 years, but was engaged and motivated to do so because I knew this is my area of real interest. Nevertheless, in order to change courses I had to commit to working while studying due to family conditions, and that ended up being one of the biggest difficulties of this experience.

When it came to the moment of deciding my area of research for this undergraduate dissertation I had worked already in some interesting areas in Brazil such as programming, game development and virtual reality. I knew these areas well enough so wanted to focus my energy in something I was not fully exposed yet. During my exchange program in the US I was able to be a trainee during 3 moths in IBM, in the area of artificial intelligence. I was intellectually defied by this area of research and highly motivated to study it further considering the professional opportunities around it. After that experience I decided this dissertation should allow me to develop further my knowledge on AI.

In the beginning I wanted to develop a market oriented AI solution, managing and analyzing data generated by consumer oriented companies to improve their sales efficiency. AI could help companies to better understand consumer behavior and trends using their own data to generate quality information. I've reached out to some companies looking for a partnership at the time but unfortunately, due to the confidentiality around their data bases and long internal decision making processes I had to change my project to be able to develop it in time. This was one key difficulty on this learning process but it led me to find a research subject close to my personal passions, Music. I was able than to combine an area of academic, professional and personal interests in this research project which allowed me to experience the idea of working with passion and the difference it makes on bringing both personal and professional realizations.

As I mentioned earlier I had to work while studying during the 5 years at college, in this final year though I had an even bigger workload which distracted me from my academic obligations. My working and studying hours would sum up to 14 hours a day driving me to exhaustion and emotional unbalances during the last semester of 2017. It delayed my graduation and, honestly, if it weren't for the support of my supervisor, my passion for the theme I was researching and my strength of will I might not have been able to finish.

The BCC experience, but specially the development of this undergraduate dis-

sertation were key to develop my academic and professional skills. But they were even more significant in my growth as a human being. It made me learn how to prioritize, find balance, pass through barriers and strengthened me to the next chapter of my life. For all of this I can say I am grateful for all of it and will always tap into what I've learned in the future challenges I face.

Bibliography

AGGARWAL, C. C.; ZHAI, C. A survey of text classification algorithms. In: _____. *Mining Text Data*. Boston, MA: Springer US, 2012. p. 163–222. ISBN 978-1-4614-3223-4. Disponível em: <https://doi.org/10.1007/978-1-4614-3223-4_6>. Cited at page 17.

AUCOUTURIER, J.-J.; PACHET, F. Representing musical genre: A state of the art. *Journal of new music research*, Taylor & Francis, v. 32, n. 1, p. 83–93, 2003. Cited at page 15.

BARONI, M.; DINU, G.; KRUSZEWSKI, G. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. [S.l.: s.n.], 2014. v. 1, p. 238–247. Cited at page 24.

BOGDANOV, D. *From music similarity to music recommendation: Computational approaches based on audio features and metadata*. 227 p. Tese (Doutorado) — Universitat Pompeu Fabra, Barcelona, Spain, 09/2013 2013. Cited at page 33.

DOWNIE, J. S.; CUNNINGHAM, S. J. Toward a theory of music information retrieval queries: System design implications. 2002. Cited at page 15.

ESPARZA, T. M.; BELLO, J. P.; HUMPHREY, E. J. From genre classification to rhythm similarity: Computational and musicological insights. *Journal of New Music Research*, Routledge, v. 44, n. 1, p. 39–57, 2015. Disponível em: <https://doi.org/10.1080/09298215.2014.929706>. Cited at page 16.

GODBOLE, S. Exploiting confusion matrices for automatic generation of topic hierarchies and scaling up multi-way classifiers. 03 2002. Cited at page 16.

HARTMANN, N. et al. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *CoRR*, abs/1708.06025, 2017. Disponível em: <http://arxiv.org/abs/1708.06025>. Cited at page 24.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Cited 3 times at pages 21, 22, and 23.

KNEES, P.; SCHEDL, M. A survey of music similarity and recommendation from music context data. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, ACM, v. 10, n. 1, p. 2, 2013. Cited at page 15.

LEE, H. et al. Unsupervised feature learning for audio classification using convolutional deep belief networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2009. p. 1096–1104. Cited at page 17.

LEE, J. H.; DOWNIE, J. Survey of music information needs, uses, and seeking behaviours: Preliminary findings. 01 2004. Cited at page 15.

LI, J.; LUONG, M.; JURAFSKY, D. A hierarchical neural autoencoder for paragraphs and documents. *CoRR*, abs/1506.01057, 2015. Disponível em: <http://arxiv.org/abs/1506.01057>. Cited at page 23.

MOLINARA, M.; RICAMATO, M. T.; TORTORELLA, F. Facing imbalanced classes through aggregation of classifiers. In: *14th International Conference on Image Analysis and Processing (ICIAP 2007)*. [S.l.: s.n.], 2007. p. 43–48. Cited at page 25.

PAPPAS, N.; POPESCU-BELIS, A. Multilingual hierarchical attention networks for document classification. *CoRR*, abs/1707.00896, 2017. Disponível em: <http://arxiv.org/abs/1707.00896>. Cited at page 22.

PEREIRA, E. M. et al. Estudos sobre uma ferramenta de classificação musical. [sn], 2009. Cited at page 16.

PERROT, D.; GJERDIGEN, R. Scanning the dial: An exploration of factors in the identification of musical style. In: *Proceedings of the 1999 Society for Music Perception and Cognition*. [S.l.: s.n.], 1999. p. 88. Cited at page 15.

PORIA, S. et al. Music genre classification: A semi-supervised approach. In: SPRINGER. *Mexican Conference on Pattern Recognition*. [S.l.], 2013. p. 254–263. Cited at page 17.

SCARINGELLA, N.; ZOIA, G.; MLYNEK, D. Automatic genre classification of music content: a survey. *Signal Processing Magazine, IEEE*, v. 23, n. 2, p. 133–141, 2006. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1598089>. Cited at page 17.

SUTSKEVER, I.; MARTENS, J.; HINTON, G. *Generating Text with Recurrent Neural Networks*. Cited at page 22.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014. Disponível em: <http://arxiv.org/abs/1409.3215>. Cited at page 23.

TSAPTSINOS, A. *Lyrics-Based Music Genre Classification Using a Hierarchical Attention Network*. 2017. Cited 2 times at pages 22 and 27.

VIEIRA, V. et al. A quantitative approach to evolution of music and philosophy. 2013. Cited at page 17.

VIEIRA, V. et al. A quantitative approach to evolution of music and philosophy. 2013. Cited at page 33.

YANG, Z. et al. Hierarchical attention networks for document classification. In: *HLT-NAACL*. [S.l.: s.n.], 2016. Cited 2 times at pages 21 and 22.

YEH, C.-C. M.; YANG, Y.-H. Supervised dictionary learning for music genre classification. In: ACM. *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*. [S.l.], 2012. p. 55. Cited at page 17.